Path-metrics, pruning, and generalization

Antoine Gonon, Nicolas Brisebarre, Elisa Riccietti, Rémi Gribonval

Path-metrics, pruning, and generalization

Antoine Gonon
Univ Lyon, EnsL, UCBL,
CNRS, Inria, LIP

Nicolas Brisebarre
Univ Lyon, EnsL, UCBL,
CNRS, Inria, LIP

Elisa Riccietti
Univ Lyon, EnsL, UCBL,
CNRS, Inria, LIP

Rémi Gribonval
Univ Lyon, EnsL, UCBL,
CNRS, Inria, LIP

Abstract

Analyzing the behavior of ReLU neural networks often hinges on understanding the relationships between their parameters and the functions they implement. This paper proves a new bound on function distances in terms of the so-called path-metrics of the parameters. Since this bound is intrinsically invariant with respect to the rescaling symmetries of the networks, it sharpens previously known bounds. It is also, to the best of our knowledge, the first bound of its kind that is broadly applicable to modern networks such as ResNets, VGGs, U-nets, and many more. In contexts such as network pruning and quantization, the proposed path-metrics can be efficiently computed using only two forward passes. Besides its intrinsic theoretical interest, the bound yields not only novel theoretical generalization bounds, but also a promising proof of concept for rescaling-invariant pruning.

1 Introduction

An important challenge about neural networks is to upper bound as tightly as possible the distances between the so-called realizations (*i.e.*, the functions implemented by the considered network) $R_\theta, R_{\theta'}$ with parameters θ, θ' when evaluated at x, in terms of a (pseudo-)distance $d(\theta, \theta')$ and a constant C_x:

$$\|R_\theta(x) - R_{\theta'}(x)\|_1 \leqslant C_x d(\theta, \theta').$$

Such an inequality could be crucially leveraged to derive generalization bounds [Neyshabur et al., 2018] or theoretical guarantees about pruning or quantization algorithms [Gonon et al., 2023]. This type of bound is for example known with

$$d(\theta, \theta') := \|\theta - \theta'\|_\infty, \qquad C_x := (W\|x\|_\infty + 1)WL^2R^{L-1}, \tag{1}$$

in the case of a layered fully-connected neural network $R_\theta(x) = M_L \,\mathrm{ReLU}(M_{L-1} \dots \mathrm{ReLU}(M_1 x))$ with L layers, maximal width W, and with weight matrices M_ℓ having some operator norm bounded by R [Gonon et al., 2023, Theorem III.1 with $p = \infty$ and $q = 1$][Neyshabur et al., 2018, Berner et al., 2020]. This known bound is however not satisfying at least for two reasons:

- it is **not invariant under neuron-wise rescalings** of the parameters θ that leave unchanged its realization R_θ, leading to crude dependencies in R and L; and
- it **only holds for simple fully-connected models organized in layers**, but not for modern networks that include pooling, skip connections, etc.

To circumvent these issues, this work proposes to leverage the so-called *path-lifting* (together with its norm, called the *path-norm*), a tool that has recently emerged [Stock and Gribonval, 2023, Bona-Pellissier et al., 2022, Marcotte et al., 2023, Gonon et al., 2024] in the theoretical analysis of modern

Preprint. Under review.

neural networks with positively homogeneous activations. Its invariance under some rescaling symmetries of the network is nicely complemented by the ease of computation of the path-norm [Gonon et al., 2024]. The path-lifting and its norm have already been used to derive guarantees of identifiability [Stock and Gribonval, 2023, Bona-Pellissier et al., 2022], characterizations of the training dynamics [Marcotte et al., 2023] and generalization bounds [Neyshabur et al., 2015, Gonon et al., 2024]. While these tools have long been limited to simple network architectures [Neyshabur et al., 2015, Kawaguchi et al., 2017, Bona-Pellissier et al., 2022, Stock and Gribonval, 2023], they were recently extended [Gonon et al., 2024] to modern architectures by including most of their standard ingredients with the exception of attention mechanisms. This extension notably covers ResNets, VGGs, U-nets, ReLU MobileNets, Inception nets or Alexnet. Moreover, Gonon et al. [2024] also showed that these extended tools could be leveraged theoretically by deriving new state-of-the-art generalization bounds based on path-norms.

The first contribution of this work is to introduce a natural (rescaling-invariant) metric based on the path-lifting, and to show that it indeed yields an upper bound for the distance of two realizations of a network. Specifically, denoting $\Phi(\theta)$ the path-lifting (a finite-dimensional vector whose definition will be recalled in Section 2) of the network parameters θ, we establish (Theorem 3.1) that for any $1 \leqslant q \leqslant \infty$, any input x, and network parameters θ, θ' with the same sign :

$$\|R_\theta(x) - R_{\theta'}(x)\|_q \leqslant \max\left(\|x\|_\infty, 1\right) \|\Phi(\theta) - \Phi(\theta')\|_1. \tag{2}$$

We call $d(\theta, \theta') := \|\Phi(\theta) - \Phi(\theta')\|_1$ the ℓ^1*-path-metric*, by analogy with the so-called ℓ^1*-path-norm* $\|\Phi(\theta)\|_1$, see *e.g.* Neyshabur et al. [2015], Barron and Klusowski [2019], Gonon et al. [2024].

Inequality (2) not only holds for the very same general model as in Gonon et al. [2024] that encompasses pooling, skip connections and so on, but is also invariant under neuron-wise rescaling symmetries, thanks to the intrinsic invariances of the path-lifting Φ, resolving the two problems mentioned above for previous bounds of this type (Equation (1)). Moreover, it also improves on Equation (1) in most cases (for example when there are at least two layers $L \geqslant 2$, and with inputs $\|x\|_\infty \geqslant 1$), see Appendix F for the curious reader.

More importantly, Equation (2) shows that distances in the uniform norm ($q = \infty$) over bounded domains, but also in weighted ℓ^q norm, between the functions R_θ and $R_{\theta'}$, are upper-bounded by a much simpler quantity: the ℓ^1-path-metric between θ and θ', that is, the ℓ^1-distance between the *finite-dimensional* vectors $\Phi(\theta)$ and $\Phi(\theta')$.

The proof of Equation (2), which we believe to be interesting in its own right, is the main theoretical contribution of this paper. The mapping $(\theta, x) \mapsto R_\theta(x)$ that takes parameters θ, an input x, and returns the output $R_\theta(x)$ of the associated ReLU network, is well-known to be piecewise affine in x [Arora et al., 2017, Theorem 2.1], but it is also piecewise polynomial in the coordinates of θ [Gonon et al., 2024, consequence of Lemma A.1][Bona-Pellissier et al., 2022, consequence of Propositions 1 and 2]. To the best of our knowledge, the proof of Equation (2) is the first to *practically leverage the idea of "adequately navigating" through the different regions in θ where the network is polynomial*, see Figure 1 for an illustration.

The second contribution is to shed the light on theoretical and practical consequences of (2). After showing that **the ℓ^1-path metric can be efficiently computed via two forward passes** in contexts such as network pruning or quantization, we use it to provide a **new pruning method invariant under rescaling symmetries**, and **a new generalization bound valid on modern networks**.

- *Pruning Algorithm based on the lifting* Φ. We provide a new pruning algorithm invariant under symmetries. Its accuracy matches that of the standard magnitude pruning method when applied to ResNets trained on Imagenet in the lottery ticket context [Frankle et al., 2020].
- *Generalization bound based on* Φ (Theorem 5.1). This is the second best bound valid in such a general framework, after the one established in Gonon et al. [2024] (see Table 3). It is derived with a different proof compared to the one in Gonon et al. [2024], offering a distinct avenue for future refinements.

Plan. Section 2 recalls the model that captures standard ingredients of modern (ReLU, maxpool, skip connections etc.) networks, using the mathematical formalization given in Gonon et al. [2024] (that generalizes previous definitions given in Neyshabur et al. [2015], Kawaguchi et al. [2017], Bona-Pellissier et al. [2022], Stock and Gribonval [2023]). Section 2 also recalls the definitions of the path-lifting and the path-activations [Gonon et al., 2024]. The main result, Theorem 3.1, establishing

Equation (2) is proved in Section 3. This leads to a pruning method invariant to rescaling in Section 4 and to a new generalization bound in Section 5.

2 Model, path-lifting and path-activations

Model. The neural network model we consider generalizes and unifies several models from the literature, including those from Neyshabur et al. [2015], Kawaguchi et al. [2017], DeVore et al. [2021], Bona-Pellissier et al. [2022], Stock and Gribonval [2023], as detailed in Gonon et al. [2024, Definition 2.2]. This model allows for any Directed Acyclic Graph (DAG) structure incorporating standard features[1] such as max-pooling, average-pooling, skip connections, convolutional layers, and batch normalization layers, thus covering modern networks like ResNets, VGGs, AlexNet, etc. The full and formal definition of the model is in Appendix A.

Parameters and realization. All network parameters (weights and biases) are gathered in a parameter vector θ, and we denote $R_\theta(x)$ the output of the network when evaluated at input x (the function $x \mapsto R_\theta(x)$ is the so-called *realization* of the network with parameters θ).

Path-lifting Φ and path-activations A. For network parameters θ and input x, this paper considers the path-lifting vector $\Phi(\theta)$ and the path-activations matrix $A(\theta, x)$ as defined in Gonon et al. [2024, Definition A.1] for such general networks. We now give a simple description of these objects that will be sufficient to grasp the main results of this paper. The full definitions are recalled in Appendix A.

The vector $\Phi(\theta) \in \mathbb{R}^{\mathcal{P}}$ is indexed by the set $\mathcal{P}$ of *paths* of the network (hence the name path-lifting), where a path is a sequence of connected nodes (neurons) starting at some neuron (an input neuron in the case of networks without biases) and ending at an output neuron. For instance, in the case of a simple one-hidden-layer ReLU network, $p = u \to v \to w$ is an admissible path if u is an input neuron, v is a hidden neuron, and w is an output neuron. The coordinate of $\Phi(\theta)$ associated with a path is the product of the weights along this path, ignoring the non-linearities. For instance, if $\theta^{a\to b}$ denotes the weight of the edge $a \to b$, we have $\Phi_p(\theta) := \theta^{u\to v}\theta^{v\to w}$ for the path $p = u \to v \to w$.

The information about non-linearities is stored in binary form ($\{0, 1\}$) in the path-activations matrix $A(\theta, x) \in \{0, 1\}^{\mathcal{P}\times d_{\text{in}}}$ indexed by the paths p and the input coordinates u: $(A(\theta, x))_{p,u} := 1$ if and only all neurons along path p are activated and p starts at the input neuron u.

In networks with biases, the definitions are similar, but the set of paths $\mathcal{P}$ also includes paths starting from hidden neurons and ending at output neurons. The matrix A is then indexed by an additional input coordinate to account for biases, resulting in $A(\theta, x) \in \{0, 1\}^{\mathcal{P}\times(d_{\text{in}}+1)}$.

Key properties of (Φ, A). The essential properties are

- $\Phi(\theta)$ is a vector, which entries are mononomial functions of the coordinates of θ;
- $A(\theta, x)$ is a binary matrix, and is a piecewise constant function of (θ, x),
- both $\Phi(\theta)$ and $A(\theta, x)$ are invariant under neuron-wise rescalings of θ that leave invariant R_θ,
- the network output is a simple function of these two objects: for scalar-valued networks it holds

$$R_\theta(x) = \left\langle \Phi(\theta), A(\theta, x) \begin{pmatrix} x \\ 1 \end{pmatrix} \right\rangle \tag{3}$$

and a similar simple formula holds for vector-valued networks [Gonon et al., 2024, Theorem A.1].

Example: For a simple one-hidden-layer network with parameters $\theta = (u_1, \dots, u_k, v_1, \dots, v_k)$ with $u_i \in \mathbb{R}^{d_{\text{in}}}$, $v_i \in \mathbb{R}^{d_{\text{out}}}$ and associated function $R_\theta(x) = \sum_{i=1}^{k} \max(0, \langle x, u_i\rangle)v_i \in \mathbb{R}^{d_{\text{out}}}$, the path-lifting is simply given by $\Phi(\theta) = (u_i v_i^T, i \in [\![1, k]\!]) \in \mathbb{R}^{kd_{\text{in}}d_{\text{out}}}$ (flattened).

The path-activation matrix $A(\theta, x) \in \mathbb{R}^{kd_{\text{in}}d_{\text{out}}\times(d_{\text{in}}+1)}$ is simply $\mathbf{I}_{d_{\text{in}}} \otimes (\mathbb{1}_{\langle x,u_i\rangle>0})_{i\in[\![1,k]\!]} \otimes \mathbf{1}_{d_{\text{out}}}$, concatenated with $\mathbf{0}_{kd_{\text{in}}d_{\text{out}}}$ (zeros because there are no biases here). We denote by $\mathbf{I}_d$ the identity matrix of size $d \times d$ and $\mathbf{1}_d$ (resp. $\mathbf{0}_d$) the column vector of size d filled with ones (resp. zeros).

For this simple example, it is easy to see that both $\Phi(\theta)$ and $A(\theta, x)$ are invariant under the neuron-wise rescaling $\theta \mapsto \lambda \cdot \theta$ corresponding to $(v_i, u_i) \to (\frac{1}{\lambda_i}v_i, \lambda_i u_i)$ with $\lambda \in (R_{>0})^k$, that leaves invariant the associated function: $R_\theta = R_{\lambda\cdot\theta}$ [Gonon et al., 2024].

[1] With the exception of the attention mechanism.

3 Bounding function distances via the path-lifting

Consider our initial problem of finding a pseudo-metric $d(\theta, \theta')$ and a constant C_x for any input x, such that for many parameters θ, θ', it holds that

$$\|R_\theta(x) - R_{\theta'}(x)\|_\infty \leqslant C_x d(\theta, \theta').$$

Since the left hand-side is invariant under rescaling symmetries, the pseudo-metric d should ideally also maintain this invariance. Yet, pseudo-metrics based on norms like $\|\theta - \theta'\|$ are not invariant under rescaling and can even be made arbitrarily large by adversarially rescaling one of the parameters. In such cases, the bound becomes vacuous because the left-hand side remains unchanged while the right-hand side can grow arbitrarily large depending on the scaling of the parameters. Although one could *make such a bound invariant* by considering the infimum over all possible rescaling symmetries, this infimum may be difficult to compute in practice. Therefore, a "good" bound should ideally be both invariant under rescaling symmetries and easy to compute.

A rescaling invariant bound using the ℓ^1-path metric. Our main result, Theorem 3.1, precisely proves that we can define a pseudo-distance (the ℓ^1-path metric) via Φ as $d(\theta, \theta') := \|\Phi(\theta) - \Phi(\theta')\|_1$, with $C_x = \max(\|x\|_\infty, 1)$. Because of the invariances of Φ, any pseudo-distance that can be written in terms of a pseudo-distance between the images of Φ is automatically invariant under rescaling symmetries. The proof is in Appendix C (where we actually prove something slightly stronger, but we stick to the next theorem for simplicity).

Theorem 3.1. *Consider an exponent $q \in [1, \infty)$ and a ReLU neural network on a general DAG network with max-pool etc. as in Section 2 (see Definition A.2 in the appendix for a precise definition). Consider parameters vectors θ, θ'. If for every coordinate i, it holds $\theta_i \theta'_i \geqslant 0$, then for every input x:*

$$\|R_\theta(x) - R_{\theta'}(x)\|_q \leqslant \max(\|x\|_\infty, 1) \|\Phi(\theta) - \Phi(\theta')\|_1. \tag{4}$$

Moreover, for every such neural network architecture, there are parameters $\theta \neq \theta'$ and an input x such that Equation (4) is an equality.

The assumption of parameters with the same sign cannot be simply removed: see Figure 5 in Appendix C for a counterexample.

Computation of the path-metrics in two forward passes. Besides its theoretical interest, the proposed pseudo-distance also has the desirable property of being easily computable in practice. Consider θ, θ' such that $\theta_i \theta'_i \geqslant 0$ and $|\theta'_i| \leqslant |\theta_i|$ for every coordinate i. Then $\|\Phi(\theta) - \Phi(\theta')\|_1$ can be computed in two forward passes. Consider the graph $\tilde{G}$ deduced from the considered one but with max-pooling activations replaced by the identity. For a vector α, denote by $|\alpha|$ the vector deduced from α by applying $x \mapsto |x|$ coordinate-wise. Denote by $\mathbf{1}$ the input full of ones. We have:

$$\|\Phi(\theta) - \Phi(\theta')\|_1 = \|\Phi(\theta)\|_1 - \|\Phi(\theta')\|_1 = \|R^{\tilde{G}}_{|\theta|}(\mathbf{1})\|_1 - \|R^{\tilde{G}}_{|\theta'|}(\mathbf{1})\|_1 = \|R^{\tilde{G}}_{|\theta|}(\mathbf{1}) - R^{\tilde{G}}_{|\theta'|}(\mathbf{1})\|_1 \tag{5}$$

where $R^{\tilde{G}}$ denotes the forward pass in the network with graph $\tilde{G}$. The proof is in Appendix B and it is heavily based on Theorem A.1 in Gonon et al. [2024]. In particular, Equation (5) is true as soon as θ' is obtained from θ by pruning, or by quantizing/truncating towards zero.

Proof sketch of Theorem 3.1 The proof is given in Appendix C. We now give a sketch of it. The output of a ReLU neuron in the layer d of a layered fully-connected network is a piecewise polynomial function of the parameters θ of degree at most d [Gonon et al., 2024, consequence of Lemma A.1][Bona-Pellissier et al., 2022, consequence of Propositions 1 and 2].

Given an input x, the proof of Theorem 3.1 consists in defining a trajectory $t \in [0, 1] \to \theta(t) \in \Theta$ (**red** curve in Figure 1) that starts at θ, ends at θ', and with finitely many breakpoints $0 = t_0 < t_1 < \cdots < t_m = 1$ such that the path-activations $A(\theta(t), x)$ are constant on the open intervals $t \in (t_k, t_{k+1})$. Each breakpoint corresponds to a value where the activation of at least one path (hence at least one neuron) changes in the neighborhood of $\theta(t)$. For instance, in the left part of Figure 1, the straight **green** line (resp. quadratic **green** curve) corresponds to a change of activation of a ReLU neuron (for a given input x to the network) in the first (resp. second) layer.

With such a trajectory, given the key property (3), each quantity $|R_{\theta(t_k)}(x) - R_{\theta(t_{k+1})}(x)|$ can be controlled in terms of $\|\Phi(\theta(t_k)) - \Phi(\theta(t_{k+1}))\|_1$, and if the path is "nice enough", then this control can be extended globally from t_0 to t_m.

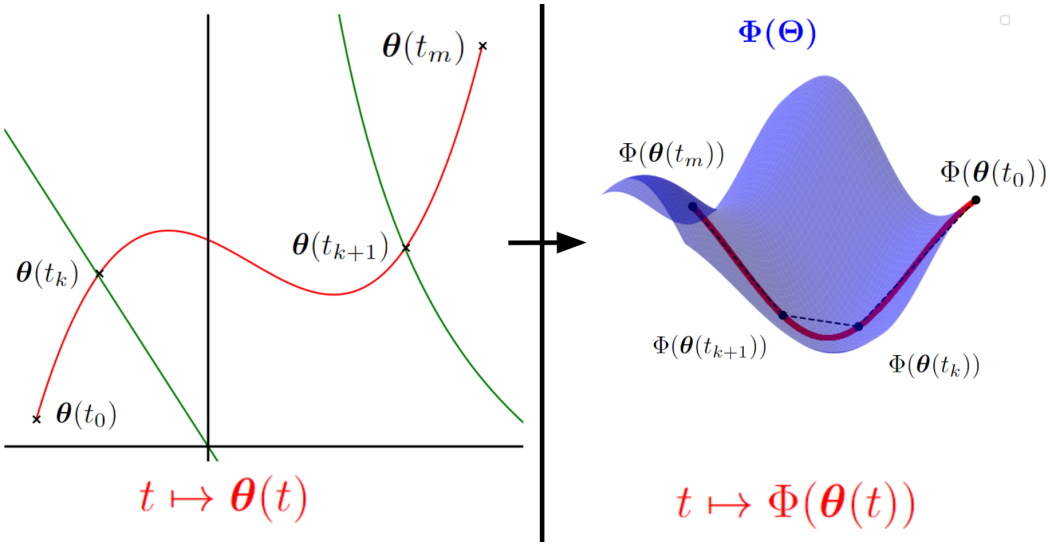


Figure 1: Illustration of the proof of Theorem 3.1, see the end of Section 3 for an explanation.

There are two obstacles: 1) proving that there are finitely many breakpoints t_k as above (think of $t \mapsto t^{n+2}\sin(1/t)$ that is n-times continuously differentiable but still crosses $t = 0$ an infinite number of times around zero), and 2) proving that the length $\sum_{k=1}^{m} \|\Phi(\theta(t_k)) - \Phi(\theta(t_{k+1}))\|_1$ of the broken line with vertices $\Phi(\theta(t_k))$ (dashed line on the right part of Figure 1) is bounded from above by $\|\Phi(\theta) - \Phi(\theta')\|_1$ times a reasonable factor. Trajectories satisfying these two properties are called "admissible" trajectories. The first property is true as soon as the trajectory $t \mapsto \theta(t)$ is smooth enough (analytic, say). The second is true *with factor one* thanks to a monotonicity property of the chosen trajectory. The core of the proof consists in exhibiting a trajectory with these properties.

4 Rescaling-invariant pruning matching the performance of IMP

As a proof of concept, we now show how to exploit the bound of Theorem 3.1 to design a rescaling-invariant pruning criterion that matches the accuracy of the widely used magnitude pruning criterion.

Notion of pruned parameter. Considering a DAG neural network G as described in Section 2, we use the shorthand $\mathbb{R}^G$ to denote the corresponding set of parameters (see Definition A.2 for a precise definition). By definition, a pruned version θ' of $\theta \in \mathbb{R}^G$ is a "Hadamard" product $\theta' = s \odot \theta$, where $s \in \mathbb{R}^G$ is a binary vector with all of its coordinates in $\{0, 1\}$ and $\|s\|_0$ is "small". A standard pruning method consists in selecting s from a pre-trained parameter θ typically by pruning out (setting to zero) entries of θ with magnitude below some threshold. This is clearly not rescaling-invariant, as the ranking of the magnitude of certain coefficients can change when applying certain rescalings.

4.1 Proposed rescaling-invariant pruning criteria

Given any θ, the parameters θ, θ' satisfy the assumptions of Theorem 3.1, hence for all input x we have $|R_\theta(x) - R'_\theta(x)| \leqslant \|\Phi(\theta) - \Phi(\theta')\|_1 \max(1, \|x\|_\infty)$. Defining $\Delta(\theta, s) := \|\Phi(\theta) - \Phi(s \odot \theta)\|_1$, a first reasonable global pruning criterion is then to aim at solving the following problem

$$\min_{s : \|s\|_0 \leqslant k} \Delta(\theta, s). \tag{6}$$

However, while (5) guarantees that, given s, the cost $\Delta(\theta, s)$ is computed in two forward passes as

$$\Delta(\theta, s) = \|R^{\tilde{G}}_{|\theta|}(\mathbf{1}_d)\|_1 - \|R^{\tilde{G}}_{|s\odot\theta|}(\mathbf{1}_d)\|_1$$

(where $\tilde{G}$ is the same as the original one G but with max-pooling activations replaced by the identity), Problem (6) is combinatorial because the set $\{s : \|s\|_0 \leqslant k\}$ has a size that grows exponentially in k.

Instead, we propose to approximate the solution of Problem (6) by minimizing an upper-bound of $\Delta(\theta, s)$. For each *individual parameter coordinate* i, we define

$$\Phi\text{-}\texttt{Cost}(\theta, i) := \Delta(\theta, s_i) \tag{7}$$

where $s_i := \mathbf{1}_G - e_i$ with $\mathbf{1}_G \in \mathbb{R}^G$ the vector filled with ones and $e_i \in \mathbb{R}^G$ the i-th canonical vector.

Bounding $\Delta(\theta, s)$ ***using*** $\Phi\text{-}\texttt{Cost}(\theta, i), i \in G$**.** Consider any subset $I \subseteq G$ to be potentially pruned out. When $s = s(I) := \mathbf{1}_G - \mathbf{1}_I$ with $\mathbf{1}_I = \sum_{i \in I} e_i$, then for any enumeration i_j, $1 \leqslant j \leqslant |I|$ of elements in I, denoting $s_j := \mathbf{1}_G - \sum_{\ell=1}^{j} e_{i_\ell} = \mathbf{1}_G - \mathbf{1}_{\cup_{\ell=1}^{j} \{i_\ell\}}$ (and $s_0 := \mathbf{1}_G$) we have

$$\begin{aligned}
\Delta(\theta, s) \underset{(5)}{=} \|\Phi(\theta)\|_1 - \|\Phi(s \odot \theta)\|_1 &= \sum_{j=1}^{|I|} \|\Phi(s_{j-1} \odot \theta)\|_1 - \|\Phi(s_j \odot \theta)\|_1 \underset{(5)}{=} \sum_{j=1}^{|I|} \Delta(s_{j-1} \odot \theta, s_j) \\
&= \sum_{j=1}^{|I|} \Phi\text{-}\texttt{Cost}(s_{j-1} \odot \theta, i_j) \qquad (8) \\
&\leqslant \sum_{j=1}^{|I|} \Phi\text{-}\texttt{Cost}(\theta, i_j). \qquad (9)
\end{aligned}$$

Φ-Pruning Method. Instead of solving the combinatorial Problem (6), we propose to minimizing the upper-bound given in Inequality (9). This is achieved via simple *reverse* hard thresholding:

1. compute $\Phi\text{-}\texttt{Cost}(\theta, i)$ for all i (two forward passes per i via Equation (5));
2. given the targeted sparsity k, select the set I of cardinal $|G| - k$ containing the k indices corresponding to the *smallest values* of this cost.

By Inequality (9) and Theorem 3.1, the index set I thus selected is such that $\|s(I)\|_0 \leqslant k$ and

$$|R_\theta(x) - R_{s(I) \odot \theta}| \leqslant \left(\sum_{i \in I} \Phi\text{-}\texttt{Cost}(\theta, i) \right) \|(x, 1)\|_\infty. \tag{10}$$

To the best of our knowledge, this is the first practical network pruning method invariant under rescaling symmetries that is endowed with guarantees on modern networks.

4.2 Experiments: proof of concept

To validate the approach we train a dense ResNet-18 on ImageNet-1k with standard hyperparameters (Appendix D). We prune (set to zero) some weights of the trained dense model with one of the following method (recall the definition of Φ-costs in Equation (7)):

- *magnitude pruning (MP)*: *layerwise* pruning of $p\%$ of the *smallest* weights in absolute value,
- *Φ-layerwise pruning (Φ-LP)*: *layerwise* pruning of $p\%$ of the weights with the smallest Φ-costs,
- *Φ-global pruning (Φ-GP)*: *global* pruning of $p\%$ of the weights with the smallest Φ-costs.

Invariance under rescaling symmetries: MP versus Φ-pruning. A key difference between these pruning methods is that Φ-pruning is invariant to neuron-wise rescaling symmetries in ReLU networks, while MP is not. Consequently, unlike Φ-pruning, MP can be affected by adversarial rescalings of the weights before pruning. Figure 2 shows a concrete example. We do not even try to choose a rescaling in an adversarial manner: we simply *choose a rescaling at random* (see Appendix D for details), before applying magnitude pruning. This results in a significant drop in accuracy for MP.

Comparing criteria and masks. The left part of Figure 3 shows the magnitude (absolute value) versus the Φ-cost for each parameter index i, illustrating a positive correlation between these two quantities. This leads to a high overlap in the sets of pruned weights, as shown in Table 1. This suggests that the parameters obtained with SGD are naturally balanced in some sense, given that magnitude pruning, which is not invariant under rescaling, largely aligns with the rescaling-invariant pruning methods. The right part of Figure 3 shows that this correlation is largely reduced after applying a random rescale as detailed in Appendix D. In this case, as we have seen on Figure 2, magnitude pruning indeed yields quite different results compared to Φ-pruning.

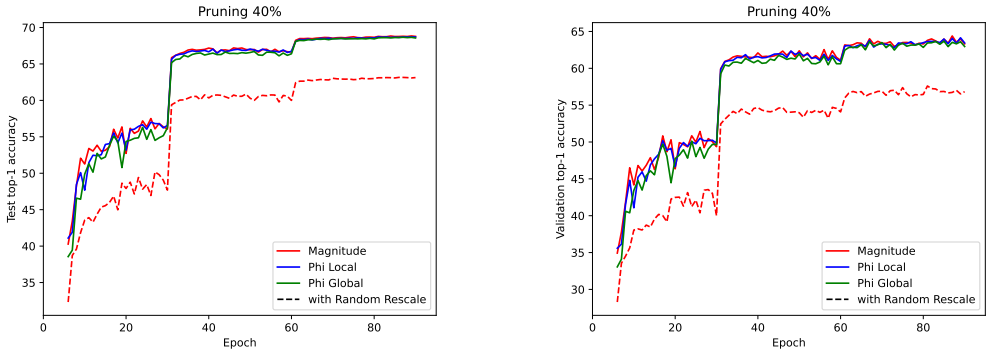


Figure 2: Training Curves: Test Top-1 Accuracy (left) and Validation Top-1 Accuracy (right) when finetuning the pruned models, with (dashed line) or without (plain line) rescaling. The results for Φ-pruning are the same with or without rescaling, hence the corresponding dashed line (random rescale applied beforehand) perfectly overlaps with the plain line (no rescale), unlike for MP.

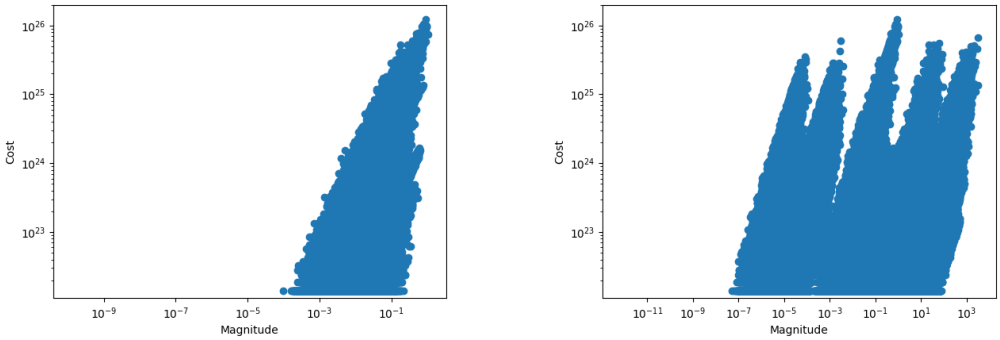


Figure 3: Scatter plot of the magnitude of each weight, versus its Φ-cost (Equation (7)). Left: dense model trained with SGD. Right: same but randomly rescaled as detailed in Appendix D.

Comparing accuracies. We find that Φ-pruning methods achieve accuracies comparable to the standard magnitude pruning method. Specifically, both methods yield the same top-1 test accuracy at the end of training (Table 2), and their training curves are similar (Figure 2). This is noteworthy because, in this context, the choice of the pruning mask is crucial for achieving high accuracy, as demonstrated by the magnitude pruning method applied to a random rescaling of the parameters, which significantly underperforms (Table 2 and Figure 2).

Computation of the Φ-costs. Computationally speaking, the time needed to compute the Φ-costs associated with all the parameters of the trained dense ResNet18 using a single V100 GPU is equivalent to the training time needed to obtain this ResNet18. Given the gain in rescaling invariance, a natural challenge is to speed this up: besides parallelization over multiple GPUs, tricks allow to jointly compute all the costs at least for the last layer (one of the most costly one), and will be the object of future investigations. Our experiments also show (cf the vertical axis of Figure 3) that the bound (10) is currently vacuous (on the order of 10^{26}). This aligns with the numerical observations reported in Gonon et al. [2024] for the ℓ^1-path-norm $\|\Phi(\theta)\|_1$), and obtaining tightened, non-vacuous bounds, e.g. with ℓ^q-path metrics with $q > 1$ (known to be of a much smaller order of magnitude [Gonon et al., 2024]) is a challenge left to future work.

Overall, Φ-pruning is competitive with respect to the widespread magnitude pruning method in terms of accuracy, is inherently invariant to rescaling symmetries, and the emerging theory of Φ offers a promising foundation for future theoretical analysis of these pruning methods. We hope these results will inspire further research in this direction.

It is important to highlight that these results were achieved without any tuning effort: we used the same hyperparameters for the new Φ-pruning methods as those commonly employed for magnitude pruning in comparable situations [Frankle et al., 2021]. For further details, see Appendix D.

Pruning level	10%	20%	40%	60%	80%
Overlap between MP and Φ-GP	70%	74%	76%	80%	86%
Overlap between MP and Φ-LP	87%	82%	90%	94%	96%

Table 1: Overlap between masks as a function of pruning level (percentage of pruned out coefficients). If S_1 and S_2 index the weights pruned (i.e. set to zero) by methods 1 and 2, the overlap is computed as $100 \times |S_1 \cap S_2|/|S_1|$. As all methods prune the same amount of weights we have $|S_1| = |S_2|$.

Pruning level	none	10%	20%	40%	60%	80%
MP (+ *Random Rescale*)	67.7%	69.0 *(68.8)*	69.0 *(68.7)*	68.8 *(63.1)*	68.2 *(57.5)*	66.5 *(15.8)*
Φ-LP $(*)$		68.8	68.9	68.7	68.1	66.1
Φ-GP $(*)$		68.6	68.8	68.6	67.9	66.0

Table 2: Top-1 accuracy after pruning, rewind and retrain, as a function of the pruning level.
$(*)$ = results valid with as well as without rescaling, as Φ-pruning is invariant to rescaling.
MP + Random Rescale corresponds to the case where we apply a random rescaling before applying MP (see Appendix D for details).

5 Application to generalization

We also concretely demonstrate how Theorem 3.1 can be used to derive generalization guarantees. We take for granted the classical notion of loss function, generalization error and weight-sharing, and we refer the reader to Appendix G for formal definitions.

Theorem 5.1. *Consider n training samples stored in a vector $\mathbf{Z}$. Consider an upper bound $B \geqslant 1$ on the ℓ^∞-norm of the inputs. Consider a network with depth D (max length of a path from input to output neurons), output dimension d_{out}. Denote by #params the number of parameters, without redundancy when there is weight-sharing[2]. Assume the loss function $\ell(y, y')$ to be L-Lipschitz in y' for every output y. It holds for every parameters θ learned on $\mathbf{Z}$:*

$$\mathbb{E}_{\mathbf{Z}}\, \ell\text{-generalization error of } \theta \leqslant 544 \frac{LB}{\sqrt{n}} \max(D, d_{out}) \sqrt{\#params} \times \|\Phi(\theta)\|_1. \tag{11}$$

The full proof of Theorem 5.1 is in Appendix G. Theorem 5.1 is the second-best generalization bound based on the so-called ℓ^1-path-norm $\|\Phi(\theta)\|_1$ that is valid on a model as general as the one described in Section 2: see Table 3 for a comparison. The interests of Theorem 5.1 compared to the better bound proved in Gonon et al. [2024] is (i) to illustrate that Theorem 3.1 can indeed be used to provide generalization guarantees and (ii) to provide an alternative avenue for future refinements as it is derived with a different proof.

Limitations. However, note that both the bound in Gonon et al. [2024] and Theorem 5.1 are essentially independent[3] of the input distribution, and because of that, they have to be vacuous in modern over-parameterized training regimes where it is possible to achieve zero training error [Zhang et al., 2021, Nagarajan and Kolter, 2019]. We hope these bounds will inspire new bounds *formulated in terms of* Φ (to preserve symmetries) but with stronger dependencies on the input distribution.

We now explain a sketch of the proof of Theorem 5.1 along with its key differences compared to the proof of the concurrent bound in Gonon et al. [2024].

Sketch of proof of Theorem 5.1 and key differences with the proof of the bound in Gonon et al. [2024]. Both proofs are based on the Rademacher complexity. The one in Gonon et al. [2024] bounds the Rademacher complexity by peeling one by one every neuron of the network as in Golowich et al. [2018]. Here, Theorem 5.1 starts by reducing the problem of bounding the Rademacher complexity to a covering problem using the classical Dudley's inequality, which is a common argument already used in the literature to establish generalization bounds [Bartlett et al., 2017]. The new problem is then to cover a set Θ of parameters θ with a finite number of balls with respect to a (pseudo-)metric of the type $d(\theta, \theta') = \|R_\theta - R_{\theta'}\|$, where $\|\cdot\|$ should be understood as a norm whose precise definition

[2]For instance, for a convolutional layer with kernel matrix K, this is the number of coefficients in K, not the number of coefficients in the matrix corresponding to the linear transformation with this convolutional kernel (which would contain many repetitions of the coefficients in K).

[3]Except for the constant B, but this dependence is too weak to make the bound informative in over-parameterized regimes.

is of no relevance at this point. This is where the new Theorem 3.1 plays a crucial role: such a cover of Θ for d can be otbained by covering $\Phi(\Theta)$ with respect to the ℓ^1-norm. Exhibiting ℓ^1-coverings of $\Phi(\Theta)$ is easier due to the finite dimensionality of the path-lifting Φ. However, using standard bounds for covering $\Phi(\Theta)$ results in an undesirable dependence on the ambient dimension of $\Phi(\Theta)$. This dimension, determined by the number of paths, is exponentially larger than the number of parameters in Θ. While little is known on the image $\Phi(\Theta)$, recent findings show that locally, $\Phi(\Theta)$ has as expected a dimension bounded by the number of parameters [Bona-Pellissier et al., 2022, Theorem 7]. In the same vein, we prove in the appendix (Theorem H.1) that it is also possible to replace the number of paths (the algebraic ambient dimension of $\Phi(\Theta)$) that would appear using standard coverings, by the dimension of $\Phi(\Theta)$ as a variety: the number of parameters without weigh-sharing redundancies, minus the number of neuron-wise rescaling symmetries. Thus, this new proof heavily relies on two new properties of the path-lifting: the new Theorem 3.1 and new coverings of $\Phi(\Theta)$, opening up new avenues to strengthen current generalization bounds.

Table 3: Generalization bounds (up to universal multiplicative constants) for a ReLU network estimator learned from n iid training points when 1) the loss $\hat{y} \in (\mathbb{R}^{d_{\text{out}}}, \|\cdot\|_2) \mapsto \ell(\hat{y}, y) \in \mathbb{R}$ is L-Lipschitz for every y, and 2) inputs are bounded in L^∞-norm by $B \geqslant 1$. Here, $d_{\text{in}}/d_{\text{out}}$ are the input/output dimensions, #params is the number of parameters without redundancy when there is weight-sharing, $K = \max_{v \in N_{*\text{-pool}}} |\operatorname{ant}(v)|$ is the maximum kernel size (see Definition A.2 in the appendix) of the $*$-max-pooling neurons, M_d is the matrix of layer d for a *layered fully-connected* network (LFCN) without bias $R_\theta(x) = M_D \operatorname{ReLU}(M_{D-1} \dots \operatorname{ReLU}(M_1 x))$, D is the depth.

	Architecture	Generalization bound
[Kakade et al., 2008, Eq. (5)] [Bach, 2024, Sec. 4.5.3]	LFCN with depth $D = 1$, no bias, $d_{\text{out}} = 1$ (linear regression)	$\frac{LB}{\sqrt{n}} \times \|\Phi(\theta)\|_1 \sqrt{\ln(d_{\text{in}})}$
[E et al., 2022, Thm. 6] [Bach, 2017, Proposition 7]	LFCN with $D = 2$, no bias, $d_{\text{out}} = 1$ (two-layer network)	$\frac{LB}{\sqrt{n}} \times \|\Phi(\theta)\|_1 \sqrt{\ln(d_{\text{in}})}$
[Neyshabur et al., 2015, Corollary 7]	DAG, no bias, $d_{\text{out}} = 1$	$\frac{LB}{\sqrt{n}} \times \|\Phi(\theta)\|_1 2^D \sqrt{\ln(d_{\text{in}})}$
[Golowich et al., 2018, Theorem 3.2]	LFCN with arbitrary D, no bias, $d_{\text{out}} = 1$	$\frac{LB}{\sqrt{n}} \times \prod_{d=1}^{D} \|M_d\|_{1,\infty} \sqrt{D + \ln(d_{\text{in}})}$
[Barron and Klusowski, 2019, Corollary 2]	LFCN with arbitrary D, no bias, $d_{\text{out}} = 1$	$\frac{LB}{\sqrt{n}} \times \|\Phi(\theta)\|_1 \sqrt{D + \ln(d_{\text{in}})}$
[Gonon et al., 2024]	DAG, *with biases*, arbitrary d_{out}, with ReLU, identity and k-max-pooling neurons for $k \in \{k_1, \dots, k_P\} \subset \{1, \dots, K\}$	$\frac{LB}{\sqrt{n}} \times \|\Phi(\theta)\|_1 \sqrt{D \ln(PK) + \ln(d_{\text{in}} d_{\text{out}})}$
This work, Theorem 5.1	DAG, with biases, arbitrary d_{out}, with ReLU, identity and $*$-max-pooling neurons	$\frac{LB}{\sqrt{n}} \times \|\Phi(\theta)\|_1 \max(D, d_{\text{out}}) \sqrt{\#\text{params}}$

6 Conclusion

This work proves that for modern ReLU networks with max pooling and/or skip connections, the ℓ^1-path-metric $d(\theta, \theta') = \|\Phi(\theta) - \Phi(\theta')\|_1$ bounds from above the distance between functions realized by the networks with parameters θ and θ'. This metric, which is invariant to the natural rescalings associated to the network parameterization, can be easily computed in two forward passes in the context of parameter pruning or quantization. Besides a new generalization bound, it leads to a pruning algorithm invariant under rescaling, competitive with standard magnitude pruning in terms of accuracy, and with an associated theoretical bound (10), which is the first of its kind to the best of our knowledge. A natural challenge is to establish similar but sharper bounds, typically with metrics still based on the path-lifting but using ℓ^p norms with $p > 1$, and/or metrics that provide functional bounds in expectation over inputs x with a given probability distribution. Progress in this direction is needed to obtain non-vacuous pruning, quantization and generalization bounds, which may also leverage recent advances in PAC-Bayes generalization bounds [Hellström et al., 2023].

Acknowledgments

This work was supported in part by the AllegroAssai ANR-19-CHIA-0009, by the NuSCAP ANR-20-CE48-0014 projects of the French Agence Nationale de la Recherche and by the SHARP ANR project ANR-23-PEIA-0008 in the context of the France 2030 program.

The authors thank the Blaise Pascal Center for the computational means. It uses the SIDUS [Quemener and Corvellec, 2013] solution developed by Emmanuel Quemener.

References

Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. *Electron. Colloquium Comput. Complex.*, 24:98, 2017. URL https://eccc.weizmann.ac.il/report/2017/098.

Francis Bach. Learning from first principles, 2024. URL https://www.di.ens.fr/~fbach/ltfp_book.pdf.

Francis R. Bach. Breaking the curse of dimensionality with convex neural networks. *J. Mach. Learn. Res.*, 18:19:1–19:53, 2017. URL http://jmlr.org/papers/v18/14-546.html.

Andrew R. Barron and Jason M. Klusowski. Complexity, statistical risk, and metric entropy of deep nets using total path variation. *CoRR*, abs/1902.00800, 2019. URL http://arxiv.org/abs/1902.00800.

Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, 2002. URL http://jmlr.org/papers/v3/bartlett02a.html.

Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6240–6249, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/b22b257ad0519d4500539da3c8bcf4dd-Abstract.html.

Julius Berner, Philipp Grohs, and Arnulf Jentzen. Analysis of the generalization error: Empirical risk minimization over deep artificial neural networks overcomes the curse of dimensionality in the numerical approximation of black-scholes partial differential equations. *SIAM J. Math. Data Sci.*, 2(3):631–657, 2020. doi: 10.1137/19M125649X. URL https://doi.org/10.1137/19M125649X.

Joachim Bona-Pellissier, François Malgouyres, and François Bachoc. Local identifiability of deep relu neural networks: the theory. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b0ae046e198a5e43141519868a959c74-Abstract-Conference.html.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009. ISBN 978-0-262-03384-8. URL http://mitpress.mit.edu/books/introduction-algorithms.

Ronald A. DeVore, Boris Hanin, and Guergana Petrova. Neural network approximation. *Acta Numer.*, 30:327–444, 2021. doi: 10.1017/S0962492921000052. URL https://doi.org/10.1017/S0962492921000052.

Weinan E, Chao Ma, and Lei Wu. The Barron space and the flow-induced function spaces for neural network models. *Constr. Approx.*, 55(1):369–406, 2022. ISSN 0176-4276. doi: 10.1007/s00365-021-09549-y. URL https://doi-org.acces.bibliotheque-diderot.fr/10.1007/s00365-021-09549-y.

Jonathan Frankle, David J. Schwab, and Ari S. Morcos. The early phase of neural network training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=Hkl1iRNFwS.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M. Roy, and Michael Carbin. Pruning neural networks at initialization: Why are we missing the mark? In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=Ig-VyQc-MLK.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018*, volume 75 of *Proceedings of Machine Learning Research*, pages 297–299. PMLR, 2018. URL http://proceedings.mlr.press/v75/golowich18a.html.

Antoine Gonon, Nicolas Brisebarre, Rémi Gribonval, and Elisa Riccietti. Approximation speed of quantized versus unquantized relu neural networks and beyond. *IEEE Trans. Inf. Theory*, 69(6): 3960–3977, 2023. doi: 10.1109/TIT.2023.3240360. URL https://doi.org/10.1109/TIT.2023.3240360.

Antoine Gonon, Nicolas Brisebarre, Elisa Riccietti, and Rémi Gribonval. A path-norm toolkit for modern networks: consequences, promises and challenges. In *International Conference on Learning Representations, ICLR 2024 Spotlight, Vienna, Austria, May 7-11*. OpenReview.net, 2024. URL https://openreview.net/pdf?id=hiHZVUIYik.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Fredrik Hellström, Giuseppe Durisi, Benjamin Guedj, and Maxim Raginsky. Generalization bounds: Perspectives from information theory and pac-bayes. *CoRR*, abs/2309.04381, 2023. doi: 10.48550/ARXIV.2309.04381. URL https://doi.org/10.48550/arXiv.2309.04381.

Sham M. Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 793–800. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2008/hash/5b69b9cb83065d403869739ae7f0995e-Abstract.html.

Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. Generalization in deep learning. *CoRR*, abs/1710.05468, 2017. URL http://arxiv.org/abs/1710.05468.

Michel Ledoux and Michel Talagrand. *Probability in Banach spaces*, volume 23 of *Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)]*. Springer-Verlag, Berlin, 1991. ISBN 3-540-52013-9. doi: 10.1007/978-3-642-20212-4. URL https://doi.org/10.1007/978-3-642-20212-4. Isoperimetry and processes.

Sibylle Marcotte, Rémi Gribonval, and Gabriel Peyré. Abide by the law and follow the flow: Conservation laws for gradient flows. *CoRR*, abs/2307.00144, 2023. doi: 10.48550/arXiv.2307.00144. URL https://doi.org/10.48550/arXiv.2307.00144.

Andreas Maurer. A vector-contraction inequality for rademacher complexities. In Ronald Ortner, Hans Ulrich Simon, and Sandra Zilles, editors, *Algorithmic Learning Theory - 27th International Conference, ALT 2016, Bari, Italy, October 19-21, 2016, Proceedings*, volume 9925 of *Lecture Notes in Computer Science*, pages 3–17, 2016. doi: 10.1007/978-3-319-46379-7_1. URL https://doi.org/10.1007/978-3-319-46379-7_1.

Vaishnavh Nagarajan and J. Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11611–11622, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/05e97c207235d63ceb1db43c60db7bbb-Abstract.html.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Workshop and Conference Proceedings*, pages 1376–1401. JMLR.org, 2015. URL http://proceedings.mlr.press/v40/Neyshabur15.html.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=Skz_WfbCZ.

E. Quemener and M. Corvellec. SIDUS—the Solution for Extreme Deduplication of an Operating System. *Linux Journal*, 2013.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014. ISBN 978-1-10-705713-5. URL http://www.cambridge.org/de/academic/subjects/computer-science/pattern-recognition-and-machine-learning/understanding-machine-learning-theory-algorithms.

Pierre Stock and Rémi Gribonval. An embedding of ReLU networks and an analysis of their identifiability. *Constr. Approx.*, 57(2):853–899, 2023. ISSN 0176-4276,1432-0940. doi: 10.1007/s00365-022-09578-1. URL https://doi.org/10.1007/s00365-022-09578-1.

Ramon Van Handel. Probability in high dimension. *Lecture Notes (Princeton University)*, 2014. URL https://web.math.princeton.edu/~rvan/APC550.pdf. [Accessed: April 2024].

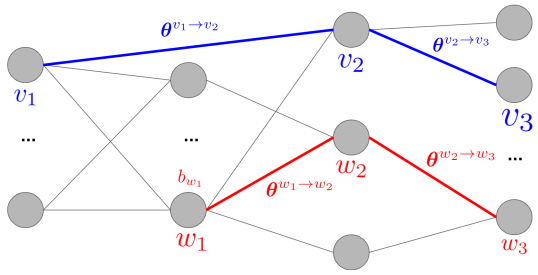
Martin J. Wainwright. *High-dimensional statistics*, volume 48 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 2019. ISBN 978-1-108-49802-9. doi: 10.1017/9781108627771. URL https://doi-org.acces.bibliotheque-diderot.fr/10.1017/9781108627771. A non-asymptotic viewpoint.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, 2021. doi: 10.1145/3446776. URL https://doi.org/10.1145/3446776.

Appendices

A Path-lifting and path-activations

This section recalls the definitions from Gonon et al. [2024] for completeness.



$$\boldsymbol{A}(\theta,x) = \begin{array}{c} \\ \mathcal{P}_I \left\{ \begin{array}{c} p \end{array} \right. \\ \\ \mathcal{P}_H \left\{ \begin{array}{c} p' \end{array} \right. \end{array} \begin{array}{c} \overbrace{\qquad\qquad\qquad}^{N_{\text{in}}} \\ v_1 \\ \cdots \\ 0 \;\ldots\; 0 \;\; a_p(\theta,x) \;\; 0 \;\ldots\; 0 \\ \cdots \\ \\ 0 \\ \\ \end{array} \left(\begin{array}{c} b \\ 0 \\ \\ \vdots \\ a_{p'}(\theta,x) \\ \vdots \end{array} \right)$$

Figure 4: The coordinate of the path-lifting Φ associated with the path $p = v_1 \to v_2 \to v_3$ is $\Phi_p(\theta) = \theta^{v_1\to v_2}\theta^{v_2\to v_3}$ since it starts from an input neuron (Definition A.5). While the path $p' = w_1 \to w_2 \to w_3$ starts from a hidden neuron (in $N \setminus (N_{\text{in}} \cup N_{\text{out}})$), so there is also the bias of w_1 to take into account: $\Phi_{p'}(\theta) = b_{w_1}\theta^{w_1\to w_2}\theta^{w_2\to w_3}$. As specified in Definition A.5, the columns of the path-activation matrix $\boldsymbol{A}$ are indexed by $N_{\text{in}} \cup \{b\}$ and its rows are indexed by $\mathcal{P} = \mathcal{P}_I \cup \mathcal{P}_H$, with $\mathcal{P}_I$ the set of paths in $\mathcal{P}$ starting from an input neuron, and $\mathcal{P}_H$ the set of paths starting from a hidden neuron.

Definition A.1 (ReLU and k-max-pooling activation functions)**.** *The ReLU function is defined as* $\mathrm{ReLU}(x) := x\mathbb{1}_{x\geqslant 0}$ *for* $x \in \mathbb{R}$. *The k-max-pooling function* $k\texttt{-pool}(x) := x_{(k)}$ *returns the k-th largest coordinate of* $x \in \mathbb{R}^d$.

Definition A.2 (ReLU neural network [Gonon et al., 2024])**.** *Consider a Directed Acyclic Graph (DAG)* $G = (N, E)$ *with edges E, and vertices N called neurons. For a neuron v, the sets* $\mathrm{ant}(v), \mathrm{suc}(v)$ *of antecedents and successors of v are* $ant(v) := \{u \in N, u \to v \in E\}, \mathrm{suc}(v) := \{u \in N, v \to u \in E\}$. *Neurons with no antecedents (resp. no successors) are called input (resp. output) neurons, and their set is denoted N_{in} (resp. N_{out}). Neurons in $N \setminus (N_{in} \cup N_{out})$ are called hidden neurons. Input and output dimensions are respectively* $d_{in} := |N_{in}|$ *and* $d_{out} := |N_{out}|$.

• *A* **ReLU neural network architecture** *is a tuple* $(G, (\rho_v)_{v\in N\setminus N_{in}})$ *composed of a DAG* $G = (N, E)$ *with attributes* $\rho_v \in \{\mathrm{id}, \mathrm{ReLU}\} \cup \{k\texttt{-pool}, k \in \mathbb{N}_{>0}\}$ *for* $v \in N \setminus (N_{out} \cup N_{in})$ *and* $\rho_v = \mathrm{id}$ *for* $v \in N_{out}$. *We will again denote the tuple* $(G, (\rho_v)_{v\in N\setminus N_{in}})$ *by G, and it will be clear from context whether the results depend only on* $G = (N, E)$ *or also on its attributes. Define* $N_\rho := \{v \in N, \rho_v = \rho\}$ *for an activation ρ, and* $N_{*\texttt{-pool}} := \cup_{k\in\mathbb{N}_{>0}} N_{k\texttt{-pool}}$. *A neuron in* $N_{*\texttt{-pool}}$ *is called a $*$-max-pooling neuron. For* $v \in N_{*\texttt{-pool}}$, *its kernel size is defined as being* $|\mathrm{ant}(v)|$.

• **Parameters** *associated with this architecture are vectors*[4] $\theta \in \mathbb{R}^G := \mathbb{R}^{E \cup N \setminus N_{in}}$. *We call bias* $b_v := \theta_v$ *the coordinate associated with a neuron* v *(input neurons have no bias), and denote* $\theta^{u \to v}$ *the weight associated with an edge* $u \to v \in E$. *We will often denote* $\theta^{\to v} := (\theta^{u \to v})_{u \in \mathrm{ant}(v)}$ *and* $\theta^{v \to} := (\theta^{u \to v})_{u \in \mathrm{suc}(v)}$.

• *The* **realization** *of a neural network with parameters* $\theta \in \mathbb{R}^G$ *is the function* $R_\theta^G : \mathbb{R}^{N_{in}} \to \mathbb{R}^{N_{out}}$ *(simply denoted* R_θ *when* G *is clear from the context) defined for every input* $x \in \mathbb{R}^{N_{in}}$ *as*

$$R_\theta(x) := (v(\theta, x))_{v \in N_{out}},$$

where we use the same symbol v *to denote a neuron* $v \in N$ *and the associated function* $v(\theta, x)$, *defined as* $v(\theta, x) := x_v$ *for an input neuron* v, *and defined by induction otherwise*

$$v(\theta, x) := \begin{cases} \rho_v(b_v + \sum_{u \in \mathrm{ant}(v)} u(\theta, x)\theta^{u \to v}) & \text{if } \rho_v = \mathrm{ReLU} \text{ or } \rho_v = \mathrm{id}, \\ k\text{-}\mathtt{pool}\left((b_v + u(\theta, x)\theta^{u \to v})_{u \in \mathrm{ant}(v)}\right) & \text{if } \rho_v = k\text{-}\mathtt{pool}. \end{cases} \tag{12}$$

Definition A.3 (Paths and depth in a DAG [Gonon et al., 2024])**.** *Consider a DAG* $G = (N, E)$ *as in Definition A.2. A path of* G *is any sequence of neurons* $v_0, \ldots, v_d$ *such that each* $v_i \to v_{i+1}$ *is an edge in* G. *Such a path is denoted* $p = v_0 \to \ldots \to v_d$. *This includes paths reduced to a single* $v \in N$, *denoted* $p = v$. *The* length *of a path is* $\mathtt{length}(p) = d$ *(the number of edges). We will denote* $p_\ell := v_\ell$ *the* ℓ*-th neuron for a general* $\ell \in \{0, \ldots, \mathtt{length}(p)\}$ *and use the shorthand* $p_{\mathtt{end}} = v_{\mathtt{length}(p)}$ *for the last neuron. The* depth of the graph G *is the maximum length over all of its paths. If* $v_{d+1} \in \mathrm{suc}(p_{\mathtt{end}})$ *then* $p \to v_{d+1}$ *denotes the path* $v_0 \to \ldots \to v_d \to v_{d+1}$. *We denote by* $\mathcal{P}^G$ *(or simply* $\mathcal{P}$*) the set of paths ending at an output neuron of* G.

Definition A.4 (Sub-graph ending at a given neuron)**.** *Given a neuron* v *of a DAG* G, *we denote* $G^{\to v}$ *the graph deduced from* G *by keeping only the largest subgraph with the same inputs as* G *and with* v *as a single output: every neuron* u *with no path to reach* v *through the edges of* G *is removed, as well as all its incoming and outcoming edges. We will use the shorthand* $\mathcal{P}^{\to v} := \mathcal{P}^{G^{\to v}}$ *to denote the set of paths in* G *ending at* v.

We now recall the definitions of the path-lifting and path-activations from Gonon et al. [2024]. An illustration can be found in Figure 4.

Definition A.5 (Path-lifting and path-activations [Gonon et al., 2024])**.** *Consider a ReLU neural network architecture* G *as in Definition A.2 and parameters* $\theta \in \mathbb{R}^G$ *associated with* G. *For* $p \in \mathcal{P}$, *define*

$$\Phi_p(\theta) := \begin{cases} \prod_{\ell=1}^{\mathtt{length}(p)} \theta^{v_{\ell-1} \to v_\ell} & \text{if } p_0 \in N_{in}, \\ b_{p_0} \prod_{\ell=1}^{\mathtt{length}(p)} \theta^{v_{\ell-1} \to v_\ell} & \text{otherwise}, \end{cases}$$

where an empty product is equal to 1 *by convention. The path-lifting* $\Phi^G(\theta)$ *of* θ *is*

$$\Phi^G(\theta) := (\Phi_p(\theta))_{p \in \mathcal{P}^G}.$$

This is often denoted Φ *when the graph* G *is clear from the context. We will use the shorthand* $\Phi^{\to v} := \Phi^{G^{\to v}}$ *to denote the path-lifting associated with* $G^{\to v}$ *(Definition A.4).*

Consider an input x *of* G. *The activation of an* edge $u \to v$ *on* (θ, x) *is defined to be* $a_{u \to v}(\theta, x) := 1$ *when* v *is an identity neuron;* $a_{u \to v}(\theta, x) := \mathbb{1}_{v(\theta,x) > 0}$ *when* v *is a ReLU neuron; and when* v *is a* k*-max-pooling neuron, define* $a_{u \to v}(\theta, x) := 1$ *if the neuron* u *is the first in* $\mathrm{ant}(v)$ *in lexicographic order to satisfy* $u(\theta, x) := k\text{-}\mathtt{pool}\left((w(\theta, x))_{w \in \mathrm{ant}(v)}\right)$ *and* $a_{u \to v}(\theta, x) := 0$ *otherwise. The activation of a* neuron v *on* (θ, x) *is defined to be* $a_v(\theta, x) := 1$ *if* v *is an input neuron, an identity neuron, or a* k*-max-pooling neuron, and* $a_v(\theta, x) := \mathbb{1}_{v(\theta,x) > 0}$ *if* v *is a ReLU neuron. We then define the activation of a* path $p \in \mathcal{P}$ *with respect to input* x *and parameters* θ *as:* $a_p(\theta, x) := a_{p_0}(\theta, x) \prod_{\ell=1}^{\mathtt{length}(p)} a_{v_{\ell-1} \to v_\ell}(\theta, x)$ *(with an empty product set to one by convention). Consider a new symbol* $v_{\mathtt{bias}}$ *that is not used for denoting neurons. The path-activations matrix* $\boldsymbol{A}(\theta, x)$ *is defined as the matrix in* $\mathbb{R}^{\mathcal{P} \times (N_{in} \cup \{v_{\mathtt{bias}}\})}$ *such that for any path* $p \in \mathcal{P}$ *and neuron* $u \in N_{in} \cup \{v_{\mathtt{bias}}\}$

$$(\boldsymbol{A}(\theta, x))_{p,u} := \begin{cases} a_p(\theta, x)\mathbb{1}_{p_0 = u} & \text{if } u \in N_{in}, \\ a_p(\theta, x) & \text{otherwise when } u = v_{\mathtt{bias}}. \end{cases}$$

[4] For an index set I, denote $\mathbb{R}^I = \{(\theta_i)_{i \in I}, \theta_i \in \mathbb{R}\}$.

B Computing the path-metric in two forward passes

This section proves that the path-metric can be computed in two forward passes.

Theorem B.1. *Consider an architecture $G = (N, E, (\rho_v)_{v \in N \setminus N_{in}})$ as in Definition A.2. Consider the architecture $\tilde{G} := (N, E, (\tilde{\rho}_v)_{v \in N \setminus N_{in}})$ with $\tilde{\rho}_v := \mathrm{id}$ if $v \in N_{*\texttt{-pool}}$, and $\tilde{\rho}_v := \rho_v$ otherwise (that is, replacing $*$-max-pooling neurons with identity ones). For a vector α, denote $|\alpha|$ the vector deduced from α by applying $x \mapsto |x|$ coordinate-wise. Denote by $\mathbf{1}$ the input full of ones. Consider parameters θ, θ' such that for every coordinate i, $\theta_i \theta'_i \geqslant 0$ and $|\theta'_i| \leqslant |\theta_i|$. It holds:*

$$\|\Phi(\theta) - \Phi(\theta')\|_1 = \|\Phi(\theta)\|_1 - \|\Phi(\theta')\|_1 = \|R^{\tilde{G}}_{|\theta|}(\mathbf{1})\|_1 - \|R^{\tilde{G}}_{|\theta'|}(\mathbf{1})\|_1 = \|R^{\tilde{G}}_{|\theta|}(\mathbf{1}) - R^{\tilde{G}}_{|\theta'|}(\mathbf{1})\|_1. \tag{13}$$

Proof. First of all, because $\theta_i \theta'_i \geqslant 0$ for every coordinate i, we have for every path p: $\Phi_p(\theta)\Phi_p(\theta') \geqslant 0$ so that $|\Phi_p(\theta) - \Phi_p(\theta')| = |\Phi_p(|\theta|) - \Phi_p(|\theta'|)|$ and:

$$\|\Phi(\theta) - \Phi(\theta')\|_1 = \|\Phi(|\theta|) - \Phi(|\theta'|)\|_1 = \sum_{p \in \mathcal{P}} |\Phi_p(|\theta|) - \Phi_p(|\theta'|)|.$$

Because $|\theta_i| \geqslant |\theta'_i|$ for every coordinate i, we have $\Phi_p(|\theta|) \geqslant \Phi_p(|\theta'|)$ for every path p. Therefore, we have:

$$\begin{aligned}
\|\Phi(\theta) - \Phi(\theta')\|_1 &= \sum_{p \in \mathcal{P}} \Phi_p(|\theta|) - \Phi_p(|\theta'|) \\
&= \left(\sum_{p \in \mathcal{P}} \Phi_p(|\theta|) \right) - \left(\sum_{p \in \mathcal{P}} \Phi_p(|\theta'|) \right) \\
&= \left(\sum_{p \in \mathcal{P}} |\Phi_p(\theta)| \right) - \left(\sum_{p \in \mathcal{P}} |\Phi_p(\theta')| \right) \\
&= \|\Phi(\theta)\|_1 - \|\Phi(\theta')\|_1.
\end{aligned} \tag{14}$$

According to Theorem A.1 in Gonon et al. [2024], it holds for every parameters θ:

$$\|\Phi(\theta)\|_1 = \|R^{\tilde{G}}_{|\theta|}(\mathbf{1})\|_1.$$

We just obtained

$$\|\Phi(\theta) - \Phi(\theta')\|_1 = \|R^{\tilde{G}}_{|\theta|}(\mathbf{1})\|_1 - \|R^{\tilde{G}}_{|\theta'|}(\mathbf{1})\|_1.$$

The latter is also equal to $\|R^{\tilde{G}}_{|\theta|}(\mathbf{1}) - R^{\tilde{G}}_{|\theta'|}(\mathbf{1})\|_1$ because $|\theta_i| \geqslant |\theta'_i|$ for every coordinate i implies that for every neuron v:

$$v^{\tilde{G}}(|\theta|, \mathbf{1}) \geqslant v^{\tilde{G}}(|\theta'|, \mathbf{1}) \geqslant 0$$

so that

$$\begin{aligned}
\|R^{\tilde{G}}_{|\theta|}(\mathbf{1})\|_1 - \|R^{\tilde{G}}_{|\theta'|}(\mathbf{1})\|_1 &= \sum_{v \in N_{\text{out}}} |v^{\tilde{G}}(|\theta|, \mathbf{1})| - |v^{\tilde{G}}(|\theta'|, \mathbf{1})| \\
&= \sum_{v \in N_{\text{out}}} v^{\tilde{G}}(|\theta|, \mathbf{1}) - v^{\tilde{G}}(|\theta'|, \mathbf{1}) \\
&= \sum_{v \in N_{\text{out}}} \left| v^{\tilde{G}}(|\theta|, \mathbf{1}) - v^{\tilde{G}}(|\theta'|, \mathbf{1}) \right| \\
&= \|R^{\tilde{G}}_{|\theta|}(\mathbf{1}) - R^{\tilde{G}}_{|\theta'|}(\mathbf{1})\|_1.
\end{aligned}$$

This proves the result. □

C Proof of Theorem 3.1

We actually prove the next theorem that is stronger than Theorem 3.1. We do not state it in the main body as it requires having in mind the definition of the path-lifting Φ, recalled in Definition A.5, to understand the following notations. For parameters θ, we will denote $\Phi^I(\theta)$ (resp. $\Phi^H(\theta)$) the sub-vector of $\Phi(\theta)$ corresponding to the coordinates associated with paths starting from an input (resp. hidden) neuron. Thus, $\Phi(\theta)$ is the concatenation of $\Phi^I(\theta)$ and $\Phi^H(\theta)$.

Theorem C.1. *Consider a ReLU neural network as in Definition A.2, with output dimension equal to one. Consider associated parameters θ, θ'. If for every coordinate i, θ_i and θ'_i have the same signs or at least one of them is zero ($\theta_i\theta'_i \geqslant 0$), we have for every input x:*

$$|R_\theta(x) - R_{\theta'}(x)| \leqslant \|x\|_\infty \|\Phi^I(\theta) - \Phi^I(\theta')\|_1 + \|\Phi^H(\theta) - \Phi^H(\theta')\|_1. \tag{15}$$

Moreover, for every neural network architecture, there are parameters $\theta \neq \theta'$ and an input x such that Equation (4) is an equality.

Theorem C.1 is intentionally stated with scalar output in order to let the reader deduce the result with multi-dimensional output with his favorite norm. As an example, we derive the next corollary, which corresponds to the Theorem 3.1 given in the text body (except for the equality case, which is also an easy consequence of the equality case of Equation (15)).

Corollary C.1. *Consider an exponent $q \in [1, \infty)$ and a ReLU neural network as in Definition A.2. Consider associated parameters θ, θ'. If for every coordinate i, it holds $\theta_i\theta'_i \geqslant 0$, then for every input $x \in \mathbb{R}^{d_{in}}$:*

$$\|R_\theta(x) - R_{\theta'}(x)\|_q \leqslant \max(\|x\|_\infty, 1)\|\Phi(\theta) - \Phi(\theta')\|_1.$$

Proof of Corollary C.1. By definition of the model, it holds:

$$\|R_\theta(x) - R_{\theta'}(x)\|_q^q = \sum_{v \in N_{\text{out}}} |v(\theta, x) - v(\theta', x)|^q.$$

Recall that $\Phi^{\to v}$ is the path-lifting associated with the sub-graph $G^{\to v}$ (Definition A.5). By Theorem C.1, it holds:

$$|v(\theta, x) - v(\theta', x)|^q \leqslant \max(\|x\|_\infty^q, 1)\|\Phi^{\to v}(\theta) - \Phi^{\to v}(\theta')\|_1^q.$$

Since $\Phi(\theta) = (\Phi^{\to v}(\theta))_{v \in N_{\text{out}}}$, this implies:

$$\|R_\theta(x) - R_{\theta'}(x)\|_q^q \leqslant \max(\|x\|_\infty^q, 1)\|\Phi(\theta) - \Phi(\theta')\|_1^q. \qquad \square$$

Figure 5: Counter-example showing that the conclusion of Theorem 3.1 does not hold when the parameters have opposite signs. If the hidden neurons are ReLU neurons, the left network implements $R_\theta(x) = \mathrm{ReLU}(x)$ (with $\theta = (1 \quad 1)^T$) and the right network implements $R_{\theta'}(x) = -\mathrm{ReLU}(-x)$ (with $\theta' = (-1 \quad -1)^T$). Equation (4) does not hold since there is a single path and the product of the weights along this path is equal to one in both cases, so that $\Phi(\theta) = \Phi(\theta') = 1$ (cf Section 2) while these two functions are nonzero and have disjoint supports.

Sketch of the proof of Theorem C.1. To prove the inequality, we define the notion of admissible trajectory, show that it is enough to find an admissible trajectory in order to conclude (Lemma C.1), and then we construct such an admissible trajectory (Corollary C.2). A geometric illustration of the spirit of the proof is given in Figure 1, as detailed in the figure legend. The formal proof of Theorem C.1, including the equality case, is given at the end of the section.

Admissible trajectory: definition. Given any input vector x and two parameters θ, θ', we define an *x-admissible trajectory*[5] betweeen θ and θ' as any continuous map $t \in [0, 1] \mapsto \theta(t)$ such that for every $t \in [0, 1]$, the vector $\theta(t)$ corresponds to parameters associated with the considered network

[5]While the standard terminology for such a map $t \mapsto \theta(t)$ is rather "path" than "trajectory", we chose "trajectory" to avoid possible confusions with the notion of "path" of a DAG associated with a neural network.

architecture, with the boundary conditions $\theta(0) = \theta$ and $\theta(1) = \theta'$, and with the additional "x-admissibility property" corresponding to the existence of *finitely many* breakpoints $0 = t_0 < t_1 < \cdots < t_m = 1$ such that the path-activations matrix (see Definition A.5) $t \in [0,1] \mapsto \boldsymbol{A}(\theta(t), x)$ is constant on each interval (t_k, t_{k+1}) and such that for every path p of the graph, using the shorthand $\theta_k := \theta(t_k)$, the "reverse triangle inequality" holds (which is then, of course, an equality):

$$\sum_{k=1}^{m} |\Phi_p(\theta_k) - \Phi_p(\theta_{k-1})| \leqslant |\Phi_p(\theta_m) - \Phi_p(\theta_0)|. \tag{16}$$

Finding an admissible trajectory is enough.

Lemma C.1. *Consider an input vector x and two parameters θ, θ'. If $t \in [0,1] \mapsto \theta(t)$ is an x-admissible trajectory between θ and θ' then*

$$|R_\theta(x) - R_{\theta'}(x)| \leqslant \|x\|_\infty \|\Phi^I(\theta) - \Phi^I(\theta')\|_1 + \|\Phi^H(\theta) - \Phi^H(\theta')\|_1. \tag{17}$$

Proof. In this proof, we denote by convention $x_u := 1$ for any u that is not an input neuron. Recall that p_0 denotes the first neuron of a path p, and x_{p_0} is the coordinate of x for neuron p_0. Since for every parameters θ and every input x, it holds [Gonon et al., 2024, Lemma A.1]

$$R_\theta(x) = \sum_{p \in \mathcal{P}} x_{p_0} a_p(\theta, x) \Phi_p(\theta),$$

we deduce that for every $k \in \{1, \dots, m\}$ and every $t_{k-1} < t' < t < t_k$, we have:

$$R_{\theta(t)}(x) - R_{\theta(t')}(x) = \sum_{p \in \mathcal{P}} x_{p_0} \left(a_p(\theta(t), x) \Phi_p(\theta(t)) - a_p(\theta(t'), x) \Phi_p(\theta(t')) \right).$$

Since both t and t' belong to the same interval (t_{k-1}, t_k) and since $t \mapsto \theta(t)$ is an admissible trajectory, the path-activations $a_p(\theta(t), x) = a_p(\theta(t'), x)$ are the same for every path p. Thus, it holds:

$$R_{\theta(t)}(x) - R_{\theta(t')}(x) = \sum_{p \in \mathcal{P}} x_{p_0} a_p(\theta(t), x) \left(\Phi_p(\theta(t)) - \Phi_p(\theta(t')) \right).$$

Recall that the set of paths $\mathcal{P}$ is partitioned into the sets $\mathcal{P}_I$ and $\mathcal{P}_H$ of paths starting respectively from an input and a hidden neuron. By the convention taken in this proof, for $p \in \mathcal{P}_H$, it holds $x_{p_0} = 1$. Thus:

$$\begin{aligned} R_{\theta(t)}(x) - R_{\theta(t')}(x) = &\sum_{p \in \mathcal{P}_I} x_{p_0} a_p(\theta(t), x) \left(\Phi_p(\theta(t)) - \Phi_p(\theta(t')) \right) \\ &+ \sum_{p \in \mathcal{P}_H} a_p(\theta(t), x) \left(\Phi_p(\theta(t)) - \Phi_p(\theta(t')) \right). \end{aligned}$$

Recall that a path-activation is always equal to 0 or 1 by definition, so that:

$$\begin{aligned} \left| R_{\theta(t)}(x) - R_{\theta(t')}(x) \right| &\leqslant \sum_{p \in \mathcal{P}_I} |x_{p_0}| \, |\Phi_p(\theta(t)) - \Phi_p(\theta(t'))| + \sum_{p \in \mathcal{P}_H} |\Phi_p(\theta(t)) - \Phi_p(\theta(t'))| \\ &\leqslant \|x\|_\infty \|\Phi^I(\theta(t)) - \Phi^I(\theta(t'))\|_1 + \|\Phi^H(\theta(t)) - \Phi^H(\theta(t'))\|_1. \end{aligned}$$

Considering the limits $t \to t_k$ and $t' \to t_{k-1}$ gives by continuity of both $\theta \mapsto R_\theta(x)$ and $\theta \mapsto \Phi(\theta)$:

$$\left| R_{\theta_k}(x) - R_{\theta_{k-1}}(x) \right| \leqslant \|x\|_\infty \|\Phi^I(\theta_k) - \Phi^I(\theta_{k-1})\|_1 + \|\Phi^H(\theta_k) - \Phi^H(\theta_{k-1})\|_1.$$

Since $\theta = \theta_0$ and $\theta' = \theta_m$, using the triangle inequality yields:

$$|R_\theta(x) - R_{\theta'}(x)| \leqslant \|x\|_\infty \sum_{k=1}^{m} \|\Phi^I(\theta_k) - \Phi^I(\theta_{k-1})\|_1 + \sum_{k=1}^{m} \|\Phi^H(\theta_k) - \Phi^H(\theta_{k-1})\|_1. \tag{18}$$

See Figure 1 for an illustration of what is happening here. By definition, since the trajectory is x-admissible, we have by Equation (16)

$$\sum_{k=1}^{m} \|\Phi^I(\theta_k) - \Phi^I(\theta_{k-1})\|_1 \leqslant \|\Phi^I(\theta_m) - \Phi^I(\theta_0)\|_1 = \|\Phi^I(\theta') - \Phi^I(\theta)\|_1$$

and

$$\sum_{k=1}^{m} \|\Phi^H(\theta_k) - \Phi^H(\theta_{k-1})\|_1 \leqslant \|\Phi^H(\theta_m) - \Phi^H(\theta_0)\|_1 = \|\Phi^H(\theta') - \Phi^H(\theta)\|_1.$$

With Equation (18), this proves Equation (17). □

Construction of an admissible trajectory. In the formal proof of Theorem C.1 we will see that it is enough to establish the result when all the coordinates of θ, θ' are nonzero.

Definition C.1. *Consider two parameters θ, θ' with only nonzero coordinates. For every $t \in [0,1]$ and every i, define the following trajectory*[6] *$t \mapsto \theta(t)$ between θ and θ':*

$$(\theta(t))_i = \operatorname{sgn}(\theta_i)|\theta_i|^{1-t}|\theta'_i|^t, \tag{19}$$

where $\operatorname{sgn}(y) := \mathbb{1}_{y>0} - \mathbb{1}_{y<0} \in \{-1, 0, +1\}$ *for any* $y \in \mathbb{R}$.

Observe that the trajectory in Equation (19) is well-defined since the coordinates of θ and θ' are nonzero by assumption. As proved in the next lemma, this trajectory has indeed finitely many breakpoints where the path-activations change. This is basically because for every coordinate i, the trajectory $t \in [0,1] \to (\theta(t))_i$ is analytic[7]. As a consequence, the set of t's where a coordinate of the path-activations matrix $\boldsymbol{A}(\theta(t), x)$ does change can be realized as a set of zeroes of an analytic function on $\mathbb{C}$, and since these zeroes must be isolated, there could only be finitely of them in the compact $[0,1]$, except if this coordinate is constant equal to zero.

Lemma C.2. *Consider $n \in \mathbb{N}_{>0}$ inputs $X = (x_1, \ldots, x_n) \in (\mathbb{R}^{d_{in}})^n$. For parameters θ, θ' with only nonzero coordinates, consider the trajectory $t \in [0,1] \mapsto \theta(t)$ defined in Equation (19). There exists finitely many breakpoints $0 = t_0 < t_1 < \cdots < t_m = 1$ such that for every $i = 1, \ldots, n$, the path-activations matrix $t \in [0,1] \mapsto \boldsymbol{A}(\theta(t), x_i)$ is constant on each interval (t_k, t_{k+1}).*

Proof of Lemma C.2. After showing that the result for arbitrary n follows from the result for $n = 1$, we establish the latter by an induction on a topological sorting of the graph G.

Reduction to $n = 1$. If for every $i = 1, \ldots, n$, we have a finite family of breakpoints $(t_k^i)_k$, then the union of these families gives a finite family of breakpoints that works for every i. It is then sufficient to prove that for a *single* arbitrary input x, there are finitely many breakpoints $0 = t_0 < t_1 < \cdots < t_m = 1$ such that the path-activations matrix $t \in [0,1] \mapsto \boldsymbol{A}(\theta(t), x)$ remains constant on each interval (t_k, t_{k+1}).

For the rest of the proof, consider a single input x, and define for any neuron v the property

$$\begin{gathered}\text{there are finitely many breakpoints } 0 = t_0 < t_1 < \cdots < t_m = 1 \text{ such that for every } k: \\ \text{the map } t \in [t_k, t_{k+1}] \mapsto v(\theta(t), x) \text{ is analytic,}\end{gathered} \tag{20}$$

and the functions $t \mapsto a_v(\theta(t), x), t \mapsto a_{u \to v}(\theta(t), x)$, for each $u \in \operatorname{ant}(v)$, are constant on (t_k, t_{k+1})

Reduction to proving Property (20) for every neuron v. We will soon prove that Property (20) holds for every neuron v. Let us see why this is enough to reach the desired conclusion. By the same argument as in the reduction to $n = 1$, the union of the breakpoints associated to all neurons yields finitely many intervals such that, on each interval, *all functions* $t \mapsto a_v(\theta(t), x)$, $v \in N$, and $a_{u \to v}(\theta(t), x)$, $u \in \operatorname{ant}(v)$, are constant. By Definition A.5 this implies that $t \mapsto \boldsymbol{A}(\theta(t), x)$ is constant on each corresponding open interval.

Proof of Property (20) for every neuron v by induction on a topological sorting [Cormen et al., 2009, Section 22.4] of the graph. We start with input neurons v since by Definition A.2, these are the ones without antecedents so they are the first to appear in a topological sorting.

Initialization: Property (20) for input neurons. For any input neuron v, it holds by Definition A.2 $v(\theta, x) = x_v$ that is constant in θ. Thus $t \in [0,1] \mapsto v(\theta(t), x)$ is trivially analytic. Since v is an input neuron, it has no antecedent, and by Definition A.5 we have $a_v(\theta, x) := 1$. This shows that Property (20) holds for input neurons.

[6] This trajectory is linear in log-parameterization: for every $t \mapsto \ln(|(\theta(t))_i|)$ is linear in t.

[7] A function $f : C \mapsto \mathbb{R}$ is analytic on a *closed* subset $C \subset \mathbb{R}$ if there exists an open set $C \subset O \subset \mathbb{R}$ such that f is the restriction to C of a function that is analytic on O.

Induction: Now, consider a non-input neuron v and assume Property (20) to hold for every neuron coming before v in the considered topological sorting. Since every antecedent of v must come before v in the topological sorting, there are finitely many breakpoints $0 = t_0 < t_1 < \cdots < t_m = 1$ such that for every $u \in \mathrm{ant}(v)$ and every k, the map $t \in [t_k, t_{k+1}] \mapsto u(\theta(t), x)$ is analytic. We distinguish three cases depending on the activation function of neuron v.

• **Case of an identity neuron.** By Definition A.2 $v(\theta(t), x) = b_v + \sum_{u \in \mathrm{ant}(v)} u(\theta(t), x)\theta(t)^{u \to v}$ and for every k it is clear that it is anaytic as it is the case for each $t \in [t_k, t_{k+1}] \mapsto u(\theta(t), x)$ by induction, and it is also the case for $t \in [t_k, t_{k+1}] \mapsto \theta(t)^{u \to v}$ by definition (Equation (19)). Since v is an identity neuron by Definition A.5 we have $a_{u \to v}(\theta(t), x) = a_v(\theta(t), x) = 1$ for every t. This establishes Property (20) for v.
• **Case of a ReLU neuron.** By Definition A.2: $v(\theta, x) = \mathrm{ReLU}(\mathrm{pre}_v(\theta, x))$ where we denote the so-called pre-activation of v by $\mathrm{pre}_v(\theta, x) := b_v + \sum_{u \in \mathrm{ant}(v)} u(\theta, x)\theta^{u \to v}$. Reasoning as in the case of identity neurons, the induction hypothesis implies that for every k the function $t \in [t_k, t_{k+1}] \mapsto \mathrm{pre}_v(\theta(t), x)$ is analytic. We distinguish two sub-cases:
– If this function is identically zero then $t \in [t_k, t_{k+1}] \mapsto v(\theta(t), x)$ is null, so it is analytic, and by Definition A.5 $a_{u \to v}(\theta(t), x) = a_v(\theta(t), x) = \mathbb{1}_{v(\theta, x) > 0} = 0$ for every $u \in \mathrm{ant}(v)$;
– Otherwise this analytic function can only vanish a finite number of times on the compact $[t_k, t_{k+1}]$: there are times $t_k = s_0 < s_1 < \cdots < s_n = t_{k+1}$ such that for each j, $s \in (s_j, s_{j+1}) \mapsto \mathrm{pre}_v(\theta(s), x)$ has constant (nonzero) sign and can be extended into an analytic function on $\mathbb{C}$. For each segment (s_j, s_{j+1}) where the sign is negative, we deduce that for every $s \in [s_j, s_{j+1}]$ we have $v(\theta(s), x) = 0$, hence by Definition A.5, $a_v(\theta(s), x) = a_{u \to v}(\theta(s), x) = 0$ for every $u \in \mathrm{ant}(v)$; on the other segments, we have $v(\theta(s), x) = \mathrm{pre}_v(\theta(s), x)$ for every $s \in [s_j, s_{j+1}]$, and therefore $a_v(\theta(s), x) = a_{u \to v}(\theta(s), x) = 1$ for every $s \in (s_j, s_{j+1})$ and $u \in \mathrm{ant}(v)$.
Overall, on all the resulting (finitely many) segments, we obtain all the properties establishing that Property (20) indeed holds for v.
• **Case of a K-max-pooling neuron.** Recall that by Definition A.2, the output of v is the K-th largest component of $\mathrm{pre}_v(\theta, x) := (u(\theta, x)\theta^{u \to v})_{u \in \mathrm{ant}(v)}$, with ties between antecedents decided by lexicographic order. Since each $t \in [t_k, t_{k+1}] \mapsto u(\theta(t), x)$ is analytic, and so does $t \mapsto \theta(t)^{u \to v}$, this is also the case of each coordinate of $\mathrm{pre}_v(\theta(t), x)$.
Consider any k. We are going to prove that there are finitely many breakpoints $t_k = s_0 < s_1 < \cdots < s_\ell = t_{k+1}$ such that on each interval (s_j, s_{j+1}), there is an antecedent $u \in \mathrm{ant}(v)$ such that

$$v(\theta(s), x) = u(\theta(s), x)\theta(s)^{u \to v}, \text{ for every } s \in (s_j, s_{j+1}).$$

By the same reasoning as above this will imply that Property (20) holds for v.
For any neurons $u \neq u' \in \mathrm{ant}(v)$, denote $\delta_{u,u'}(\theta) := u(\theta(t), x)\theta(t)^{u \to v} - u'(\theta(t), x)\theta(t)^{u' \to v}$ and let U be the set of $u \in \mathrm{ant}(v)$ such that: for each $u' \in \mathrm{ant}(v)$, either $t \mapsto \delta_{u,u'}(\theta(t))$ is not identically zero on $[t_k, t_{k+1}]$, or u is before u' in lexicographic order. With this definition, for each pair $u \neq u' \in U$, the function $t \in [t_k, t_{k+1}] \mapsto \delta_{u,u'}(\theta(t), x)$ is not identically zero and is analytic, so that there are only finitely many breakpoints $t_k = s_0^{u,u'} < s_1^{u,u'} < \cdots < s_{\ell(u,u')}^{u,u'} = t_{k+1}$ where it vanishes on the compact $[t_k, t_{k+1}]$. Considering the union over all pairs $u, u' \in U$ of these finite families of breakpoints, we get a finite family of breakpoint $t_k = s_0 < s_1 < \cdots < s_\ell = t_{k+1}$ such that on each interval (s_j, s_{j+1}), the ordering between the coordinates of $\mathrm{pre}_v(\theta(s), x)$ in U is strict and stays the same. To conclude, it is not hard to check that, by the definition of U and of $*$-max-pooling, the output of v only depends on the coordinates of $\mathrm{pre}_v(\theta(s), x)$ indexed by U. This yields the claim and concludes the proof. □

For $y \in \mathbb{R}$, recall that we consider $\mathrm{sgn}(y) = \mathbb{1}_{y>0} - \mathbb{1}_{y<0} \in \{-1, 0, +1\}$ and extend it to vectors by applying it coordinate-wise.

Corollary C.2. *Consider two parameters θ, θ' with nonzero coordinates and such that $\mathrm{sgn}(\theta) = \mathrm{sgn}(\theta')$. Then the trajectory defined in Equation (19) is x-admissible for every input vector x.*

Proof. First, the trajectory is well-defined since the coordinates are nonzero, and it satisfies the boundary conditions $\theta(0) = \theta$ and $\theta(1) = \theta'$ since the coordinates have the same signs.

Second, Lemma C.2 proves that for every x, there are finitely many breakpoints $0 = t_0 < t_1 < \cdots < t_m = 1$ such that the path-activations matrix $t \in [0, 1] \mapsto \boldsymbol{A}(\theta(t), x)$ is constant on each interval (t_{k-1}, t_k).

It now only remains to prove that Equation (16) holds to prove that this is an x-admissible trajectory. Consider a path p. For a coordinate i of the parameters, we write $i \in p$ either if $i = p_0$ and p_0 is a hidden neuron, or if $i = e$ is an edge along the path p. Define $\operatorname{sgn}(p) := \prod_{i \in p} \operatorname{sgn}(\theta_i)$ and note that $\operatorname{sgn}(p) \neq 0$ since θ has only nonzero coordinates by assumption. Denote $|\theta|$ the vector deduced from θ by applying the absolute value coordinate-wise. It is easy to check by definition of the path-lifting Φ that for every $t \in [0, 1]$:

$$\Phi_p(\theta(t)) = \operatorname{sgn}(p)\Phi_p(|\theta|)^{1-t}\Phi_p(|\theta'|)^t = \operatorname{sgn}(p)\Phi_p(|\theta(t_0)|)^{1-t}\Phi_p(|\theta(t_m)|)^t.$$

Denote by $a := \Phi_p(|\theta'|) = \Phi_p(|\theta(t_m)|)$ and by $b = \Phi_p(|\theta|) = \Phi_p(|\theta(t_0)|)$. The latter rewrites:

$$\Phi_p(\theta(t)) = \operatorname{sgn}(p)a^t b^{1-t}.$$

Thus, Equation (16) holds if, and only if,

$$\sum_{k=1}^{m} |\operatorname{sgn}(p)| \left|a^{t_k}b^{1-t_k} - a^{t_{k-1}}b^{1-t_{k-1}}\right| \leqslant |\operatorname{sgn}(p)| \, |a - b| \, .$$

Simplifying by $\operatorname{sgn}(p) \neq 0$, Equation (16) is equivalent to:

$$\sum_{k=1}^{m} \left|a^{t_k}b^{1-t_k} - a^{t_{k-1}}b^{1-t_{k-1}}\right| \leqslant |a - b| \, .$$

Let us now observe that $t \mapsto a^t b^{1-t}$ is monotonic and conclude. We only do so when $a \geqslant b$, the other case being similar. Since by definition, we also have a and b positive, it holds for $t > t'$

$$a^{t-t'} \geqslant b^{t-t'} \text{ that is equivalent to } a^t b^{1-t'} \geqslant a^t b^{1-t'}.$$

We then have a telescopic sum:

$$\sum_{k=1}^{m} \left|a^{t_k}b^{1-t_k} - a^{t_{k-1}}b^{1-t_{k-1}}\right| = \sum_{k=1}^{m} a^{t_k}b^{1-t_k} - a^{t_{k-1}}b^{1-t_{k-1}}$$

$$= a^{t_m}b^{1-t_m} - a^{t_0}b^{1-t_0} = a - b = |a - b| \, .$$

This shows Equation (16), proving that $t \mapsto \theta(t)$ is an admissible trajectory, and thus the result. □

Proof of Theorem C.1. **Equality case.** Consider an arbitrary neural network architecture, an input neuron v_0 and a path $p = v_0 \to v_1 \to \dots v_d$. Consider θ (resp. θ') with only zero coordinates, except for $\theta^{v_\ell \to v_{\ell+1}} = a > 0$ (resp. $(\boldsymbol{\theta'})^{v_\ell \to v_{\ell+1}} = b > 0$) for every $\ell \in [\![0, d-1]\!]$. Consider the input x to have only zero coordinates except for $x_{v_0} > 0$. It is easy to check that $R_\theta(x) = a^d x_{v_0}$ and $R_{\theta'}(x) = b^d x_{v_0}$. Since $\|x\|_\infty = x_{v_0}$, $\|\Phi^I(\theta) - \Phi^I(\theta')\|_1 = |a^d - b^d|$ and $\|\Phi^H(\theta) - \Phi^H(\theta')\|_1 = 0$, this shows that Equation (15) is an equality for these parameters.

Proof of the inequality. By continuity of both handsides of (15) with respect to θ, θ', it is enough to prove the result when all coordinates of θ, θ' are nonzero, i.e., under the stronger assumption that $\theta_i \theta'_i > 0$ for every coordinate index i. Under this assumption, by Corollary C.2, the trajectory $t \mapsto \theta(t)$ defined in Equation (19) is x-admissible for every input vector x. The conclusion follows by Lemma C.1. □

D Details on the experiments of Section 4.2

Model and data. We train a dense ResNet18 [He et al., 2016] on ImageNet-1k, using 99% of the 1,281,167 images of the training set for training, the other 1% for validation. The PyTorch code for normalization at inference is standard:

```
inference_normalization = transforms.Compose([
        transforms.Resize(256),
        transforms.CenterCrop(224),
        transforms.ToTensor(),
        transforms.Normalize(
            mean=[0.485, 0.456, 0.406],
            std=[0.229, 0.224, 0.225]
            ),
    ])
```

Optimization. We use SGD for 90 epochs, learning rate 0.1, weight-decay 0.0001, batch size 1024, and a multi-step scheduler where the learning rate is divided by 10 at epochs 30, 60 and 80. The epoch out of the 90 ones with maximum validation top-1 accuracy is considered as the final epoch. Doing 90 epochs took us about 18 hours on a single A100-40GB GPU.

Pruning. At the end of the training phase, we prune (i.e. set to zero) $p\%$ of the remaining weights of each convolutional layer, and $\frac{p}{2}\%$ of the final fully connected layer for a layerwise method. For a global pruning method, we prune the same amount of weights but globally. We save the mask and rewind the weights to their values after the first 5 epochs of the dense network, and train for 85 remaining epochs. This exactly corresponds to the hyperparameters and pruning algorithm of the lottery ticket literature [Frankle et al., 2021].

Random rescaling. Consider a pair of consecutive convolutional layers in the same basic block of the ResNet18 architecture, for instance the ones of the first basic block: `model.layer1[0].conv1` and `model.layer1[0].conv2` in PyTorch, with `model` being the ResNet18. Denote by C the number of output channels of the first convolutional layer, which is also the number of input channels of the second one. For each channel $c \in [\![1, C]\!]$, we choose uniformly at random a rescaling factor $\lambda \in \{1, 128, 4096\}$ and multiply the output channel c of the first convolutional layer by λ, and divide the input channel c of the second convolutional layer by λ. In order to preserve the input-output relationship, we also multiply by λ the running mean and the bias of the batch normalization layer that is in between (`model.layer1[0].bn1` in the previous example). Here is an illustrative Python code (that should be applied to the correct layer weights as described above):

```
    factors = np.array([1, 128, 4096])

    out_channels1, _, _, _ = weights_conv1.shape

    for out in range(out_channels1):
                factor = np.random.choice(factors)
                weights_conv1[out, :, :, :] *= factor
                weights_conv2[:, out, :, :] /= factor
                running_mean[out] *= factor
                bias[out] *= factor
```

E Lipschitz property of Φ

We first establish Lipschitz properties of $\theta \mapsto \Phi(\theta)$. Combined with the main result of this paper, Theorem 3.1, or with Corollary C.1, they establish a Lipschitz property of $\theta \mapsto R_\theta(x)$ for each x, and of the functional map $\theta \mapsto R_\theta(\cdot)$ in the uniform norm on any bounded domain. This is complementary to the Lipschitz property of $x \mapsto R_\theta(x)$ studied elsewhere in the literature, see e.g. [Gonon et al., 2024]. These results are also used to bound the covering numbers of $\Phi(\Theta)$ in the proof of Theorem 5.1.

Lemma E.1. *Consider $q \in [1, \infty)$, parameters θ and θ', and a neuron v. Then, it holds:*

$$\begin{aligned}&\|\Phi^{\to v}(\theta) - \Phi^{\to v}(\theta')\|_q^q \\ &\leqslant \max_{p \in \mathcal{P}^{\to v}} \sum_{\ell=1}^{\texttt{length}(p)} \left(\prod_{k=\ell+1}^{\texttt{length}(p)} \|\theta^{\to p_k}\|_q^q \right) \left(|b_{p_\ell} - b'_{p_\ell}|^q + \|\theta^{\to p_\ell} - (\boldsymbol{\theta'})^{\to p_\ell}\|_q^q \max_{u \in \mathrm{ant}(p_\ell)} \|\Phi^{\to u}(\theta')\|_q^q \right)\end{aligned} \tag{21}$$

with the convention that an empty sum and product are respectively equal to zero and one. Recall also that by convention, biases of $*$-max-pooling neurons v are set to $b_v = 0$ (Definition A.5).

Note that when all the paths in $\mathcal{P}^{\to v}$ have the same length L, Equation (21) is homogeneous: multiplying both θ and θ' coordinate-wise by a scalar λ scales both sides of the equations by λ^L.

Proof. The proof of Equation (21) goes by induction on a topological sorting of the graph. The first neurons of the sorting are the neurons without antecedents, *i.e.*, the input neurons by definition. Consider an input neuron v. There is only a single path ending at v: the path $p = v$. By Definition A.5, $\Phi^{\to v}(\cdot) = \Phi_v(\cdot) = 1$ so the left hand-side is zero. On the right-hand side, there is only a single

choice for a path ending at v: this is the path $p = v$ that starts and ends at v. Thus $D = 0$, and the maximum is zero (empty sum). This proves Equation (21) for input neurons.

Consider a neuron $v \notin N_{\text{in}}$ and assume that this is true for every neuron before v in the considered topological sorting. Recall that, by definition, $\Phi^{\to v}$ is the path-lifting of $G^{\to v}$ (see Definition A.5). The paths in $G^{\to v}$ are $p = v$, and the paths going through antecedents of v (v has antecedents since it is not an input neuron). So we have $\Phi^{\to v}(\theta) = \begin{pmatrix} (\Phi^{\to u}(\theta)\theta^{u\to v})_{u\in \text{ant}(v)} \\ b_v \end{pmatrix}$, where we again recall that $\Phi^{\to u}(\cdot) = 1$ for input neurons u, and $b_u = 0$ for $*$-max-pooling neurons. Thus, we have:

$$\begin{aligned}
&\|\Phi^{\to v}(\theta) - \Phi^{\to v}(\theta)\|_q^q \\
&= |b_v - b'_v|^q + \sum_{u\in \text{ant}(v)} \|\Phi^{\to u}(\theta)\theta^{u\to v} - \Phi^{\to u}(\theta')(\boldsymbol{\theta'})^{u\to v}\|_q^q \\
&\leqslant |b_v - b'_v|^q + \sum_{u\in \text{ant}(v)} \left(\|\Phi^{\to u}(\theta) - \Phi^{\to u}(\theta')\|_q^q |\theta^{u\to v}|^q + \|\Phi^{\to u}(\theta')\|_q^q |\theta^{u\to v} - (\boldsymbol{\theta'})^{u\to v}|^q\right) \\
&\leqslant |b_v - b'_v|^q + \|\theta^{\to v}\|_q^q \max_{u\in \text{ant}(v)} \|\Phi^{\to u}(\theta) - \Phi^{\to u}(\theta')\|_q^q + \|\theta^{\to v} - (\boldsymbol{\theta'})^{\to v}\|_q^q \max_{u\in \text{ant}(v)} \|\Phi^{\to u}(\theta')\|_q^q.
\end{aligned}$$

Using the induction hypothesis (Equation (21)) on the antecedents of v and observing that $p \in \mathcal{P}^{\to v}$ if, and only if there are $u \in \text{ant}(v), r \in \mathcal{P}^{\to u}$ such that $p = r \to v$ gives (we highlight in **blue** the important changes):

$$\begin{aligned}
&\|\Phi^{\to v}(\theta) - \Phi^{\to v}(\theta)\|_q^q \leqslant |b_v - b'_v|^q + \|\theta^{\to v} - (\boldsymbol{\theta'})^{\to v}\|_q^q \max_{u\in \text{ant}(v)} \|\Phi^{\to u}(\theta')\|_q^q \\
&+ \|\boldsymbol{\theta}^{\to v}\|_{\boldsymbol{q}}^{\boldsymbol{q}} \max_{\boldsymbol{u\in \text{ant}(v)}} \max_{\boldsymbol{r\in \mathcal{P}^{\to u}}} \sum_{\ell=1}^{\texttt{length}(r)} \left(\prod_{k=\ell+1}^{\texttt{length}(r)} \|\theta^{\to r_k}\|_q^q\right) \left(|b_{r_\ell} - b'_{r_\ell}|^q + \|\theta^{\to r_\ell} - (\boldsymbol{\theta'})^{\to r_\ell}\|_q^q \max_{w\in \text{ant}(r_\ell)} \|\Phi^{\to w}(\theta')\|_q^q\right). \\
&= |b_v - b'_v|^q + \|\theta^{\to v} - (\boldsymbol{\theta'})^{\to v}\|_q^q \max_{u\in \text{ant}(v)} \|\Phi^{\to u}(\theta')\|_q^q \\
&+ \max_{\boldsymbol{p\in \mathcal{P}^{\to v}}} \sum_{\ell=1}^{\texttt{length}(\boldsymbol{p})-1} \left(\prod_{k=\ell+1}^{\texttt{length}(p)} \|\theta^{\to p_k}\|_q^q\right) \left(|b_{p_\ell} - b'_{p_\ell}|^q + \|\theta^{\to p_\ell} - (\boldsymbol{\theta'})^{\to p_\ell}\|_q^q \max_{w\in \text{ant}(p_\ell)} \|\Phi^{\to w}(\theta')\|_q^q\right) \\
&= \max_{p\in \mathcal{P}^{\to v}} \sum_{\ell=1}^{\texttt{length}(\boldsymbol{p})} \left(\prod_{k=\ell+1}^{\texttt{length}(p)} \|\theta^{\to p_k}\|_q^q\right) \left(|b_{p_\ell} - b'_{p_\ell}|^q + \|\theta^{\to p_\ell} - (\boldsymbol{\theta'})^{\to p_\ell}\|_q^q \max_{w\in \text{ant}(p_\ell)} \|\Phi^{\to w}(\theta')\|_q^q\right).
\end{aligned}$$

This proves Equation (21) for v and concludes the induction. □

In the sequel it will be useful to restrict the analysis to *normalized* parameters, defined as parameters $\tilde{\theta}$ such that $\left\| \begin{pmatrix} \tilde{\theta}^{\to v} \\ \tilde{b}_v \end{pmatrix} \right\|_1 \in \{0, 1\}$ for every $v \in N \setminus (N_{\text{out}} \cup N_{\text{in}})$. Thanks to the rescaling-invariance of ReLU neural network parameterizations, Algorithm 1 in Gonon et al. [2024] allows to rescale *any* parameters θ into a normalized version $\tilde{\theta}$ such that $R_{\tilde{\theta}} = R_\theta$ and $\Phi(\theta) = \Phi(\tilde{\theta})$ [Gonon et al., 2024, Lemma B.2]. This implies the next simpler results for normalized parameters.

Theorem E.1. *Consider $q \in [1, \infty)$. For every normalized parameters θ, θ' obtained as the output of Algorithm 1 in Gonon et al. [2024], it holds:*

$$\begin{aligned}
&\|\Phi(\theta) - \Phi(\theta')\|_q^q \leqslant \sum_{v\in N_{out}\setminus N_{in}} |b_v - b'_v|^q + \|\theta^{\to v} - (\boldsymbol{\theta'})^{\to v}\|_q^q \\
&+ \min\left(\|\Phi(\theta)\|_q^q, \|\Phi(\theta')\|_q^q\right) \max_{p\in \mathcal{P}: p_{\text{end}}\notin N_{in}} \sum_{\ell=1}^{\texttt{length}(p)-1} \left(|b_{p_\ell} - b'_{p_\ell}|^q + \|\theta^{\to p_\ell} - (\boldsymbol{\theta'})^{\to p_\ell}\|_q^q\right). \quad (22)
\end{aligned}$$

where we recall that $b_v = 0$ for $*$-max-pooling neurons v.

Denote by $\mathtt{N}(\theta)$ the normalized version of θ, obtained as the output of Algorithm 1 in Gonon et al. [2024]. It can be checked that if $\theta = \mathtt{N}(\tilde{\theta})$ and $\theta' = \mathtt{N}(\tilde{\theta}')$, and if all the paths have the same lengths

L, then multiplying both $\tilde{\theta}$ and $\tilde{\theta}'$ coordinate-wise by a scalar λ does not change their normalized versions θ and θ', except for the biases and the incoming weights of all output neurons that are scaled λ^L. As a consequence, Equation (22) is homogeneous: both path-liftings on the left-hand-side and the right-hand-side are multiplied by λ^L, and so is the sum over $v \in N_{\text{out}} \setminus N_{\text{in}}$ in the right-hand-side, while the maximum over p is unchanged since it only involves normalized coordinates that do not change.

For networks used in practice, it holds $N_{\text{out}} \cap N_{\text{in}} = \emptyset$ so that $N_{\text{out}} \setminus N_{\text{in}}$ is just N_{out}, but the above theorem also covers the somewhat pathological case of DAG architectures G where one or more input neurons are also output neurons.

Proof of Theorem E.1. Since $\Phi(\theta) = \left(\Phi^{\to v}(\theta)\right)_{v \in N_{\text{out}}}$, it holds

$$\|\Phi(\theta) - \Phi(\theta')\|_q^q = \sum_{v \in N_{\text{out}}} \|\Phi^{\to v}(\theta) - \Phi^{\to v}(\theta')\|_q^q.$$

By Definition A.5, it holds for every input neuron v: $\Phi^{\to v}(\cdot) = 1$. Thus, the sum can be taken over $v \in N_{\text{out}} \setminus N_{\text{in}}$:

$$\|\Phi(\theta) - \Phi(\theta')\|_q^q = \sum_{v \in N_{\text{out}} \setminus N_{\text{in}}} \|\Phi^{\to v}(\theta) - \Phi^{\to v}(\theta')\|_q^q.$$

Besides, observe that many norms appearing in Equation (21) are at most one for normalized parameters. Indeed, for such parameters it holds for every $u \in N \setminus (N_{\text{in}} \cup N_{\text{out}})$: $\|\theta^{\to u}\|_q^q \leqslant 1$ [Gonon et al., 2024, Lemma B.2]. As a consequence, for $p \in \mathcal{P}$ and any $\ell \in [\![0, \texttt{length}(p) - 1]\!]$ we have:

$$\prod_{k=\ell+1}^{\texttt{length}(p)} \|\theta^{\to p_k}\|_q^q = \left(\prod_{k=\ell+1}^{\texttt{length}(p)-1} \underbrace{\|\theta^{\to p_k}\|_q^q}_{\leqslant 1} \right) \|\theta^{\to p_{\text{end}}}\|_q^q \leqslant \|\theta^{\to p_{\text{end}}}\|_q^q.$$

Moreover, for normalized parameters θ and $u \notin N_{\text{out}}$, it also holds $\|\Phi^{\to u}(\theta)\|_q^q \leqslant 1$ [Gonon et al., 2024, Lemma B.3]. Thus, Equation (21) implies for any $v \in N_{\text{out}}$, and any normalized parameters θ and θ':

$$\begin{aligned}
&\|\Phi^{\to v}(\theta) - \Phi^{\to v}(\theta')\|_q^q \\
&\leqslant |b_v - b'_v|^q + \|\theta^{\to v} - (\boldsymbol{\theta'})^{\to v}\|_q^q + \|\theta^{\to v}\|_q^q \max_{p \in \mathcal{P}^{\to v}} \sum_{\ell=1}^{\texttt{length}(p)-1} \left(|b_{p_\ell} - b'_{p_\ell}|^q + \|\theta^{\to p_\ell} - (\boldsymbol{\theta'})^{\to p_\ell}\|_q^q \right).
\end{aligned}$$

Thus, we get:

$$\begin{aligned}
&\|\Phi(\theta) - \Phi(\theta')\|_q^q \\
&= \sum_{v \in N_{\text{out}} \setminus N_{\text{in}}} \|\Phi^{\to v}(\theta) - \Phi^{\to v}(\theta')\|_q^q \\
&\leqslant \sum_{v \in N_{\text{out}} \setminus N_{\text{in}}} \left(|b_v - b'_v|^q + \|\theta^{\to v} - (\boldsymbol{\theta'})^{\to v}\|_q^q \right) \\
&+ \sum_{v \in N_{\text{out}} \setminus N_{\text{in}}} \|\theta^{\to v}\|_q^q \max_{p \in \mathcal{P}^{\to v}} \sum_{\ell=1}^{\texttt{length}(p)-1} \left(|b_{p_\ell} - b'_{p_\ell}|^q + \|\theta^{\to p_\ell} - (\boldsymbol{\theta'})^{\to p_\ell}\|_q^q \right) \\
&\leqslant \sum_{v \in N_{\text{out}} \setminus N_{\text{in}}} \left(|b_v - b'_v|^q + \|\theta^{\to v} - (\boldsymbol{\theta'})^{\to v}\|_q^q \right) \\
&+ \left(\sum_{v \in N_{\text{out}} \setminus N_{\text{in}}} \|\theta^{\to v}\|_q^q \right) \max_{p \in \mathcal{P} : p_{\text{end}} \notin N_{\text{in}}} \sum_{\ell=1}^{\texttt{length}(p)-1} \left(|b_{p_\ell} - b'_{p_\ell}|^q + \|\theta^{\to p_\ell} - (\boldsymbol{\theta'})^{\to p_\ell}\|_q^q \right).
\end{aligned}$$

It remains to use that $\sum_{v \in N_{\text{out}} \setminus N_{\text{in}}} \|\theta^{\to u_v}\|_q^q \leqslant \|\Phi(\theta)\|_q^q$ for normalized parameters θ [Gonon et al., 2024, Theorem B.1, case of equality] to conclude that:

$$\|\Phi(\theta) - \Phi(\theta')\|_q^q \leqslant \sum_{v \in N_{\text{out}} \setminus N_{\text{in}}} \left(|b_v - b'_v|^q + \|\theta^{\to v} - (\boldsymbol{\theta'})^{\to v}\|_q^q \right) + \|\boldsymbol{\Phi(\theta)}\|_{\boldsymbol{q}}^{\boldsymbol{q}} \max_{p \in \mathcal{P} : p_{\text{end}} \notin N_{\text{in}}} \sum_{\ell=1}^{\texttt{length}(p)-1} \left(|b_{p_\ell} - b'_{p_\ell}|^q + \|\theta^{\to p_\ell} - (\boldsymbol{\theta'})^{\to p_\ell}\|_q^q \right).$$

The term in **blue** can be replaced by $\boldsymbol{\min\left(\|\Phi(\theta)\|_q^q, \|\Phi(\theta')\|_q^q\right)}$ by repeating the proof with θ and θ' exchanged (everything else is invariant under this exchange). □

F Recovering a known bound with Theorem 3.1

It is already known in the literature that for every input x and every parameters θ, θ' (even with different signs) of a layered fully-connected neural network with L affine layers and $L+1$ layers of neurons, $N_0 = N_{\text{in}}, \dots, N_L = N_{\text{out}}$, width $W := \max_{0 \leqslant \ell \leqslant L} |N_\ell|$, and each matrix having some operator norm bounded by $R \geqslant 1$, it holds [Gonon et al., 2023, Theorem III.1 with $p = q = \infty$ and $D = \|x\|_\infty$][Neyshabur et al., 2018, Berner et al., 2020]:

$$\|R_\theta(x) - R_{\theta'}(x)\|_1 \leqslant (W\|x\|_\infty + 1) W L^2 R^{L-1} \|\theta - \theta'\|_\infty.$$

Can it be retrieved from Theorem 3.1? Next corollary almost recovers it: with $W \max(\|x\|_\infty, 1)$ instead of $W\|x\|_\infty + 1$, and $2L$ instead of L^2. This is better as soon as there are at least $L \geqslant 2$ layers and as soon as the input satisfies $\|x\|_\infty \geqslant 1$.

Corollary F.1. *[Gonon et al., 2023, Theorem III.1] Consider a simple layered fully-connected neural network architecture with $L \geqslant 1$ layers, corresponding to functions $R_\theta(x) = M_L \,\mathrm{ReLU}(M_{L-1} \dots \mathrm{ReLU}(M_1 x))$ with each M_ℓ denoting a matrix, and parameters $\theta = (M_1, \dots, M_L)$. For a matrix M, denote by $\|M\|_{1,\infty}$ the maximum ℓ^1 norm of a row of M. Consider $R \geqslant 1$ and define the set Θ of parameters $\theta = (M_1, \dots, M_L)$ such that $\|M_\ell\|_{1,\infty} \leqslant R$ for every $\ell \in [\![1, L]\!]$. Then, for every parameters $\theta, \theta' \in \Theta$, and every input x:*

$$\|R_\theta(x) - R_{\theta'}(x)\|_1 \leqslant \max(\|x\|_\infty, 1) 2 L W^2 R^{L-1} \|\theta - \theta'\|_\infty.$$

Proof. For every neuron v, define $f(v) := \ell$ such that neuron v belongs to the output neurons of matrix M_ℓ (i.e., of layer ℓ). By Lemma E.1 with $q = 1$, we have for every neuron v

$$\begin{aligned}
&\|\Phi^{\to v}(\theta) - \Phi^{\to v}(\theta')\|_1 \\
&\leqslant \max_{p \in \mathcal{P}^{\to v}} \sum_{\ell=1}^{\texttt{length}(p)} \left(\prod_{k=\ell+1}^{\texttt{length}(p)} \underbrace{\|\theta^{\to p_k}\|_1}_{\leqslant \|M_{f(p_k)}\|_{1,\infty} \leqslant R} \right) \\
&\qquad \left(\underbrace{|b_{p_\ell} - b'_{p_\ell}|}_{=0 \text{ (no biases)}} + \underbrace{\|\theta^{\to p_\ell} - (\boldsymbol{\theta'})^{\to p_\ell}\|_1}_{\leqslant |\operatorname{ant}(p_\ell)| \|\theta - \theta'\|_\infty \leqslant W \|\theta - \theta'\|_\infty} \max_{u \in \operatorname{ant}(p_\ell)} \|\Phi^{\to u}(\theta')\|_1 \right) \qquad (23) \\
&\leqslant W \|\theta - \theta'\|_\infty \max_{p \in \mathcal{P}^{\to v}} \sum_{\ell=1}^{\texttt{length}(p)} R^{\texttt{length}(p) - \ell} \max_{u \in \operatorname{ant}(p_\ell)} \|\Phi^{\to u}(\theta')\|_1 \qquad (24)
\end{aligned}$$

with the convention that an empty sum and product are respectively equal to zero and one. Consider $\theta' = 0$. It holds $\|\Phi^{\to u}(\theta')\|_1 = 0$ for every $u \notin N_{\text{in}}$, and $\|\Phi^{\to u}(\theta')\|_1 = 1$ for input neurons u (Definition A.5). Therefore, we have:

$$\max_{u \in \operatorname{ant}(p_\ell)} \|\Phi^{\to u}(\theta')\|_1 = \mathbb{1}_{\operatorname{ant}(p_\ell) \cap N_{\text{in}} \neq \emptyset} = \mathbb{1}_{\ell = 1 \text{ and } p_0 \in N_{\text{in}}}. \qquad (25)$$

Specializing Equation (23) to $\theta' = 0$ and using Equation (25) yields

$$\|\Phi^{\to v}(\theta)\|_1 \leqslant \max_{p\in\mathcal{P}^{\to v}} \sum_{\ell=1}^{\texttt{length}(p)} \left(\prod_{k=\ell+1}^{\texttt{length}(p)} R\right) \underbrace{\|\theta^{\to p_\ell}\|_1}_{\substack{\leqslant \|M_{f(p_\ell)}\|_{1,\infty} \\ \leqslant R}} \underbrace{\max_{u\in\mathrm{ant}(p_\ell)} \|\Phi^{\to u}(\theta')\|_1}_{=\mathbb{1}_{\ell=1 \text{ and } p_0\in N_{\text{in}}}}$$

$$= \max_{p\in\mathcal{P}^{\to v}: p_0\in N_{\text{in}}} R^{\texttt{length}(p)}. \tag{26}$$

Since the network is layered, every neuron $u \in \mathrm{ant}(p_\ell)$ is on the $\ell-1$-th layer, and every $p' \in \mathcal{P}^{\to u}$ is of length $\ell-1$, hence we deduce using Equation (24), Equation (26) for θ' and u:

$$\|\Phi^{\to v}(\theta) - \Phi^{\to v}(\theta')\|_1 \leqslant W\|\theta-\theta'\|_\infty \max_{p\in\mathcal{P}^{\to v}} \sum_{\ell=1}^{\texttt{length}(p)} R^{\texttt{length}(p)-\ell} \underbrace{\max_{u\in\mathrm{ant}(p_\ell)} \max_{p'\in\mathcal{P}^{\to u}: p'_0\in N_{\text{in}}} R^{\texttt{length}(p')}}_{=R^{\ell-1}}$$

$$= W\|\theta-\theta'\|_\infty \max_{p\in\mathcal{P}^{\to v}} \underbrace{\sum_{\ell=1}^{\texttt{length}(p)} R^{\texttt{length}(p)-1}}_{\leqslant LR^{L-1}}$$

$$\leqslant LWR^{L-1}\|\theta-\theta'\|_\infty.$$

We get:

$$\begin{aligned}\|\Phi(\theta) - \Phi(\theta')\|_1 &= \sum_{v\in N_{\text{out}}\setminus N_{\text{in}}} \|\Phi^{\to v}(\theta) - \Phi^{\to v}(\theta')\|_1 \\ &\leqslant |N_{\text{out}} \setminus N_{\text{in}}| \cdot LWR^{L-1}\|\theta-\theta'\|_\infty \\ &\leqslant LW^2R^{L-1}\|\theta-\theta'\|_\infty.\end{aligned}$$

Using Corollary C.1 with $q=1$, we deduce that as soon as θ, θ' satisfy $\theta_i\theta'_i \geqslant 0$ for every parameter coordinate i, then for every input x:

$$\|R_\theta(x) - R_{\theta'}(x)\|_1 \leqslant \max(\|x\|_\infty, 1)LW^2R^{L-1}\|\theta-\theta'\|_\infty. \tag{27}$$

Now, consider general parameters θ and θ'. Define θ^{inter} to be such that for every parameter coordinate i:

$$\theta_i^{\text{inter}} = \begin{cases} \theta'_i & \text{if } \theta_i\theta'_i \geqslant 0, \\ 0 & \text{otherwise.}\end{cases}$$

By definition, it holds for every parameter coordinate i: $\theta_i^{\text{inter}}\theta_i \geqslant 0$ and $\theta_i^{\text{inter}}\theta'_i \geqslant 0$ so we can apply Equation (27) to the pairs $(\theta, \theta^{\text{inter}})$ and $(\theta^{\text{inter}}, \theta')$ to get:

$$\begin{aligned}\|R_\theta(x) - R_{\theta'}(x)\|_1 &\leqslant \|R_\theta(x) - R_{\theta^{\text{inter}}}(x)\|_1 + \|R_{\theta^{\text{inter}}}(x) - R_{\theta'}(x)\|_1 \\ &\leqslant \max(\|x\|_\infty, 1)LW^2R^{L-1}\left(\|\theta-\theta^{\text{inter}}\|_\infty + \|\theta^{\text{inter}}-\theta'\|_\infty\right).\end{aligned}$$

It remains to see that $\|\theta-\theta^{\text{inter}}\|_\infty + \|\theta^{\text{inter}}-\theta'\|_\infty 2\|\theta-\theta'\|_\infty$. Consider a parameter coordinate i. *If* $\theta_i\theta'_i \geqslant 0$ then $\theta_i^{\text{inter}} = \theta'_i$ and:

$$|\theta_i - \theta'_i| = |\theta_i - \theta_i^{\text{inter}}| + |\theta_i^{\text{inter}} - \theta'_i|.$$

Otherwise, $\theta_i^{\text{inter}} = 0$ and:

$$\begin{aligned}|\theta_i - \theta'_i| &= |\theta_i| + |\theta'_i| \\ &= |\theta_i - \theta_i^{\text{inter}}| + |\theta_i^{\text{inter}} - \theta'_i|.\end{aligned}$$

This implies $\|\theta-\theta^{\text{inter}}\|_\infty = \max_i |\theta_i - \theta_i^{\text{inter}}| \leqslant \max_i |\theta_i - \theta_i^{\text{inter}}| + |\theta_i^{\text{inter}} - \theta'_i| = \|\theta-\theta'\|_\infty$ and similarly $\|\theta^{\text{inter}}-\theta'\|_\infty \leqslant \|\theta-\theta'\|_\infty$. This yields the desired result:

$$\|R_\theta(x) - R_{\theta'}(x)\|_1 \leqslant \max(\|x\|_\infty, 1)2LW^2R^{L-1}\|\theta-\theta'\|_\infty. \qquad \square$$

G Proof of Theorem 5.1: Generalization bound

The goal of this section is to prove the bound on the generalization error given in Theorem 5.1. First, we recall the definition of the generalization error.

Definition G.1. *(Generalization error) Consider an architecture G (Definition A.2) with input and output dimensions d_{in} and d_{out}, and a so-called loss function $\ell : \mathbb{R}^{d_{out}} \times \mathbb{R}^{d_{out}} \to \mathbb{R}$. The* ℓ-generalization error *of parameters θ on a collection Z of $n \in \mathbb{N}_{>0}$ pairs of input/output $z_i = (x_i, y_i) \in \mathbb{R}^{d_{in}} \times \mathbb{R}^{d_{out}}$ and with respect to a probability measure μ on $\mathbb{R}^{d_{in}} \times \mathbb{R}^{d_{out}}$ is:*

$$\ell\text{-generalization error}(\theta, Z, \mu) := \underbrace{\mathbb{E}_{(\mathbf{X}_0, \mathbf{Y}_0)\sim\mu}\left(\ell\left(R_\theta(\mathbf{X}_0), \mathbf{Y}_0\right)\right)}_{\text{test error}} - \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell\left(R_\theta(x_i), y_i\right)}_{\text{training error when trained on } Z}.$$

We are going to bound the generalization error with covering numbers. We start by recalling the definition of covering numbers (see, e.g., Definition 5.5 in Van Handel [2014]).

Definition G.2. *Consider a pseudo-metric space (S, d). Let $B_r(x)$ the closed ball centered at $x \in S$ of radius $r > 0$. A family $x_1, \dots, x_n$ of points of S is called an r-covering of (S, d) if $S \subset \cup_{i=1}^n B_r(x_i)$. The covering number of (S, d) for radius $r > 0$, denoted $\mathcal{N}(S, d, t)$, is the minimum cardinality of an r-covering of (S, d).*

We now state a result that uses very classical arguments to bound the generalization error by Dudley's integral Shalev-Shwartz and Ben-David [2014], Van Handel [2014], Maurer [2016]. It is valid for an arbitrary class of functions (not only neural networks). We will then bound Dudley's integral using arguments specific to neural networks (notably Theorem 3.1).

Theorem G.1. *Consider a set $\mathcal{F} := \{R_\theta, \theta \in \Theta\}$ of measurable functions from $\mathbb{R}^{d_{in}}$ to $\mathbb{R}^{d_{out}}$ parameterized by an arbitrary set Θ. Consider a loss function $\ell : \mathbb{R}^{d_{out}} \times \mathbb{R}^{d_{out}} \to \mathbb{R}$ such that*

$$\ell(\hat{y}_1, y) - \ell(\hat{y}_2, y) \leqslant L\|\hat{y}_1 - \hat{y}_2\|_2, \quad \forall y, \hat{y}_1, \hat{y}_2 \in \text{support}(\mathbf{Y}_1), \tag{28}$$

for some $L > 0$. Consider a probability measure μ on the pairs of input/output $\mathbb{R}^{d_{in}} \times \mathbb{R}^{d_{out}}$. Consider $n+1$ iid random variables $\mathbf{Z}_i = (\mathbf{X}_i, \mathbf{Y}_i) \sim \mu$, $0 \leqslant i \leqslant n$, and denote $\mathbf{Z} = (\mathbf{Z}_i)_{i=1,\dots,n}$. Define the pseudo-metric $d_{\mathbf{X}}$ on Θ by:

$$d_{\mathbf{X}}(\theta, \theta')^2 := \|R(\theta, \mathbf{X}) - R(\theta', \mathbf{X})\|_2^2 = \sum_{v \in N_{out}} \sum_{i=1}^{n} |v(\theta, \mathbf{X}_i) - v(\theta', \mathbf{X}_i)|^2 \tag{29}$$

Then[8] for any estimator $\hat{\theta} : \mathbf{Z} \mapsto \hat{\theta}(\mathbf{Z}) \in \Theta$ (recalling the definition of a covering number in Definition G.2)

$$\mathbb{E}_{\mathbf{Z}}\ell\text{-generalization error}(\hat{\theta}(\mathbf{Z}), \mathbf{Z}, \mu) \leqslant \frac{24\sqrt{2}L}{n}\mathbb{E}_{\mathbf{X}}\left(\int_0^\infty \sqrt{\ln \mathcal{N}(\Theta, d_{\mathbf{X}}, t)}dt\right). \tag{30}$$

Proof of Theorem 5.1. **1st step: control the generalization error by the Rademacher complexity.**

Consider a family $(\boldsymbol{\varepsilon}_j)_{j\in J}$, with J that will be clear from the context, of iid Rademacher random variables (meaning that $\mathbb{P}(\boldsymbol{\varepsilon}_j = 1) = \mathbb{P}(\boldsymbol{\varepsilon}_j = -1) = 1/2$). Denote by $[\![n]\!] \times N_{\text{out}}$ (where $[\![n]\!] := \{1, \dots, n\}$) and define the random matrices $E = (\boldsymbol{\varepsilon}_{i,v})_{i,v} \in \mathbb{R}^{[\![n]\!]\times N_{\text{out}}}$ and $R(\theta, \mathbf{X}) = (v(\theta, \mathbf{X}_i))_{i,v} \in \mathbb{R}^{[\![n]\!]\times N_{\text{out}}}$ so that $\langle E, R(\theta, \mathbf{X})\rangle = \sum_{i,v} \boldsymbol{\varepsilon}_{i,v} v(\theta, \mathbf{X}_i)$. It then holds:

$$\begin{aligned}\mathbb{E}_{\mathbf{Z}}\ \ell\text{-generalization error of } \hat{\theta}(\mathbf{Z}) &\leqslant \frac{2}{n}\mathbb{E}_{\mathbf{Z},\boldsymbol{\varepsilon}}\left(\sup_\theta \sum_{i=1}^{n} \boldsymbol{\varepsilon}_i \ell\left(R_\theta(\mathbf{X}_i), \mathbf{Y}_i\right)\right) \\ &\leqslant \frac{2\sqrt{2}L}{n}\mathbb{E}_{\mathbf{Z},\boldsymbol{\varepsilon}}\left(\sup_\theta \langle E, R(\theta, \mathbf{X})\rangle\right).\end{aligned}$$

[8] The definition of the generalization error (Definition G.1) has been given for deterministic θ to keep things simple. The careful reader will have noted that the term corresponding to the test error in Definition G.1 has to be modified when θ is a function of $\mathbf{Z}$. Indeed, the expectation has to be taken on an iid copy $\mathbf{Z}_0$ conditionally on each $\mathbf{Z}_i$, $i = 1, \dots, n$: the test error should be defined as $\mathbb{E}_{\mathbf{Z}_0\sim\mu}\left(\ell(R_{\hat{\theta}(\mathbf{Z})}(\mathbf{X}_0), \mathbf{Y}_0)|\mathbf{Z}\right)$. This correct definition will be used in the proof, but it has no importance here to understand the statement of the theorem.

The first inequality is the symmetrization property given by Shalev-Shwartz and Ben-David [2014, Theorem 26.3], and the second inequality is the vector-valued contraction property given by Maurer [2016]. These are the relevant versions of very classical arguments that are widely used to reduce the problem to the Rademacher complexity of the model [Bach, 2024, Propositions 4.2 and 4.3][Wainwright, 2019, Equations (4.17) and (4.18)][Bartlett and Mendelson, 2002, Proof of Theorem 8][Shalev-Shwartz and Ben-David, 2014, Theorem 26.3][Ledoux and Talagrand, 1991, Equation (4.20)]. Note that the assumption on the loss is used for the second inequality.

2nd step: sub-Gaussianity.

We consider the characterization of sub-Gaussianity given in Definition 5.20 of Van Handel [2014]: a real random process $(\mathbf{N}_t)_{t\in T}$ is sub-Gaussian on the pseudo-metric space (T,d) (recall that a pseudo-metric does not necessarily separate points, that is $d(s,t)=0$ does not imply $s=t$) if it is centered and if:

$$\mathbb{E}\left(\exp\left(\lambda(\mathbf{N}_t-\mathbf{N}_s)\right)\right)\leqslant\exp\left(\frac{\lambda^2 d(s,t)^2}{2}\right),\forall\lambda>0,\forall s,t\in T.$$

Consider the real random process $\mathbf{S}=(\mathbf{S}_\theta)_{\theta\in\Theta}$ defined by

$$\mathbf{S}_\theta=\langle E,R(\theta,\mathbf{X})\rangle\,.$$

We now establish that conditionally on $\mathbf{X}$, the random process $\mathbf{S}$ is sub-Gaussian on the pseudo-metric space $(\Theta,d_{\mathbf{X}})$ (where $d_{\mathbf{X}}$ is defined in Equation (29)). The process is centered since $\mathbb{E}\left(\mathbf{S}_\theta|\mathbf{X}\right)=\langle\mathbb{E}(E),R(\theta,\mathbf{X})\rangle=0$ for every $\theta\in\Theta$. Now, consider $\theta,\theta'\in\Theta$, $(i,v)\in[\![n]\!]\times N_{\text{out}}$ and denote by $d_{i,v}=v(\theta,\mathbf{X}_i)-v(\theta',\mathbf{X}_i)$. For any $t>0$, it holds $\frac{1}{2}(e^t+e^{-t})\leqslant e^{t^2/2}$ so for every $\lambda>0$:

$$\mathbb{E}\left(\exp\left(\lambda\boldsymbol{\varepsilon}_{i,v}d_{i,v}\right)|\mathbf{X}\right)=\frac{1}{2}\left(\exp(\lambda d_{i,v})+\exp(-\lambda d_{i,v})\right)\leqslant\exp\left(\frac{\lambda^2 d_{i,v}^2}{2}\right).$$

Thus, we have

$$\mathbb{E}\left(\exp\left(\lambda(\mathbf{S}_\theta-\mathbf{S}_{\theta'})\right)|\mathbf{X}\right)=\prod_{(i,v)\in[\![n]\!]\times N_{\text{out}}}\mathbb{E}\left(\exp\left(\lambda\boldsymbol{\varepsilon}_{i,v}d_{i,v}\right)|\mathbf{X}\right)\leqslant\exp\left(\frac{\lambda^2 d_{\mathbf{X}}(\theta,\theta')^2}{2}\right).$$

This shows the claim about the sub-Gaussianity of $\mathbf{S}$.

3rd step: Dudley's inequality.

Using Dudley's integral inequality [Van Handel, 2014, Corollary 5.25] conditionally on $\mathbf{X}$ yields almost surely:

$$\mathbb{E}_{\boldsymbol{\varepsilon}}\sup_{\theta\in\Theta}S_\theta\leqslant 12\int_0^\infty\sqrt{\ln\mathcal{N}(\Theta,d_{\mathbf{X}},t)}dt$$

where $\mathbb{E}_{\boldsymbol{\varepsilon}}$ denotes the expectation conditioned on everything (here $\mathbf{X}$) except $\boldsymbol{\varepsilon}$. Putting all the first three steps together, we get:

$$\mathbb{E}_{\mathbf{Z}}\,\ell\text{-generalization error of }\hat{\theta}(\mathbf{Z})\leqslant\frac{24\sqrt{2}L}{n}\mathbb{E}_{\mathbf{X}}\left(\int_0^\infty\sqrt{\ln\mathcal{N}(\Theta,d_{\mathbf{X}},t)}dt\right).$$

□

We now get specifc to neural networks. The main ingredient is Theorem 3.1.

Lemma G.1 (Bounding Dudley's integral with covering numbers of $(\Phi(\Theta),\|\cdot\|_1)$)**.** *Consider a ReLU neural network architecture G (Definition A.2) and a set $\Theta\subset\mathbb{R}^G$ of parameters associated to this architecture. Denote by $r=\sup_{\theta\in\Theta}\|\Phi(\theta)\|_1$, Θ^* the set of parameters with only nonzero coordinates, $\mathcal{S}=\{\operatorname{sgn}(\theta),\theta\in\Theta^*\}$ the associated set of sign vectors (with $\operatorname{sgn}(x)=\mathbb{1}_{x\geqslant 0}-\mathbb{1}_{x\leqslant 0}\in\{-1,0,1\}$), and for each $s\in\mathcal{S}$ denote $\Theta_s=\Theta\cap\{\theta:\theta_i s_i\geqslant 0,\forall i\}$. For $t>0$, define*

$$f(t):=\max_{s\in\mathcal{S}}\mathcal{N}(\Phi(\Theta_s),\|\cdot\|_1,t).\tag{31}$$

Consider n inputs $x_1,\dots,x_n$ of G. Define $\sigma_X=\left(\sum_{i=1}^n\max(1,\|x_i\|_\infty^2)\right)^{1/2}$ and consider the pseudo metric $d_X(\theta,\theta'):=\left(\sum_{i=1}^n\|R_\theta(x_i)-R_{\theta'}(x_i)\|_2^2\right)^{1/2}$. Then, it holds (recall the definition of a covering number in Definition G.2):

$$\int_0^\infty\sqrt{\ln\mathcal{N}(\Theta,d_X,t)}dt\leqslant 2r\sigma_X\sqrt{\ln|\mathcal{S}|}+\sigma_X\int_0^{2r}\sqrt{\ln(f(u))}\,du.\tag{32}$$

Proof. For any parameters θ and input x, it holds

$$\|R_\theta(x)\|_2 = \|R_\theta(x) - R_0(x)\|_2 \underset{\text{Corollary C.1}}{\leqslant} \max(1, \|x\|_\infty)\|\Phi(\theta) - \Phi(0)\|_1 = \max(1, \|x\|_\infty)\|\Phi(\theta)\|_1.$$

Recall that $r = \sup_{\theta \in \Theta} \|\Phi(\theta)\|_1$. Then for every $\theta, \theta' \in \Theta$ and every input x, it holds

$$\|R_\theta(x) - R_{\theta'}(x)\|_2 \leqslant \|R_\theta(x)\|_2 + \|R_{\theta'}(x)\|_2 \leqslant 2\max(1, \|x\|_\infty) r.$$

Since $\sigma_X^2 = \sum_{i=1}^n \max(1, \|x_i\|_\infty^2)$, we have:

$$d_X(\theta, \theta')^2 = \sum_{i=1}^n \|R_\theta(x_i) - R_{\theta'}(x_i)\|_2^2 \leqslant 4\sigma_X^2 r^2.$$

This shows that any single vector θ of Θ is a $2\sigma_X r$-covering of this set with respect to d_X. Thus, we have $\mathcal{N}(\Theta, d_X, t) = 1$ for $t \geqslant 2\sigma_X r$ so that:

$$\int_0^\infty \sqrt{\ln \mathcal{N}(\Theta, d_X, t)} dt = \int_0^{2\sigma_X r} \sqrt{\ln \mathcal{N}(\Theta, d_X, t)} dt.$$

We now bound the covering number for a general $t \geqslant 0$ using Theorem 3.1. Recall that Θ^* is the set of parameters with only nonzero coordinates, $\mathcal{S} = \{\text{sgn}(\theta), \theta \in \Theta^*\}$ is the associated set of sign vectors, and for each $s \in \mathcal{S}$, $\Theta_s = \Theta \cap \{\theta : \theta_i s_i \geqslant 0, \forall i\}$. Therefore, $\Theta = \cup_{s \in \mathcal{S}} \Theta_s$ and the union of t-coverings of each Θ_s is a t-covering of Θ. So for each $t > 0$ we have

$$\mathcal{N}(\Theta, d_X, t) \leqslant \sum_{s \in \mathcal{S}} \mathcal{N}(\Theta_s, d_X, t)$$

Theorem 3.1 implies (through Corollary C.1) that for each s, and every $\theta, \theta' \in \Theta_s$:

$$d_X(\theta, \theta') = \left(\sum_{i=1}^n \|R_\theta(x_i) - R_{\theta'}(x_i)\|_2^2 \right)^{1/2} \leqslant \sigma_X \|\Phi(\theta) - \Phi(\theta')\|_1,$$

Thus, picking an abritrary pre-image by Φ of a t/σ_X-covering of $\Phi(\Theta_s)$ for the ℓ^1-norm yields a t-covering of Θ_s for the pseudo-metric d_X, so that

$$\mathcal{N}(\Theta_s, d_X, t) \leqslant \mathcal{N}\left(\Phi(\Theta_s), \|\cdot\|_1, t/\sigma_X\right) \leqslant \underbrace{\max_{s \in \mathcal{S}} \mathcal{N}(\Phi(\Theta_s), \|\cdot\|_1, t/\sigma_X)}_{= f(t/\sigma_X) \text{ (Equation (31))}}$$

and thus

$$\begin{aligned}\int_0^{2\sigma_X r} \sqrt{\ln \mathcal{N}(\Theta, d_X, t)}\, dt &\leqslant \int_0^{2\sigma_X r} \sqrt{\ln\left(|\mathcal{S}| f(t/\sigma_X)\right)}\, dt \\ &\overset{u = t/\sigma_X}{=} \sigma_X \int_0^{2r} \sqrt{\ln\left(|\mathcal{S}| f(u)\right)}\, du \leqslant 2r\sigma_X \sqrt{\ln |\mathcal{S}|} + \sigma_X \int_0^{2r} \sqrt{\ln(f(u))}\, du. \quad \square\end{aligned}$$

For the parameter set $\Theta = \Theta(r) := \{\theta \in \mathbb{R}^G, \|\Phi(\theta)\|_1 \leqslant r\}$, and other similar parameter sets with weight-sharing (associated e.g. to convolution layers), it is enough to study the covering numbers associated with the positive orthant: Θ_s with $s = \mathbf{1}$.

Lemma G.2. *Consider the setting of Lemma G.1. Denote by $\mathbf{1}$ the vector constant equal to one and by $|\theta| \in \Theta_{\mathbf{1}}$ the vector deduced from $\theta \in \Theta_s$ by applying $x \mapsto |x|$ coordinate-wise. If for every $s \in \mathcal{S}$, the map $x \in \Theta_s \mapsto s \odot |x| \in \Theta_s$ is one-to-one (with inverse $x \in \Theta_{\mathbf{1}} \mapsto s \odot x \in \Theta_s$), then*

$$\mathcal{N}(\Phi(\Theta_s), \|\cdot\|_1, t) = \mathcal{N}(\Phi(\Theta_{\mathbf{1}}), \|\cdot\|_1, t).$$

Proof. For every $\theta, \theta' \in \Theta_s$, it is easy to check that by definition (Definition A.5) $\|\Phi(\theta) - \Phi(\theta')\|_1 = \|\Phi(|\theta|) - \Phi(|\theta'|)\|_1$. This shows that under the assumptions, there is a one-to-one correspondence between the t-coverings of $(\Phi(\Theta_s), \|\cdot\|_1)$ and of $(\Phi(\Theta_{\mathbf{1}}), \|\cdot\|_1)$. $\square$

When covering a set in dimension d, the covering numbers typically grow exponentially with d. In our case, $\Phi(\Theta)$ lives in a space indexed by the paths, but the actual degree of freedom expected is the dimension of Θ, which is much less in general. Moreover, in many practical cases of interest, Θ has often weight sharing. Is it possible to bound these covering numbers exponentially in $d :=$ the number of free parameters, taling into account possible weight sharing? The next example shows that this is indeed possible in some situations.

Example G.1. *Consider the model $R_\theta : x \in \mathbb{R}^d \mapsto W\,\mathrm{ReLU}(W^T x) \in \mathbb{R}^d$ with $\theta = (W, W^T)$, $W = (w_1 \dots w_d) \in \mathbb{R}^{d\times d}$, and each column w_i being in $\mathbb{R}^d$. In this case, there are $2d^2$ coordinates in θ, but only d^2 of them are free. The path-lifting is $\Phi(\theta) = (w_i \otimes w_i)_{i=1,\dots,d} \in \mathbb{R}^{d^3}$ (flattened) where $u \otimes v = uv^T$ is the tensor product of vectors u and v. Consider $r > 0$ and $\Theta = \Theta(r) := \{\theta \in \mathbb{R}^G, \|\Phi(\theta)\|_1 \leqslant r\}$. For parameters $\theta = (W, W^T)$, its normalized version $\mathtt{N}(\theta)$ defined as the output of Algorithm 1 in [Gonon et al., 2024], reproduced in Algorithm 1 for convenience, satisfies $\mathtt{N}(\theta) = (\mathtt{N}(W), \mathtt{N}(W^T))$ where $\mathtt{N}(W) := (\frac{w_1}{\|w_1\|_1} \dots \frac{w_d}{\|w_d\|_1})$ and $\mathtt{N}(W^T) := (\|w_1\|_1^2 \frac{w_1}{\|w_1\|_1} \dots \|w_d\|_1^2 \frac{w_d}{\|w_d\|_1})^T$. Fix the parameters θ and $t \in (0, \min(12, r)]$. Consider the problem of finding θ' such that $\|\Phi(\theta) - \Phi(\theta')\|_1 \leqslant t$. Consider a t-covering of the unit sphere in dimension d for the ℓ^1-norm of cardinal at most equal to $(12/t)^{d-1}$ (see the end of Appendix H for the existence of such a covering). For every $i = 1, \dots, d$, choose u_i in this covering in such a way that $\|\mathtt{N}(w_i) - u_i\|_1 \leqslant t$ where we denote $\mathtt{N}(w) := \frac{w}{\|w\|_1}$ for any vector w. Consider also $r_i = \sqrt{\lfloor \|w_i\|_1^2/rt \rfloor}\, t$. Since $\|\Phi(\theta)\|_1 = \sum_{i=1}^d \|w_i\|_1^2 \leqslant r$, we have $\|w_i\|_1^2/r \leqslant 1$ and there are at most $\lfloor \frac{1}{t} \rfloor + 1 \leqslant \frac{12}{t}$ possible values for r_i if we further restrict $t \in (0, \min(11, r))$. Define $w_i' := r_i u_i$. This results in at most $(\frac{12}{t})^d$ possible values for w_i'. Since θ' is built from d vectors w_i' that can be chosen independently of each other, there are at most $\prod_{i=1}^d (\frac{12}{t})^d = (\frac{12}{t})^{d^2}$ choices for θ'. Since $\mathtt{N}(w_i') = u_i$, it holds that $\|\mathtt{N}(w_i) - \mathtt{N}(w_i')\|_1 \leqslant t$. Moreover, we have $|r_i^2 - \|w_i\|_1^2| \leqslant rt$. We deduce that:*

$$
\begin{aligned}
\|\Phi(\theta) - \Phi(\theta')\|_1 &= \sum_{i=1}^d \|w_i \otimes w_i - w_i' \otimes w_i'\|_1 = \sum_{i=1}^d \left\| \|w_i\|_1^2 \mathtt{N}(w_i) \otimes \mathtt{N}(w_i) - \|w_i'\|_1^2 \mathtt{N}(w_i') \otimes \mathtt{N}(w_i') \right\|_1 \\
&\leqslant \sum_{i=1}^d \left(\|w_i\|_1^2 \|\mathtt{N}(w_i) \otimes (\mathtt{N}(w_i) - \mathtt{N}(w_i'))\|_1 + \left\| (\|w_i\|_1^2 \mathtt{N}(w_i) - \|w_i'\|_1^2 \mathtt{N}(w_i')) \otimes \mathtt{N}(w_i') \right\|_1 \right) \\
&= \sum_{i=1}^d \left(\|w_i\|_1^2 \underbrace{\|\mathtt{N}(w_i)\|_1}_{=1} \|\mathtt{N}(w_i) - \mathtt{N}(w_i')\|_1 + \left\| \|w_i\|_1^2 \mathtt{N}(w_i) - \|w_i'\|_1^2 \mathtt{N}(w_i') \right\|_1 \underbrace{\|\mathtt{N}(w_i')\|_1}_{=1} \right) \\
&= \sum_{i=1}^d \left(\|w_i\|_1^2 \|\mathtt{N}(w_i) - \mathtt{N}(w_i')\|_1 + \left\| \|w_i\|_1^2 \mathtt{N}(w_i) - \|w_i'\|_1^2 \mathtt{N}(w_i') \right\|_1 \right) \\
&= \sum_{i=1}^d \left(\|w_i\|_1^2 \|\mathtt{N}(w_i) - \mathtt{N}(w_i')\|_1 + \left\| \|w_i\|_1^2 \mathtt{N}(w_i) - \|w_i\|_1^2 \mathtt{N}(w_i') + \|w_i\|_1^2 \mathtt{N}(w_i') - \|w_i'\|_1^2 \mathtt{N}(w_i') \right\|_1 \right) \\
&\leqslant \sum_{i=1}^d \left(\|w_i\|_1^2 \|\mathtt{N}(w_i) - \mathtt{N}(w_i')\|_1 + \|w_i\|_1^2 \|\mathtt{N}(w_i) - \mathtt{N}(w_i')\|_1 + \left| \|w_i\|_1^2 - \|w_i'\|_1^2 \right| \underbrace{\|\mathtt{N}(w_i')\|_1}_{=1} \right) \\
&= \sum_{i=1}^d \left(2\|w_i\|_1^2 \underbrace{\|\mathtt{N}(w_i) - \mathtt{N}(w_i')\|_1}_{\leqslant t} + \underbrace{\left| \|w_i\|_1^2 - \|w_i'\|_1^2 \right|}_{\leqslant rt} \right) \leqslant \underbrace{\left(\sum_{i=1}^d \|w_i\|_1^2 \right)}_{\leqslant r} 2t + drt \leqslant (d+2)rt.
\end{aligned}
$$

This shows that if we replace t by $\frac{t}{(d+2)r}$, we get a t-covering of $\Phi(\Theta)$ in the ℓ^1-norm of size at most equal to $\left(\frac{12(d+2)r}{t}\right)^{d^2}$. **In this situation, when $t > 0$ goes to zero, the covering number essentially grows exponentially with d^2, that is the number of free cordinates, rather than $2d^2$ the number of total coordinates after weight sharing.**

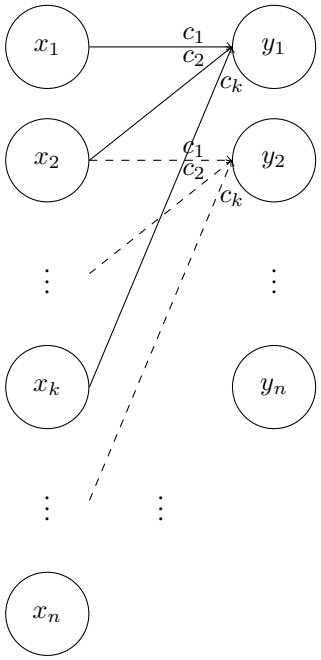


Figure 6: Illustration of a convolutional circular layer with kernel size k as described in Example G.2. The connections corresponding to the first row of the matrix C are drawn as plain arrows, the ones corresponding to the second row are drawn as dashed arrows.

In the previous example, some weights are shared across *successive* layers of the networks. In contrast, most practical application share weights in the same (convolutional) layer, that leads to very different properties of $\Phi(\theta)$: a path cannot contain several copies of a same weight, in contrast to the previous example. We now prove that for usual layered feedforward networks, it is still possible to control the covering numbers by taking into account weight sharing.

Definition G.3. *Consider a DAG $G = (N, E)$. A partition $N = \cup_{\ell=0}^{L} N_\ell$ of the neurons is said to be* directed *if for every $k \leqslant \ell$, we have $E \cap (N_\ell \times N_k) = \emptyset$ (no edge going from N_ℓ to N_k). It is said* regular *if for every $k < \ell$, every $u, v \in N_\ell$, it holds $|\operatorname{ant}(u) \cap N_k| = |\operatorname{ant}(v) \cap N_k|$ (same number of antecedents in N_k).*

Example G.2.

- *Every DAG admits at least one directed and regular partition. Indeed, consider any topological sorting $v_1, \dots, v_L$ of the neurons. The partition defined by $N_\ell := \{v_\ell\}$ for every $\ell = 1, \dots, L$ is both directed and regular.*
- *Consider a graph with a single (circular) convolutional layer with kernel size k as in Figure 6, corresponding to a circulant matrix*

$$C = \begin{pmatrix} c_1 & c_2 & \cdots & c_k & 0 & \dots & 0 \\ 0 & c_1 & \ddots & & c_k & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & & c_k \\ \vdots & & \dots & \ddots & \ddots & \ddots & \vdots \\ c_3 & \ddots & \dots & & \ddots & c_1 & c_2 \\ c_2 & c_3 & \cdots & 0 & 0 & 0 & c_1 \end{pmatrix}$$

With $N_0 := \{x_1, \dots, x_n\}$ and $N_1 := \{y_1, \dots, y_n\}$ the sets of input and output neurons of this layer, the partition $N = N_0 \cup N_1$ is directed and regular: by definition of the kernel size, every $u \in N_1$ satisfies $|\operatorname{ant}(u) \cap N_0| = |\operatorname{ant}(u)| = k$.

- *With the previous example, it is easy to see that for a neural network organized in $L+1$ layers of neurons, a directed and regular partition of the neurons is given by $N_0, \dots, N_L$ where N_ℓ is the set of neurons in layer ℓ.*

Definition G.4. *Consider a directed regular partition $N_0, \dots, N_L$ of a graph G. A set of parameters $\Theta \subset \mathbb{R}^G$ associated with G is said to be* weight-sharing compatible *with $N_0, \dots, N_L$ if for every $0 \leqslant k < \ell \leqslant L$, every pair of neurons $u, v \in N_\ell$ shares weights and biases in the following sense:*

- $b_u = b_v$
- *there exists a bijection $\sigma_{uv} : \mathrm{ant}(u) \cap N_k \to \mathrm{ant}(v) \cap N_k$ such that for every $\theta \in \Theta$, every $w \in \mathrm{ant}(u) \cap N_k$, $\theta^{w \to u} = \theta^{\sigma_{uv}(w) \to v}$.*

Example G.3. *The set of parameters corresponding to all circular matrices C as in Example G.2 is weight-sharing compatible with the partition given in Example G.2 in this case.*

For a fully-connected layer, denote by N_{in} the input neurons and $v_1, \dots, v_d$ an enumeration of the output neurons. The set of parameters is weight-sharing compatible with the directed regular partition given by $N_0 = N_{in}$ and $N_i = \{v_i\}$ for $i = 1, \dots, d$. Note that the set of parameters is not *weight-sharing compatible with the directed regular partition $N_0 = N_{in}$ and $N_1 = \{v_1, \dots, v_d\}$ because the neurons v_i do not share the same weights and cannot be gathered in the same set N_1 of the partition.*

For convenience, we recall in Algorithm 1 the Algorithm 1 given in Gonon et al. [2023] in the specific case of the ℓ^1-norm that we consider here.

Algorithm 1 Normalization of parameters for the ℓ^1-norm

Consider a topological sorting $v_1, \dots, v_k$ of the neurons
for $v = v_1, \dots, v_k$ **do**
  if $v \notin N_{\text{in}} \cup N_{\text{out}}$ **then**
    $\lambda_v \leftarrow \left\| \begin{pmatrix} \theta^{\to v} \\ b_v \end{pmatrix} \right\|_1$
    if $\lambda_v = 0$ **then**
      $\theta^{v \to} \leftarrow 0$
    else
      $\begin{pmatrix} \theta^{\to v} \\ b_v \end{pmatrix} \leftarrow \frac{1}{\lambda_v} \begin{pmatrix} \theta^{\to v} \\ b_v \end{pmatrix}$ ▷ normalize incoming weights and bias
      $\theta^{v \to} \leftarrow \lambda_v \times \theta^{v \to}$ ▷ rescale outgoing weights to preserve the function R_θ

Algorithm 1 introduces for each neuron $u \in N \setminus (N_{\text{in}} \cup N_{\text{out}})$ and each θ a normalizing scalar $\lambda_v(\theta)$ defined at the moment where u is processed in the for loop of Algorithm 1. The next lemma shows that this normalizing scalr is the same for all the neurons in a given set N_ℓ of a directed regular partition with weight-sharing.

Lemma G.3. *Consider a set of parameters $\Theta \subset \mathbb{R}^G$ weight-sharing compatible with a directed regular partition $N_0, \dots, N_L$ of a DAG G. It holds:*

$$\lambda_u(\theta) = \lambda_v(\theta), \forall \theta \in \Theta, \forall u, v \in N_\ell \setminus (N_{in} \cup N_{out}), \forall \ell \in [\![0, L]\!].$$

Proof. The proof is by induction on L.

Initialization. For $L = 0$, since the partition is directed, there is no edge going from N_0 to N_0 so all neurons are input ones and there is nothing to check. Therefore, the property is trivially true.

Induction. Assume this is true for $L \geqslant 0$ and consider the case $L+1$. Since the partition is directed, the neurons in $N_0, \dots, N_L$ are normalized by Algorithm 1 in the same way, irrespectively of whether we consider the graph G or its maximal subgraph with neurons restricted to the sets $N_0, \dots, N_L$. This shows the desired property for every $\ell \leqslant L$. It remains to consider $\ell = L+1$. Take $\theta \in \Theta$. We just saw that when normalizing θ with Algorithm 1, for every $k \leqslant L$, all the neurons in N_k have the same normalization scalar: denote it by $\lambda_k(\theta)$. Denote also $\theta^{N_k \to u} := (\theta^{w \to u})_{w \in \mathrm{ant}(u) \cap N_k}$. Since the partition is directed, and since the neurons are normalized in the order of a topological

sorting, Algorithm 1 normalizes the neurons $u \in N_{L+1}$ only after having normalized all the ones in $N_0, \dots, N_L$. Therefore, when normalizing $u \in N_{L+1}$, we have

$$\lambda_u(\theta) = |b_u| + \sum_{k \leqslant L} \lambda_k(\theta) \|\theta^{N_k \to u}\|_1.$$

Consider $u, v \in N_{L+1}$. Since Θ is weight-sharing compatible, we have $b_u = b_v$, and $\|\theta^{N_k \to u}\|_1 = \|\theta^{N_k \to v}\|_1$. This proves that $\lambda_u(\theta) = \lambda_v(\theta)$ and concludes the induction. □

An easy consequence of Lemma G.3 is that the neurons in the same set N_ℓ of a directed regular partition with weight-sharing must be normalized in the same way by Algorithm 1. This is because they share the same weights before normalization, and have the same normalization scalar according to Lemma G.3.

Corollary G.1. *In the context of Lemma G.3, consider $\ell \in [\![0, L]\!]$ and assume that either $N_\ell \cap V = \emptyset$ or that $N_\ell \subset V$ for both $V = N_{in}$ and N_{out}. For $\theta \in \Theta$, denote $\mathtt{N}(\theta)$ its normalized version, obtained as the output of Algorithm 1 in Gonon et al. [2024] on input θ (see Algorithm 1). It holds for every $u, v \in N_\ell \setminus N_{in}$:*

$$\mathtt{N}(b)_u = \mathtt{N}(b)_v,$$
$$\mathtt{N}(\theta)^{w \to u} = \mathtt{N}(\theta)^{\sigma_{uv}(w) \to v}, \forall w \in \mathrm{ant}(u).$$

Proof. The assumption guarantees that all neurons in N_ℓ are updated in the same way by the normalizing algorithm (Algorithm 1).

Case $N_\ell \subset N_{in}$. There is nothing to prove.

Case $N_\ell \subset N_{out}$. When u is an output neuron, b_u is not modified by Algorithm 1 so $\mathtt{N}(b)_u = b_u$. Moreover, for every $w \in \mathrm{ant}(u)$, the last time $\theta^{w \to u}$ is modified is when w is considered in Algorithm 1, so:

$$\mathtt{N}(\theta)^{w \to u} = \lambda_w(\theta) \theta^{w \to u}.$$

It is easy to conclude using weight-sharing (Definition G.4) and Lemma G.3.

Case $N_\ell \cap (N_{in} \cup N_{out}) = \emptyset$. All neurons $u \in N \setminus (N_{\text{in}} \cup N_{\text{out}})$ are such that the last time b_u and $\theta^{w \to u}$ ($w \in \mathrm{ant}(u)$) are modified by Algorithm 1 is when u is being considered in the `for` loop, so it holds:

$$\mathtt{N}(b)_u = \frac{1}{\lambda_u(\theta)} b_u,$$
$$\mathtt{N}(\theta)^{w \to u} = \frac{1}{\lambda_u(\theta)} \theta^{w \to u}.$$

We again conclude using weight-sharing (Definition G.4) and Lemma G.3. □

We now use this to cover the set $(\Phi(\Theta), \|\cdot\|_1)$ for a set of parameters Θ that has weight-sharing. This results in the following generalization bound.

Theorem G.2. *Consider iid $\mathbf{X}_1, \dots, \mathbf{X}_n$ random inputs of G. Denote $\sigma_X = \mathbb{E}_{\mathbf{X}} \left(\sum_{i=1}^n \max(1, \|\mathbf{X}_i\|_\infty^2) \right)^{1/2}$.*

Consider a set of parameters Θ weight-sharing compatible (Definition G.4) with a directed regular partition $N_0, \dots, N_L$ (Definition G.3) of a DAG G (Definition A.2). Assume that for $V = N_{in}$ and $V = N_{out}$, each N_ℓ is either disjoint from V or is a subset of V. Assume also that $N_{in} \cap N_{out} = \emptyset$. Define L_0 to be the unique integer in $[\![0, L]\!]$ such that $N_{L_0} \subset N_{in}$ and $N_{L_0+1} \cap N_{in} = \emptyset$. For $\ell \in [\![L_0, L]\!]$, denote by $k_\ell := |\mathrm{ant}(u)| = \sum_{j<\ell} |\mathrm{ant}(u) \cap N_j|$ for $u \in N_\ell$, the common number of antecedents of the neurons in N_ℓ. Define # rescalings $:= L - L_0$ and #params $:= \sum_{\ell=L_0}^{L} (k_\ell + 1)$. Recall that $D = \max_{p \in \mathcal{P}} \mathtt{length}(p)$ is the depth of the graph and $d_{out} = |N_{out}|$ is the output dimension. Denote by $r := \sup_{\theta \in \Theta} \|\Phi(\theta)\|_1$. It holds:

$$\mathbb{E}_{\mathbf{Z}} \, \ell\text{-generalization error of } \hat{\theta}(\mathbf{Z}) \leqslant 544 \frac{\sigma_X}{n} L \max(D, d_{out}) \sqrt{\#\text{params}} \times r. \tag{33}$$

Proof. Consider Θ^* the set of parameters with only nonzero coordinates, $\mathcal{S} = \{\mathrm{sgn}(\theta), \theta \in \Theta^*\}$ the associated set of sign vectors (with $\mathrm{sgn}(x) = \mathbb{1}_{x \geqslant 0} - \mathbb{1}_{x \leqslant 0} \in \{-1, 0, 1\}$), and for each $s \in \mathcal{S}$ denote $\Theta_s = \Theta \cap \{\theta : \theta_i s_i \geqslant 0, \forall i\}$. Theorem G.1, Lemma G.1 and Lemma G.2 guarantee altogether that for (recall the definition of a covering number in Definition G.2):

$$f(u) := \mathcal{N}(\Phi(\Theta_{\mathbf{1}}, \|\cdot\|_1, u)$$

we have by Theorem G.1 and lemmas G.1 and G.2

$$\begin{aligned}
\mathbb{E}_{\mathbf{Z}} \ell\text{-generalization error}(\hat{\theta}(\mathbf{Z}), \mathbf{Z}, \mu) &\leqslant \frac{24\sqrt{2}L}{n} \mathbb{E}_{\mathbf{X}} \left(\int_0^\infty \sqrt{\ln \mathcal{N}(\Theta, d_{\mathbf{X}}, t)} dt \right) \\
&\leqslant \frac{24\sqrt{2}L}{n} \mathbb{E}_{\mathbf{X}} \left(2r\sigma_{\mathbf{X}} \sqrt{\ln |\mathcal{S}|} + \sigma_{\mathbf{X}} \int_0^{2r} \sqrt{\ln(f(u))}\, du \right)
\end{aligned}$$

Because of the weight-sharing assumption, the number $|\mathcal{S}|$ of signs is at most equal to $2^{\#\text{params}}$. Moreover, Theorem H.1 guarantees for every $u > 0$

$$f(u) = \mathcal{N}(\Phi(\Theta_{\mathbf{1}}), \|\cdot\|_1, u) \leqslant 2^{\#\text{ rescalings}} \max\left(1, \left(\frac{24 \max(D, d_{\text{out}})r}{u}\right)^{\#\text{params} - \#\text{ rescalings}}\right). \quad (34)$$

We get

$$\begin{aligned}
\int_0^{2r} \sqrt{\ln(f(u))} du &= \int_0^{2r} \sqrt{\ln\left(2^{\#\text{ rescalings}} \left(\frac{24 \max(D, d_{\text{out}})r}{u}\right)^{\#\text{params} - \#\text{ rescalings}}\right)} du \\
&\leqslant 2r\sqrt{\ln(2)\#\text{ rescalings}} + \sqrt{\#\text{params} - \#\text{ rescalings}} \int_0^{2r} \sqrt{\ln\left(\frac{24 \max(D, d_{\text{out}})r}{u}\right)} du.
\end{aligned}$$

For the last integral, do a change of variable $t = u/24 \max(D, d_{\text{out}})r$ to get:

$$\begin{aligned}
\int_0^{2r} \sqrt{\ln\left(\frac{24 \max(D, d_{\text{out}})r}{u}\right)} du &= 24 \max(D, d_{\text{out}})r \int_0^{1/12 \max(2D, d_{\text{out}})} \sqrt{\ln(1/t)} dt \\
&\leqslant 24 \max(D, d_{\text{out}})r \underbrace{\int_0^{1/12} \sqrt{\ln(1/t)} dt}_{\leqslant 1/3} \\
&\leqslant 8 \max(D, d_{\text{out}})r.
\end{aligned}$$

Putting everything together, we get:

$$\begin{aligned}
&\mathbb{E}_{\mathbf{Z}} \text{ } \ell\text{-generalization error of } \hat{\theta}(\mathbf{Z}) \\
&\leqslant \frac{24\sqrt{2}L}{n} \mathbb{E}_{\mathbf{X}} \left(2r\sigma_{\mathbf{X}} \sqrt{\ln |\mathcal{S}|} + \sigma_{\mathbf{X}} \int_0^{2r} \sqrt{\ln(f(u))}\, du \right) \\
&\leqslant \frac{24\sqrt{2}L}{n} \mathbb{E}_{\mathbf{X}} \left(2r\sigma_{\mathbf{X}} \sqrt{\ln(2)\#\text{params}} + 2r\sigma_{\mathbf{X}} \sqrt{\ln(2)\#\text{ rescalings}} + 8r\sigma_{\mathbf{X}} \max(D, d_{\text{out}}) \sqrt{\#\text{params} - \#\text{ rescalings}} \right) \\
&\leqslant \frac{24\sqrt{2}L}{n} 2r\sigma_X \left(\sqrt{\ln(2)\#\text{params}} + \sqrt{\ln(2)\#\text{ rescalings}} + 4 \max(D, d_{\text{out}}) \sqrt{\#\text{params} - \#\text{ rescalings}} \right) \\
&\leqslant \frac{48\sqrt{2}L}{n} r\sigma_X \sqrt{\#\text{params}} \left(\underbrace{2\sqrt{\ln(2)}}_{\simeq 1.38 \leqslant 4 \max(D, d_{\text{out}})} + 4 \max(D, d_{\text{out}}) \right) \\
&\leqslant \frac{384\sqrt{2}L}{n} r\sigma_X \sqrt{\#\text{params}} \max(D, d_{\text{out}}).
\end{aligned}$$

Since $240\sqrt{2} \simeq 543$, this yields the bound $544 \frac{\sigma_X}{n} L \max(D, d_{\text{out}}) \sqrt{\#\text{params}} \times r$. Moreover, $\sigma_X = \mathbb{E}_{\mathbf{X}} \left(\sum_{i=1}^n \max(1, \|\mathbf{X}_i\|_\infty^2) \right)^{1/2} \leqslant \sqrt{n} B$. This yields the claim. □

H Covering numbers of $\Phi(\Theta)$

Theorem E.1 implies a bound on the covering numbers (Definition G.2) of $\Phi(\Theta)$.

Theorem H.1. *Consider a set of parameters Θ to be weight-sharing compatible (Definition G.4) with a directed regular partition $N_0, \dots, N_L$ (Definition G.3) of a DAG G (Definition A.2). Assume that for $V = N_{in}$ and $V = N_{out}$, each N_ℓ is either disjoint from V or is a subset of V. Assume also that $N_{in} \cap N_{out} = \emptyset$. Define L_0 to be the unique integer in $[\![0, L]\!]$ such that $N_{L_0} \subset N_{in}$ and $N_{L_0+1} \cap N_{in} = \emptyset$. For $\ell \in [\![L_0, L]\!]$, denote by k_ℓ the common number of antecedents of all neurons in N_ℓ and define # rescalings $:= L - L_0$ and #params $= \sum_{\ell=L_0}^{L}(k_\ell + 1)$. Recall that $D = \max_{p\in\mathcal{P}} \texttt{length}(p)$ is the depth of the graph and $d_{out} = |N_{out}|$ is the output dimension. Denote by $r := \sup_{\theta\in\Theta} \|\Phi(\theta)\|_1$. It holds:*

$$\mathcal{N}(\Phi(\Theta), \|\cdot\|_1, t) \leqslant 2^{\#\text{ rescalings}} \max\left(1, \frac{24\max(D, d_{out})r}{t}\right)^{\#\text{params}-\#\text{ rescalings}}$$

where the definition of covering numbers is recalled in Definition G.2.

Proof. For $\theta \in \Theta$, denote $\mathtt{N}(\theta)$ its "normalized version", obtained as the output of Algorithm 1 in Gonon et al. [2024] on input θ (see Algorithm 1). By Lemma B.1 in Gonon et al. [2024], $\Phi(\theta) = \Phi(\mathtt{N}(\theta))$ so for every $\theta, \theta' \in \Theta$:

$$\|\Phi(\theta) - \Phi(\theta')\|_1 = \|\Phi(\mathtt{N}(\theta)) - \Phi(\mathtt{N}(\theta'))\|_1.$$

For every neuron $u \in N \setminus N_{\text{in}}$ and all parameters θ, denote by $\theta(u) := (b_u, \theta^{\to u})$, and recall that $b_v = 0$ for $*$-max-pooling neurons v (Definition A.5). By Theorem E.1 with $q = 1$, we have:

$$\begin{aligned}
&\|\Phi(\mathtt{N}(\theta)) - \Phi(\mathtt{N}(\theta'))\|_1 \\
&\leqslant \sum_{v\in N_{\text{out}}\setminus N_{\text{in}}} |\mathtt{N}(b)_v - \mathtt{N}(b)'_v| + \|\mathtt{N}(\theta)^{\to v} - (\mathtt{N}(\theta'))^{\to v}\|_1 \\
&+ \underbrace{\min\left(\|\Phi(\mathtt{N}(\theta))\|_1, \|\Phi(\mathtt{N}(\theta'))\|_1\right)}_{\leqslant r} \max_{p\in\mathcal{P}:p_{\text{end}}\notin N_{\text{in}}} \sum_{\ell=1}^{\texttt{length}(p)-1} \left(|\mathtt{N}(b)_{p_\ell} - \mathtt{N}(b)'_{p_\ell}| + \|\mathtt{N}(\theta)^{\to p_\ell} - (\mathtt{N}(\theta'))^{\to p_\ell}\|_1\right) \\
&\leqslant \underbrace{\sum_{v\in N_{\text{out}}\setminus N_{\text{in}}} \|\mathtt{N}(\theta)(v) - \mathtt{N}(\theta')(v)\|_1}_{=:(1)} \\
&+ r \underbrace{\max_{p\in\mathcal{P}:p_{\text{end}}\notin N_{\text{in}}} \sum_{\ell=1}^{\texttt{length}(p)-1} \|\mathtt{N}(\theta)(p_\ell) - \mathtt{N}(\theta')(p_\ell)\|_1}_{=:(2)}.
\end{aligned}$$

Consider $L_0 \in [\![0, L]\!]$ such that $N_{L_0} \subset N_{\text{in}}$ and $N_{L_0+1} \cap N_{\text{in}} = \emptyset$. The integer L_0 is well defined since every N_ℓ is either disjoint from N_{in} or is a subset of N_{in}, and at least one of them must be disjoint since $N_0, \dots, N_L$ is a partition of the neurons and $N_{\text{in}} \cap N_{\text{out}} = \emptyset$.

For every $\ell \in [\![L_0, L]\!]$, consider an arbitrary $v_\ell \in N_\ell$. By Corollary G.1, we have $\mathtt{N}(\theta)(v) = \mathtt{N}(\theta)(v_\ell)$ for every $v \in N_\ell$, every $\ell \in [\![L_0, L]\!]$ and every parameters $\theta \in \Theta$. We get

$$\begin{aligned}
(1) &= \sum_{\ell=0}^{L} \sum_{v\in(N_{\text{out}}\cap N_\ell)\setminus N_{\text{in}}} \|\mathtt{N}(\theta)(v) - \mathtt{N}(\theta')(v)\|_1 \\
&= \sum_{\ell=L_0}^{L} |N_{\text{out}} \cap N_\ell| \|\mathtt{N}(\theta)(v_\ell) - \mathtt{N}(\theta')(v_\ell)\|_1.
\end{aligned}$$

Consider $p \in \mathcal{P}$ and $f : [\![0, \texttt{length}(p)]\!] \mapsto [\![0, L]\!]$ the function defined by $p_\ell \in N_{f(\ell)}$ for every $\ell \in [\![0, \texttt{length}(p)]\!]$. Once again using Corollary G.1, since $p_\ell \notin N_{\text{in}}$ for $\ell > 0$, we have $\mathtt{N}(\theta)(p_\ell) = \mathtt{N}(\theta)(v_{f(\ell)})$ for every $\ell \in [\![1, \texttt{length}(p)]\!]$ and every parameters $\theta \in \Theta$. This yields

$$(2) = \max_{p\in\mathcal{P}:p_{\text{end}}\notin N_{\text{in}}} \sum_{\ell=1}^{\texttt{length}(p)-1} \|\mathtt{N}(\theta)(v_{f(\ell)}) - \mathtt{N}(\theta')(v_{f(\ell)})\|_1.$$

Assume that for every $\ell \in [\![0, L]\!]$, it holds

$$\|\mathtt{N}(\theta)(v_{f(\ell)}) - \mathtt{N}(\theta')(v_{f(\ell)})\|_1 \leqslant \begin{cases} \frac{t}{2d_{\text{out}}} & \text{if } \ell = L, \\ \frac{t}{2Dr} & \text{otherwise.} \end{cases}$$

where we recall that $d_{\text{out}} = |N_{\text{out}}$ is the output dimension and $D = \max_{p \in \mathcal{P}} \mathtt{length}(p)$ is the depth of the graph. This implies:

$$(1) = \sum_{\ell=L_0}^{L} |N_{\text{out}} \cap N_\ell| \|\mathtt{N}(\theta)(v_\ell) - \mathtt{N}(\theta')(v_\ell)\|_1 \leqslant \frac{t}{2d_{\text{out}}} \sum_{\ell=L_0}^{L} |N_{\text{out}} \cap N_\ell| \leqslant \frac{t}{2d_{\text{out}}}.$$

Consider $p \in \mathcal{P}$. Since the partition $N_0, \dots, N_L$ is directed and $p_\ell \to p_{\ell+1}$ is an edge, we have $f(k) < f(\ell)$ for every $k < \ell$. In particular, $f(\ell) < f(\mathtt{length}(p)) \leqslant L$ for every $\ell \in [\![1, \mathtt{length}(p) - 1]\!]$, so we have

$$(2) = \max_{p \in \mathcal{P} : p_{\text{end}} \notin N_{\text{in}}} \sum_{\ell=1}^{\mathtt{length}(p)-1} \|\mathtt{N}(\theta)(v_{f(\ell)}) - \mathtt{N}(\theta')(v_{f(\ell)})\|_1 \leqslant \frac{t}{2Dr} \max_{p \in \mathcal{P} : p_{\text{end}} \notin N_{\text{in}}} (\mathtt{length}(p) - 1) \leqslant \frac{t}{2r}.$$

Therefore, we get:

$$\begin{aligned} \|\Phi(\mathtt{N}(\theta)) - \Phi(\mathtt{N}(\theta'))\|_1 &\leqslant (1) + r(2) \\ &\leqslant \frac{t}{2} + r\frac{t}{2r} = t. \end{aligned}$$

For a neuron $v \notin N_{\text{in}}$, denote $\mathtt{N}(\Theta)(v) := \{\mathtt{N}(\theta)(v), \theta \in \Theta\}$. In terms of covering numbers, we just proved that:

$$\mathcal{N}(\Phi(\theta), \|\cdot\|_1, t) \leqslant \prod_{\substack{\ell \in [\![L_0, L]\!] \\ N_\ell \subset N_{\text{out}}}} \mathcal{N}(\mathtt{N}(\Theta)(v_\ell), \|\cdot\|_1, t/2d_{\text{out}}) \prod_{\substack{\ell \in [\![L_0, L]\!] \\ N_\ell \cap N_{\text{out}} = \emptyset}} \mathcal{N}(\mathtt{N}(\Theta)(v_\ell), \|\cdot\|_1, t/2Dr).$$

We now bound the latter.

Consider $v \in N_{out} \setminus N_{in}$. By Lemma B.1 in [Gonon et al., 2024], it holds $\mathtt{N}(\theta)(v) \leqslant r$ for every $\theta \in \Theta$. In this situation, $\mathtt{N}(\Theta)(v)$ is a subset of the closed ℓ^1-ball $B_{k_v+1}(0, r)$, with $k_v := |\operatorname{ant}(v)|$ (k for kernel size) so

$$\mathcal{N}(\mathtt{N}(\Theta)(v), \|\cdot\|_1, t/2d_{\text{out}}) \leqslant \mathcal{N}(B_{k_v+1}(0, r), \|\cdot\|_1, t/4d_{\text{out}}).$$

It is well known that the covering with respect to $\|\cdot\|_1$ of the closed ball $B_d \subset \mathbb{R}^d$ with center 0 and radius R satisfies [Wainwright, 2019, Lemma 5.7]:

$$\mathcal{N}(B_d, \|\cdot\|_1, t) \leqslant \max\left(1, \frac{3R}{t}\right)^d.$$

Since the partition $N_0, \dots, N_L$ is regular, all neurons in N_ℓ have the same number of antecedents: denote it by k_ℓ. We get:

$$\prod_{\substack{\ell \in [\![L_0, L]\!] \\ N_\ell \subset N_{\text{out}}}} \mathcal{N}(\mathtt{N}(\Theta)(v_\ell), \|\cdot\|_1, t/2d_{\text{out}}) \leqslant \prod_{\substack{\ell \in [\![L_0, L]\!] \\ N_\ell \subset N_{\text{out}}}} \max\left(1, \frac{12d_{\text{out}}r}{t}\right)^{k_\ell}$$

Case $v \in N \setminus (N_{out} \cup N_{in})$. For every $\theta \in \Theta$, Lemma B.1 in Gonon et al. [2024] guarantees that $\|\mathtt{N}(()\theta)(v)\|_1 \in \{0, 1\}$ so $\mathtt{N}(\Theta)(v) \subset \{0\} \cup S^{k_v}$ with S^{k_v} the sphere of radius 1 in dimension $k_v + 1$ with respect to $\|\cdot\|_1$. We deduce that a t-covering of $\mathtt{N}(\Theta)(v)$ is given by the union of the null vector and a $t/2$-covering of the sphere S^{k_v}:

$$\mathcal{N}(\mathtt{N}(\Theta)(v), \|\cdot\|_1, t/2Dr) \leqslant 1 + \mathcal{N}(S^{k_v}, \|\cdot\|_1, t/4Dr).$$

The unit sphere S_d in dimension $d + 1$ satisfies

$$S_d = f(B_d) \cup g(B_d)$$

where $f(x_1, \dots, x_d) = (x_1, \dots, x_d, 1 - \|x\|_1)$ and $g(x_1, \dots, x_d) = (x_1, \dots, x_d, \|x\|_1 - 1)$. For every $x, \tilde{x} \in \mathbb{R}^d$:

$$\|f(x) - f(\tilde{x})\|_1 = \sum_{i \leqslant d} |x_i - \tilde{x_i}| + |(1 - \|x\|_1) - (1 - \|\tilde{x}\|_1)| \leqslant 2\|x - \tilde{x}\|_1.$$

Thus, the union of the images of a $\frac{t}{2}$-covering of B_d under both f and g is a t-covering of S_d:

$$\mathcal{N}(S_d, \|\cdot\|_1, t) \leqslant 2\mathcal{N}(B_d, \|\cdot\|_1, t/2) \leqslant 2 \max\left(1, \frac{6}{t}\right)^d.$$

We deduce that

$$\prod_{\substack{\ell \in [\![L_0, L]\!] \\ N_\ell \cap N_{\text{out}} = \emptyset}} \mathcal{N}(\mathtt{N}(\Theta)(v_\ell), \|\cdot\|_1, t/2Dr) \leqslant \prod_{\substack{\ell \in [\![L_0, L]\!] \\ N_\ell \cap N_{\text{out}} = \emptyset}} 2 \max\left(1, \frac{24Dr}{t}\right)^{k_\ell}.$$

We now return to our covering of $\Phi(\Theta)$ and deduce that:

$$\mathcal{N}(\Phi(\Theta), \|\cdot\|_1, t) \leqslant \prod_{\substack{\ell \in [\![L_0, L]\!] \\ N_\ell \subset N_{\text{out}}}} \max\left(1, \frac{12 d_{\text{out}} r}{t}\right)^{k_\ell} \prod_{\substack{\ell \in [\![L_0, L]\!] \\ N_\ell \cap N_{\text{out}} = \emptyset}} 2 \max\left(1, \frac{24Dr}{t}\right)^{k_\ell}.$$

Denote # rescalings $:= L - L_0$ and #params $:= \sum_{\ell=L_0}^{L} (k_\ell + 1)$. We get the desired result:

$$\mathcal{N}(\Phi(\Theta), \|\cdot\|_1, t) \leqslant 2^{\#\text{ rescalings}} \max\left(1, \frac{24 \max(D, d_{\text{out}}) r}{t}\right)^{\#\text{params} - \#\text{ rescalings}}. \qquad \square$$