A Appendix

A.1 DCP Pseudocode

Algorithm 1 Dynamic Convolutional Partition (DCP)

Input: Convolutional block $F : \mathbb{R}^{n_0 \times n_0 \times c_0} \to \mathbb{R}^{n_\ell \times n_\ell \times c_\ell}$. Empirically estimated $N_{\max}$.

Output: Flattened subnetworks $\mathcal{F} = \left\{ \widetilde{f}_{ij}^{k} \middle| i, j \in [d_k], k \in [s] \right\}$.

1: STOP $\leftarrow$ false, $s \leftarrow 0, \mathcal{C}_s \leftarrow \emptyset, D = \emptyset, R = \emptyset, \mathcal{F} \leftarrow \emptyset$;
2: **while** !STOP **do**
3: $d \leftarrow 2$;
4: Compute the set of d-part RIPs of n_ℓ, as $\mathcal{P}_d \leftarrow R_d^{n_\ell}$
5: **for** $r \in \mathcal{P}_d$ **do**
6: Compute constraint dimensions for subnetwork indexed by $(\ell - 2, \ell)$;
7: **if** Eq. (19) is violated **then**
8: $d \leftarrow d + 1$, restart from Line 4;
9: **else**
10: $s \leftarrow s + 1, d_s \leftarrow d, r_s \leftarrow r$;
11: **while** $\ell \geq 1$ **do**
12: $L \leftarrow \ell$
13: Compute constraint dimensions for subnetwork indexed by $(L - 1, \ell)$;
14: **if** Eq. (19) is not violated **then**
15: $L \leftarrow L - 1$;
16: **else**
17: $C_s \leftarrow C_s \cup \{L, \ell\}, D \leftarrow D \cup \{d_s\}, R \leftarrow R \cup \{r_s\}$
18: $s \leftarrow s + 1, \ell \leftarrow L$;
19: Restart from Line 3;
20: **end if**
21: **end while**
22: STOP $\leftarrow$ true;
23: **end if**
24: **end for**
25: **end while**
26: Return flattened subnetwork $\mathcal{F}$ using C_s, D and R.

A.2 Proof of Theorem 4.1.

Focusing on the flattened, partitioned CNN F_u as defined in Equation (11) and according to the definition of Lipschitz constant based on l_2-norm, for any two arbitrary inputs of F_u denoted by $\mathbf{x}^0$ and $\mathbf{y}^0$, we have

$$\left\| F_u\left(\mathbf{x}^0\right) - F_u\left(\mathbf{y}^0\right) \right\|_2^2 = \sum_{i,j=1}^{d} \left| \widetilde{f}_{ij}\left(\mathbf{x}_{ij}^0\right) - \widetilde{f}_{ij}\left(\mathbf{y}_{ij}^0\right) \right|^2 \leq \sum_{i,j=1}^{d} L\left(\widetilde{f}_{ij}\right) \left\| \mathbf{x}_{ij}^0 - \mathbf{y}_{ij}^0 \right\|_2^2. \tag{22}$$

Defining the two vectors:

$$\boldsymbol{\gamma} \coloneqq \left[L\left(\widetilde{f}_{11}\right)^2, L\left(\widetilde{f}_{12}\right)^2, \ldots, L\left(\widetilde{f}_{dd}\right)^2 \right]^\top,$$

$$\underline{\Delta} \coloneqq \left[\|\mathbf{x}_{11}^0 - \mathbf{y}_{11}^0\|_2^2, \ \|\mathbf{x}_{12}^0 - \mathbf{y}_{12}^0\|_2^2, \ldots, \|\mathbf{x}_{dd}^0 - \mathbf{y}_{dd}^0\|_2^2 \right]^\top,$$

and applying the Cauchy-Schwartz inequality, we obtain

$$\left\| F_u\left(\mathbf{x}^0\right) - F_u\left(\mathbf{y}^0\right) \right\|_2^2 \leq \langle \boldsymbol{\gamma}, \underline{\Delta} \rangle \leq \|\boldsymbol{\gamma}\|_2 \|\underline{\Delta}\|_2. \tag{23}$$

Given

$$\|\underline{\Delta}\|_2^2 = \sum_{i,j=1}^{d} \left\| \mathbf{x}_{ij}^0 - \mathbf{y}_{ij}^0 \right\|_2^4, \tag{24}$$

we observe the following

$$\left\|\mathbf{x}^0 - \mathbf{y}^0\right\|_2^4 = \left(\sum_{i,j=1}^{d} \left\|\mathbf{x}_{ij}^0 - \mathbf{y}_{ij}^0\right\|_2^2\right)^2 = \sum_{i,j=1}^{d} \left\|\mathbf{x}_{ij}^0 - \mathbf{y}_{ij}^0\right\|_2^4 + \zeta = \|\underline{\Delta}\|_2^2 + \zeta, \tag{25}$$

where ζ denotes all the additional non-negative terms from the expansion. This results in

$$\|\underline{\Delta}\|_2 \leq \left\|\mathbf{x}^0 - \mathbf{y}^0\right\|_2^2. \tag{26}$$

Combining (23) and (26), we have

$$\left\|F_u\left(\mathbf{x}^0\right) - F_u\left(\mathbf{y}^0\right)\right\|_2 \leq \sqrt{\|\boldsymbol{\gamma}\|_2}\left\|\mathbf{x}^0 - \mathbf{y}^0\right\|_2 = \left(\sum_{i,j=1}^{d} L\left(\widetilde{f_{ij}}\right)^4\right)^{\frac{1}{4}} \left\|\mathbf{x}^0 - \mathbf{y}^0\right\|_2. \tag{27}$$

We have shown that $L(F_u)$ is bounded above by $\left(\sum_{i,j=1}^{d} L\left(\widetilde{f_{ij}}\right)^4\right)^{\frac{1}{4}}$, which concludes the proof.

A.3 Proof of Corollary 4.2

An ℓ-layer convolutional block is a composite function of the $s$subnetworks, as detailed by (13) and (14). From Lemma 3.1, it follows that

$$L(F_u) \leq \prod_{k=1}^{s} L(F_k^u). \tag{28}$$

Then apply Theorem 4.1 to bound the Lipschitz constant of each subnetwork F_k^u by

$$L(F_k^u) \leq \left(\sum_{i,j=1}^{d_k} L\left(\widetilde{f}_{ij}^{(k)}\right)^4\right)^{\frac{1}{4}}. \tag{29}$$

Substituting (29) into (28) gives the required result:

$$L(F_u) \leq \prod_{k=1}^{s} L(F_k^u) \leq \prod_{k=1}^{s}\left(\sum_{i,j=1}^{d_k} L\left(\widetilde{f}_{ij}^{(k)}\right)^4\right)^{\frac{1}{4}}. \tag{30}$$

A.4 Proof of Proposition 4.3.

For an ℓ-layer convolutional block with a filter of dimension h and stride s applied at each convolutional layer, the formula for computing the output size is: $n_k = \lfloor \frac{n_{k-1}-h}{s} \rfloor + 1$, i.e. $n_{k-1} = s(n_k - \epsilon_k - 1) + h$, for $k \in \{1, 2, \ldots, \ell\}$ and $\epsilon_k \in \{0, 1\}$. The value of ϵ_k accounts for if $\lfloor \cdot \rfloor$ is a non-integer. From this the dimension of the k-th receptive field can be expressed in terms of n_ℓ as follows

$$n_k = s^{\ell-k} n_\ell + \sum_{i=1}^{\ell-k} (f s^{i-1} - s^i - \epsilon_i s^i), \text{ for } k \in \{0, 1, \ldots, \ell - 1\}. \tag{31}$$

Similarly, the k-th receptive field corresponding to the largest convolutional block is given by

$$\rho_k = s^{\ell-k} \rho_* + \sum_{i=1}^{\ell-k} (f s^{i-1} - s^i - \epsilon_i s^i), \text{ for } k \in \{0, 1, \ldots, \ell - 1\}, \tag{32}$$

where we recall that ρ_* denotes the dimension of of the largest convolutional block in the final layer, resulting from applying a d-part RIP to n_ℓ.

To obtain (20) and (21), we state a few facts. First the number of equality constraints in (3.2) is one, i.e. $m = 1$. Secondly, for deep networks the per-iteration time complexity is dominated by the cubic

term. Using the fact that $N = \sum_{i=0}^{\ell-1} c_i n_i^2$ and $N_* = \sum_{i=0}^{\ell-1} c_i \rho_i^2$ for the largest convolutional block, substitution of (31) gives

$$\begin{aligned} N =& c_{\ell-1}\left[sn_\ell + \sum_{i=1}^{1}(fs^{i-1} - s^i - \epsilon_i s^i)\right]^2 + c_{\ell-2}\left[s^2 n_\ell + \sum_{i=1}^{2}(fs^{i-1} - s^i - \epsilon_i s^i)\right]^2 + \\ & \qquad \ldots + c_0\left[s^\ell n_\ell + \sum_{i=1}^{\ell}(fs^{i-1} - s^i - \epsilon_i s^i)\right]^2 \\ =& n_\ell^2 \sum_{i=0}^{\ell-1} c_i s^{2(\ell-i)} + 2n_\ell\left[\sum_{j=1}^{\ell} c_{\ell-j} s^j \sum_{i=1}^{j}(fs^{i-1} - s^i - \epsilon_i s^i)\right] + \\ & \qquad \sum_{j=1}^{\ell} c_{\ell-j}\left(\sum_{i=1}^{j}(fs^{i-1} - s^i - \epsilon_i s^i)\right)^2. \end{aligned} \tag{33}$$

In an analogous way, we find N_* to be

$$\begin{aligned} N_* = \rho_*^2 \sum_{i=0}^{\ell-1} c_i s^{2(\ell-i)} + 2\rho_*\left[\sum_{j=1}^{\ell} c_{\ell-j} s^j \sum_{i=1}^{j}(fs^{i-1} - s^i - \epsilon_i s^i)\right] + \\ \sum_{j=1}^{\ell} c_{\ell-j}\left(\sum_{i=1}^{j}(fs^{i-1} - s^i - \epsilon_i s^i)\right)^2. \end{aligned} \tag{34}$$

To simplify the notation, let a_0, a_1, a_2 denote the first, second and third summation terms in (33) respectively- and likewise for (34). Then by considering the cubic binomial expansion of N_*/N, we obtain the following

$$\left(\frac{N_*}{N}\right)^3 = \frac{a_0^3\rho_*^6 + 8a_1^3\rho_*^3 + a_2^3 + \zeta_1}{a_0^3 n_\ell^6 + 8a_1^3 n_\ell^3 + a_2^3 + \zeta_2}, \tag{35}$$

where ζ_1 and ζ_2 denote lower order terms. Using the fact that there exist $b_0, b_1 > a_0$ such that $a_0^3 n_\ell^6 \leq N^3 \leq b_0^3 n_\ell^6$ and $a_0^3\rho_*^6 \leq N_*^3 \leq b_1^3\rho_*^6$, then (35) is bounded above as follows

$$\left(\frac{N_*}{N}\right)^3 \leq \left(\frac{b_1}{a_0}\right)^3 \left(\frac{\rho_*}{n_\ell}\right)^6. \tag{36}$$

To derive the results in (20) and (21) we use the fact that $\lceil \frac{n_\ell}{d} \rceil \leq \rho_* \leq n_\ell - (d-1)$ and consider the two following cases:

Case I. When $d = n_\ell$, ρ_* can be at most 1. Substituting this value into (36) gives:

$$\left(\frac{N_*}{N}\right)^3 \leq \left(\frac{b_1}{a_0}\right)^3 \frac{1}{n_\ell^6} = O\left(\frac{1}{n_\ell^6}\right), \text{ as } n_\ell \to \infty. \tag{37}$$

Case II. When $d = 2$, ρ_* can be at most $\lceil \frac{n_\ell}{2} \rceil$. Substituting this value into (36) gives:

$$\left(\frac{N_*}{N}\right)^3 \leq \left(\frac{b_1}{a_0}\right)^3 \frac{1}{2^6} = O(1). \tag{38}$$

A.5 Additional Information and Experiments

A.5.1 $N_{\max}$ Estimation

The estimation of $N_{\max}$ involves generating multiple SDPs of increasing size and recording the constraint size for which computational bottlenecks. Averaging over these instances will give an estimation for $N_{\max}$. While this approximation will be tight, it is computationally expensive as the problem size for which bottlenecks are reached, is not known a priori.

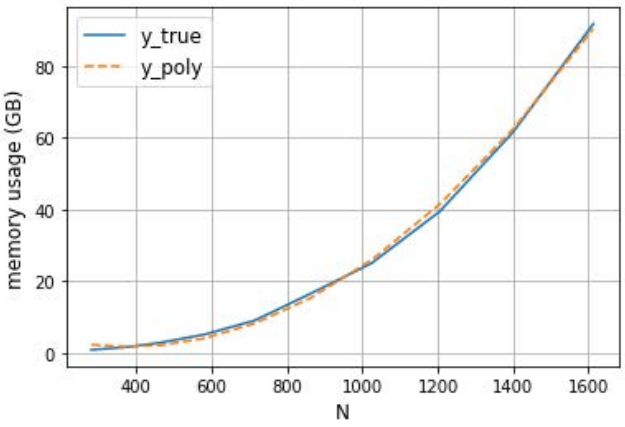

Figure 3: Memory usage for increasing problem dimensions.

As an alternative, we implement a prediction-based method, which may give a more conservative estimate of $N_{\max}$ but is computationally cheaper. It runs LipSDP on several small to medium sized problems, recording the memory usage. Using the collected data, $N_{\max}$ is estimated via extrapolation. While this still involves multiple calls of LipSDP, it avoids having to increment the problem size until computational bottlenecks, which isn't known a priori, and is thus is more efficient. In Figure 3, we show the recorded memory usage of the solver for increasing problem sizes (y_true) and compared against a quadratic fit of the data (y_poly). Using the approximating polynomial, we extrapolated the value of $N_{\max}$ based on a maximum memory capacity of 120GB of RAM, which gave a value of 1800. We chose a more conservative value than this in practice, i.e., $N_{\max} = 1400$, to account for memory associated with additional computations and running the operating system.

A.5.2 Provably 1-Lipschitz Network

To further assess the effectiveness of the DCP method, we apply it to provably 1-Lipschitz networks. As mentioned in (Anil et al., 2019), a provably 1-Lipschitz network is one which is composed of 1-Lipschitz affine transformations and activations. To this end, we consider a 2-layer convolutional block with an input dimension of 28×28, followed by two convolutional layers of sizes $12 \times 12 \times c$ and $4 \times 4 \times 1$, where the number of channels, c, varies from 1 to 3. We used the ReLU activation function that is 1-Lipschitz, and orthogonalised the weight matrices. The results are shown in Table 2. LipSDP gives estimated upper-bounds less than 1, which we conjecture is due to the approximation error underpinning the SDP formulation. DCP-LipSDP gives an upper-bound on the LipSDP values.

Table 2: Comparison of Lipschitz estimation for provably 1-Lipschitz, 2-layer convolutional blocks of size $28 \times 28 \to 12 \times 12 \times c \to 4 \times 4$.

Channel Number (c)	LipSDP	DCP-LipSDP
1	0.85	1.03
2	0.63	0.79
3	0.64	0.82