A. Supplementary Discussions for Sections 2 and 3

In this section we elaborate on some of the discussions in Sections 2 and 3 and give proofs of the various statements.

A.1. Generalization Bounds from Compression

We will first prove Theorem 2.1, which gives generalization guarantees for the compressed function.

Proof. (Theorem 2.1) For each $A \in \mathcal{A}$, the training loss $\hat{L}_0(g_A)$ is just the average of m i.i.d. Bernoulli random variables with expectation equal to $L_0(g_A)$. Therefore by Chernoff bound we have

$$\Pr[L_0(g_A) - \hat{L}_0(g_A) \geq \tau] \leq \exp(-2\tau^2 m).$$

Therefore, suppose we choose $\tau = \left(\sqrt{\frac{q \log r}{m}}\right)$, with probability at least $1 - \exp(-2q \log r)$ we have $L_0(g_A) \leq \hat{L}_0(g_A) + \tau$. There are only r^q different $A \in \mathcal{A}$, hence by union bound, with probability at least $1 - \exp(-q \log r)$, for all $A \in \mathcal{A}$ we have

$$L_0(g_A) \leq \hat{L}_0(g_A) + \left(\sqrt{\frac{q \log r}{m}}\right).$$

Next, since f is (γ, S)-compressible with respect to g, there exists $A \in \mathcal{A}$ such that for $x \in S$ and any y we have

$$|f(x)[y] - g_A(x)[y]| \leq \gamma.$$

For these training examples, as long as the original function f has margin at least γ, the new function g_A classifies the example correctly. Therefore

$$\hat{L}_0(g_A) \leq \hat{L}_\gamma(f).$$

Combining these two steps, we immediately get the result. □

Using the same approach, we can also prove the following Corollaries that allow the compression to fail with some probability

Corollary A.1. *In the setting of Theorem 2.1, if the compression works for $1 - \zeta$ fraction of the training sample, then with high probability*

$$L_0(g_A) \leq \hat{L}_\gamma(f) + \zeta + O\left(\sqrt{\frac{q \log r}{m}}\right).$$

Proof. The proof is using the same approach, except in this case we have

$$\hat{L}_0(g_A) \leq \hat{L}_\gamma(f) + \zeta.$$

□

A.2. Example 1: Compress a Vector

This section gives detailed calculations supporting the first example in Section 2.

Lemma 4. *Algorithm 2 Vector-Compress(γ, c) returns a vector $\hat{c}$ such that for any fixed u (independent of choice of $\hat{c}$), with probability at least $1 - \eta$, $|\hat{c}^\top u - c^\top u| \leq \gamma$. The vector $\hat{c}$ has at most $O((\log h)/\eta\gamma^2)$ non-zero entries with high probability.*

Proof. By the construction in Algorithm 2, it is easy to check that for all i, $\mathbb{E}[\hat{c}_i] = c_i$. Also, $\text{Var}[\hat{c}] = 2p_i(1-p_i)\frac{c_i^2}{p_i^2} \leq \frac{2c_i^2}{p_i} \leq \eta\gamma^2$.

Therefore, for any vector u that is independent with the choice of $\hat{c}$, we have $\mathbb{E}[\hat{c}^\top u] = c^\top u$ and $\text{Var}[\hat{c}^\top u] \leq \|u\|^2/4 \leq \eta\gamma^2$. Therefore by Chebyshev's inequality we know $\Pr[|\hat{c}^\top u - c^\top u| \geq \gamma] \leq \eta$.

Algorithm 2 Vector-Compress(γ, c)

Require: vector c with $\|c\| \le 1$, η.
Ensure: vector $\hat{c}$ s.t. for any fixed vector $\|u\| \le 1$, with probability at least $1-\eta$, $|c^\top u - \hat{c}^\top u| \le \gamma$. Vector $\hat{c}$ has $O((\log h)/\eta\gamma^2)$ nonzero entries.
for $i = 1$ to d **do**
  Let $z_i = 1$ with probability $p_i = \frac{2c_i^2}{\eta\gamma^2}$ (and 0 otherwise)
  Let $\hat{c}_i = z_i c_i / p_i$.
end for
Return $\hat{c}$

On the other hand, the expected number of non-zero entries in $\hat{c}$ is $\sum_{i=1}^d p_i = 2/\eta\gamma^2$. By Chernoff bound we know with high probability the number of non-zero entries is at most $O((\log h)/\eta\gamma^2)$. □

Next we handle the discretization:

Lemma 5. *Let $\tilde{c}$ = Vector-Compress$(\gamma/2, c)$. For each coordinate i, let $\hat{c}_i = 0$ if $|\tilde{c}_i| \ge 2\eta\gamma\sqrt{h}$, otherwise let $\hat{c}_i$ be the rounding of $\tilde{c}_i$ to the nearest multiple of $\gamma/2\sqrt{h}$. For any fixed u with probability at least $1-\eta$, $|\hat{c}^\top u - c^\top u| \le \gamma$.*

Proof. Let c' be a truncated version of c: $c'_i = c_i$ if $|c_i| \ge \gamma/4\sqrt{h}$, and $c'_i = 0$ otherwise. It is easy to check that $\|c' - c\| \le \gamma/4$. By Algorithm 2, we observe that $\tilde{c}$ = Vector-Compress$(\gamma/2, c')$ ($|\tilde{c}_i| \ge 2\eta\gamma\sqrt{h}$ if and only if $|c_i| \le \gamma/4\sqrt{h}$). Finally, by the rounding we know $\|\hat{c} - \tilde{c}\| \le \gamma/4$. Combining these three terms, we know with probability at least $1-\eta$,

$$\begin{aligned}|\hat{c}^\top u - c^\top u| &\le |\hat{c}^\top u - \tilde{c}^\top u| + |\tilde{c}^\top u - (c')^\top u| + |(c')^\top u - c^\top u| \\ &\le \gamma/4 + \gamma/2 + \gamma/4 = \gamma.\end{aligned}$$

□

Combining the above two lemmas, we know there is a compression algorithm with $O((\log h)/\eta\gamma^2)$ discrete parameters that works with probability at least $1-\eta$. Applying Corollary A.1 we get

Lemma 6. *For any number of sample m, there is an efficient algorithm to generate a compressed vector $\hat{c}$, such that*

$$L(\hat{c}) \le \tilde{O}((1/\gamma^2 m)^{1/3}).$$

Proof. We will choose $\eta = (1/\gamma^2 m)^{1/3}$. By Lemma 4 and Lemma 5, we know there is a compression algorithm that works with probability $1-\eta$, and has at most $\tilde{O}((\log h)/\eta\gamma^2)$ parameters. By Corollary A.1, we know

$$L(\hat{c}) \le \tilde{O}(\eta + \sqrt{1/\eta\gamma^2 m}) \le \tilde{O}((1/\gamma^2 m)^{1/3}).$$

□

Note that the rate we have here is not optimal as it depends on $m^{1/3}$ instead of $\sqrt{m}$. This is mostly due to Lemma 4 cannot give a high probability bound (indeed if we consider all the basis vectors as the test vectors u, Vector-Compress is always going to fail on some of them).

Compression with helper string To fix this problem we use a different algorithm that uses a helper string, see Algorithm 3

Note that in Algorithm 3, the parameters for the output are the z_i's. The vectors v_i's are sampled independently, and hence can be considered to be in the helper string.

Lemma 7. *For any fixed vector u, Algorithm 3 Vector-Project(c, γ) produces a vector $\hat{c}$ such that with probability at least $1-\eta$, we have $|\hat{c}^\top u - c^\top u| \le \gamma$.*

Proof. This is in fact a well-known corollary of Johnson-Lindenstrauss Lemma. Observe that

$$\hat{c}^\top u = \frac{1}{k}\sum_{i=1}^k \langle v_i, c\rangle\langle v_i, u\rangle.$$

Algorithm 3 Vector-Project(γ, c)

Require: vector c with $\|c\| \leq 1$, η.
Ensure: vector $\hat{c}$ s.t. for any fixed vector $\|u\| \leq 1$, with probability at least $1-\eta$, $|c^\top u - \hat{c}^\top u| \leq \gamma$.
Let $k = 16\log(1/\eta)/\gamma^2$
Sample k random Gaussian vectors $v_1, ..., v_k \sim \mathcal{N}(0, I)$.
Compute $z_i = \langle v_i, c\rangle$
(Optional): Round z_i to the closes multiple of $\gamma/2\sqrt{hk}$.
Return $\hat{c} = \frac{1}{k}\sum_{i=1}^k z_i v_i$

The expectation $\mathbb{E}[\langle v_i, c\rangle\langle v_i, u\rangle] = \mathbb{E}[c^\top v_i v_i^\top u] = c^\top \mathbb{E}[v_i v_i^\top]u = c^\top u$. The variance is bounded by $O(1/k) \leq O(\gamma/\sqrt{\log n})$. Standard concentration bounds show that

$$\Pr[|\hat{c}^\top u - c^\top u| > \gamma/2] \leq \exp(-\gamma^2 k/16) \leq \eta.$$

The discretization is easy to check as with high probability the matrix V with columns v_i's have spectral norm at most $2\sqrt{h}$, so the vector before and after discretization can only change by $\gamma/2$. □

Lemma 8. *For any number of sample m, there is an efficient algorithm with helper string to generate a compressed vector $\hat{c}$, such that*

$$L(\hat{c}) \leq \tilde{O}(\sqrt{1/\gamma^2 m}).$$

Proof. We will choose $\eta = 1/m$. By Lemma 7, we know there is a compression algorithm that works with probability $1-\eta$, and has at most $O((\log 1/\eta)/\gamma^2)$ parameters. By Corollary A.1, we know

$$L(\hat{c}) \leq \tilde{O}(\eta + \sqrt{1/\gamma^2 m}) \leq \tilde{O}(\sqrt{1/\gamma^2 m}).$$

□

A.3. Proof for Generalization Bound in (Neyshabur et al., 2017a)

We gave a compression in Lemma 1, the discretization in this case is trivial just by rounding the weights to nearest multiples of $\|A\|_F/h^2$. The following lemma from (Neyshabur et al., 2017a) (based on a simple induction of the noise) shows how the noises from different layers add up.

Lemma 9. *Let f_A be a d-layer network with weights $A = \{A^1, \ldots, A^d\}$. Then for any input x, weights A and $\hat{A}$, if for any layer i, $\|A^i - \hat{A}^i\| \leq \frac{1}{d}\|A^i\|$, then we have:*

$$\|f_A(x) - f_{\hat{A}}(x)\| \leq e\|x\| \left(\prod_{i=1}^d \|A^i\|_2\right) \sum_{i=1}^d \frac{\|A^i - \hat{A}^i\|_2}{\|A^i\|_2}$$

Compressing each layer i with $\delta = \delta = \gamma(e\|x\|d\prod_{i=1}^d \|A^i\|_2)^{-1}$ ensures $|f_A(x) - f_{\hat{A}}(x)| \leq \gamma$. Since each $\hat{A}^i$ has rank $\frac{\|A^i\|_F^2}{\delta^2\|A^i\|_2^2}$, the total number of paremeters of the compressed network will be $2e^2d^2h\|x\|^2\prod_{i=1}^d\|A^i\|_2^2\sum_{i=1}^d\frac{\|A^i\|_F^2}{\|A^i\|_2^2}$. Therefore we can apply Theorem 2.1 to get the generalization bound.

A.4. Further Discussion on Interlayer Smoothness

In order to understand the above condition, we can look at a single layer case where $j = i+1$:

$$\begin{aligned}
&\|M^{i,i+1}(x^i + \eta) - J_{x^i}^{i,i+1}(x^i + \eta)\| \\
&= \|A^{i+1}\phi(x^i + \eta) - A^{i+1}(\phi'(x^i) \odot (x^i + \eta))\| \\
&= \|A^{i+1}\nu\| \leq \frac{\|\eta\|\|A^{i+1}\phi(x^i)\|}{\rho_\delta\|x^i\|}
\end{aligned}$$

where $\odot$ is the entrywise product operator and $\nu = (\phi'(x^i+\eta) - \phi'(x^i)) \odot (x^i+\eta)$. Since the activation function is ReLU, $\phi'(x^i+\eta)$ and $\phi'(x^i))$ disagree whenever the perturbation has the opposite sign and higher absolute value compare to the input and hence $\|\nu\| \leq \|\eta\|$. Let us first see what happens if the perturbation ν is adversarially aligned to the weights:

$$\begin{aligned}
&\|M^{i,i+1}(x^i+\eta) - J_{x^i}^{i,i+1}(x^i+\eta)\| \\
&= \|A^{i+1}\nu\| \leq \|A^{i+1}\|\|\eta\| \\
&= \frac{\|\eta\|\|A^{i+1}\phi(x^i)\|}{\|x^i\|} \cdot \frac{\|A^{i+1}\|\|x^i\|}{\|A^{i+1}\phi(x^i)\|} \\
&\leq \frac{\|\eta\|\|A^{i+1}\phi(x^i)\|}{\|x^i\|} \cdot \frac{\|A^{i+1}\|\|x^i\|}{\mu_{i+1}\|A^{i+1}\|_F\|\phi(x^i)\|} \\
&\leq \frac{\|\eta\|\|A^{i+1}\phi(x^i)\|}{\|x^i\|} \cdot \frac{c\|A^{i+1}\|}{\mu_{i+1}\|A^{i+1}\|_F} \\
&= \frac{\|\eta\|\|A^{i+1}\phi(x^i)\|}{\|x^i\|} \cdot \frac{c}{\mu_{i+1}r_{i+1}}
\end{aligned}$$

where r_{i+1} is the stable rank of layer $i+1$. Therefore the interlayer smoothness from layer i to layer $i+1$ is at least $\rho_\delta = \mu_{i+1}r_{i+1}/c$. However, the noise generated from Algorithm 1 has similar properties to Gaussian noise (see Lemma 2). If ν behaves similar to Gaussian noise, then $\|A^{i+1}\nu\| \approx \|A^{i+1}\|_F\|\nu\|/\sqrt{h^i}$ and therefore ρ_δ is as high as $\sqrt{h^i\mu_{i+1}/c}$. Since the layer cushion of networks trained on real data is much more than that of networks with random weights, ρ_δ is greater than one in this case. Another observation is that in practice, the noise is well-distributed and only a small portion of activations change from active to inactive and vice versa. Therefore, we can expect $\|\nu\|$ to be smaller than $\|\eta\|$ which further improves the interlayer smoothness. This appeared in Neyshabur et al. (2017b) that showed for one layer we can even use $\frac{\|\eta\|^{1.5}\|x^j\|}{\rho_\delta\|x^i\|}$ as the RHS of interlayer smoothness. Our current proof requires $1/\rho_\delta$ to be of order $1/d$, this requirement can be removed (with ρ_δ appear in sample complexity) if we make the stronger assumption that the RHS is a lower order term in $\|\eta\|$.

B. Complete Proofs for Section 4

B.1. Conditions

We discussed and verified several conditions in Section 3. Here, we formally state these conditions:

Condition B.1. Let S be the training set.

1. **Layer cushion (μ_i)**: For any layer i, we define the layer cushion μ_i as the largest number such that for any $x \in S$:
$$\mu_i\|A^i\|_F\|\phi(x^{i-1})\| \leq \|A^i\phi(x^{i-1})\|$$

2. **Interlayer cushion ($\mu_{i,j}$)**: For any two layers $i \leq j$, we define interlayer cushion $\mu_{i,j}$ as the largest number such that for any $x \in S$:
$$\mu_{i,j}\|J_{x^i}^{i,j}\|_F\|x^i\| \leq \|J_{x^i}^{i,j}x^i\|$$
Furthermore, we define minimal interlayer cushion $\mu_{i\rightarrow} = \min_{i\leq j\leq d}\mu_{i,j} = \min\{1/\sqrt{h^i}, \min_{i<j\leq d}\mu_{i,j}\}$.

3. **Activation contraction (c)**: The activation contaction c is defined as the smallest number such that for any layer i and any $x \in S$,
$$\|x^i\| \leq c\|\phi(x^i)\|$$

4. **Interlayer smoothness (ρ_δ)**: Interlayer smoothness is defined the largest number such that with probability $1-\delta$ over noise η for any two layers $i < j$ any $x \in S$:
$$\|M^{i,j}(x^i+\eta) - J_{x^i}^{i,j}(x^i+\eta)\| \leq \frac{\|\eta\|\|x^j\|}{\rho_\delta\|x^i\|}$$

B.2. Proofs

Proof. (of Lemma 2) For any fixed vectors u, v, we have

$$u^\top \hat{A} v = \frac{1}{k}\sum_{k'=1}^{k} u^\top Z_{k'} v = \frac{1}{k}\langle A, M_{k'}\rangle\langle uv^\top, M_{k'}\rangle.$$

This is exactly the same as the case of Johnson-Lindenstrauss transformation. By standard concentration inequalities we know

$$\Pr\left[\left|\frac{1}{k}\sum_{k'=1}^{k}\langle A, M_{k'}\rangle\langle uv^\top, M_{k'}\rangle - \langle A, uv^\top\rangle\right| \geq \epsilon\|A\|_F\|uv^\top\|_F\right] \leq \exp(-k\epsilon^2).$$

Therefore for the choice of k we know

$$\Pr\left[|u^\top \hat{A} v - u^\top A v\| \geq \varepsilon\|A\|_F\|u\|\|v\|\right] \leq \eta.$$

Now for any pair of matrix/vector $(U, x) \in G$, let u_i be the i-th row of U, by union bound we know with probability at least $1-\delta$ for all u_i we have $|u_i^\top \Delta v\| \leq \varepsilon\|A\|_F\|u_i\|\|v\|$. Since $\|U\Delta x\|^2 = \sum_{i=1}^n (u_i^\top \Delta x)^2$ and $\|U\|_F^2 = \sum_{i=1}^n \|u_i\|^2$, we immediately get $\|U\Delta x\| \geq \varepsilon\|A\|_F\|U\|_F\|x\|$. □

Proof. (of Lemma 3) We will prove this by induction. For any layer $i \geq 0$, let $\hat{x}_i^j$ be the output at layer j if the weights $A^1, \ldots, A^i$ in the first i layers are replaced with $\tilde{A}^1, \ldots, \tilde{A}^i$. The induction hypothesis is then the following:

Consider any layer $i \geq 0$ and any $0 < \varepsilon \leq 1$. The following is true with probability $1 - \frac{i\delta}{2d}$ over $\tilde{A}^1, \ldots, \tilde{A}^i$ for any $j \geq i$:

$$\|\hat{x}_i^j - x^j\| \leq (i/d)\varepsilon\|x^j\|.$$

For the base case $i = 0$, since we are not perturbing the input, the inequality is trivial. Now assuming that the induction hypothesis is true for $i-1$, we consider what happens at layer i. Let $\hat{A}^i$ be the result of Algorithm 1 on A^i with $\varepsilon_i = \frac{\varepsilon\mu_i\mu_{i\to}}{4cd}$ and $\eta = \frac{\delta}{6d^2h^2m}$. We can now apply Lemma 2 on the set $G = \{(J_{x^i}^{i,j}, x^i) | x \in S, j \geq i\}$ which has size at most dm. Let $\Delta^i = \hat{A}^i - A^i$, for any $j \geq i$ we have

$$\|\hat{x}_i^j - x^j\| = \|(\hat{x}_i^j - \hat{x}_{i-1}^j) + (\hat{x}_{i-1}^j - x^j)\| \leq \|(\hat{x}_i^j - \hat{x}_{i-1}^j)\| + \|\hat{x}_{i-1}^j - x^j\|.$$

The second term can be bounded by $(i-1)\varepsilon\|x^j\|/d$ by induction hypothesis. Therefore, in order to prove the induction, it is enough to show that the first term is bounded by ε/d. We decompose the error into two error terms one of which corresponds to the error propagation through the network if activation were fixed and the other one is the error caused by change in the activations:

$$\begin{aligned}
\|(\hat{x}_i^j - \hat{x}_{i-1}^j)\| &= \|M^{i,j}(\hat{A}^i\phi(\hat{x}^{i-1})) - M^{i,j}(A^i\phi(\hat{x}^{i-1}))\| \\
&= \|M^{i,j}(\hat{A}^i\phi(\hat{x}^{i-1})) - M^{i,j}(A^i\phi(\hat{x}^{i-1})) + J_{x^i}^{i,j}(\Delta^i\phi(\hat{x}^{i-1})) - J_{x^i}^{i,j}(\Delta^i\phi(\hat{x}^{i-1}))\| \\
&\leq \|J_{x^i}^{i,j}(\Delta^i\phi(\hat{x}^{i-1}))\| + \|M^{i,j}(\hat{A}^i\phi(\hat{x}^{i-1})) - M^{i,j}(A^i\phi(\hat{x}^{i-1})) - J_{x^i}^{i,j}(\Delta^i\phi(\hat{x}^{i-1}))\|
\end{aligned}$$

The first term can be bounded as follows:

$$\begin{aligned}
&\|J_{x^i}^{i,j}\Delta^i\phi(\hat{x}^{i-1})\| \\
&\leq (\varepsilon\mu_i\mu_{i\to}/6cd)\|J_{x^i}^{i,j}\|\|A^i\|_F\|\phi(\hat{x}^{i-1})\| && \text{Lemma 2} \\
&\leq (\varepsilon\mu_i\mu_{i\to}/6cd)\|J_{x^i}^{i,j}\|\|A^i\|_F\|\hat{x}^{i-1}\| && \text{Lipschitzness of the activation function} \\
&\leq (\varepsilon\mu_i\mu_{i\to}/3cd)\|J_{x^i}^{i,j}\|\|A^i\|_F\|x^{i-1}\| && \text{Induction hypothesis} \\
&\leq (\varepsilon\mu_i\mu_{i\to}/3d)\|J_{x^i}^{i,j}\|\|A^i\|\|\phi(x^{i-1})\| && \text{Activation Contraction} \\
&\leq (\varepsilon\mu_{i\to}/3d)\|J_{x^i}^{i,j}\|\|A^i\phi(x^{i-1})\| && \text{Layer Cushion} \\
&= (\varepsilon\mu_{i\to}/3d)\|J_{x^i}^{i,j}\|\|x^i\| && x^i = A^i\phi(x^{i-1}) \\
&\leq (\varepsilon/3d)\|x^j\| && \text{Interlayer Cushion}
\end{aligned}$$

The second term can be bounded as:

$$\begin{aligned}&\|M^{i,j}(\hat{A}^i\phi(\hat{x}^{i-1})) - M^{i,j}(A^i\phi(\hat{x}^{i-1})) - J^{i,j}_{x^i}(\Delta^i\phi(\hat{x}^{i-1}))\|\\&= \|(M^{i,j} - J^{i,j}_{x^i})(\hat{A}^i\phi(\hat{x}^{i-1})) - (M^{i,j} - J^{i,j}_{x^i})(A^i\phi(\hat{x}^{i-1}))\|\\&= \|(M^{i,j} - J^{i,j}_{x^i})(\hat{A}^i\phi(\hat{x}^{i-1}))\| + \|(M^{i,j} - J^{i,j}_{x^i})(A^i\phi(\hat{x}^{i-1})\|.\end{aligned}$$

Both terms can be bounded using interlayer smoothness condition of the network. First, notice that $A^i\phi(\hat{x}^{i-1}) = \hat{x}^i_{i-1}$. Therefore by induction hypothesis $\|A^i\phi(\hat{x}^{i-1}) - x^i\| \leq (a-1)\varepsilon\|x^i\|/d \leq \varepsilon\|x^i\|$. Now by interlayer smoothness property, $\|(M^{i,j} - J^{i,j}_{x^i})(A^i\phi(\hat{x}^{i-1})\| \leq \frac{\|x^b\|\varepsilon}{\rho_\delta} \leq (\varepsilon/3d)\|x^j\|$. On the other hand, we also know $\hat{A}^i\phi(\hat{x}^{i-1}) = \hat{x}^i_{i-1} + \Delta^i\phi(\hat{x}^{i-1})$, therefore $\|\hat{A}^i\phi(\hat{x}^{i-1}) - x^i\| \leq \|A^i\phi(\hat{x}^{i-1}) - x^i\| + \|\Delta^i\phi(\hat{x}^{i-1})\| \leq (i-1)\varepsilon/d + \varepsilon/3d \leq \varepsilon$, so again we have $\|(M^{i,j} - J^{i,j}_{x^i})(\hat{A}^i\phi(\hat{x}^{i-1}))\| \leq (\varepsilon/3d)\|x^j\|$. Putting everything together completes the induction. □

Lemma 10. *For any fully connected network f_A with $\rho_\delta \geq 3d$, any probability $0 < \delta \leq 1$ and any margin $\gamma > 0$, f_A can be compressed (with respect to a random string) to another fully connected network $f_{\tilde{A}}$ such that for any $x \in S$, $\hat{L}_0(f_{\hat{A}}) \leq \hat{L}_\gamma(f_A)$ and the number of parameters in $f_{\tilde{A}}$ is at most:*

$$\tilde{O}\left(\frac{c^2d^2\max_{x\in S}\|f_A(x)\|_2^2}{\gamma^2}\sum_{i=1}^{d}\frac{1}{\mu_i^2\mu_{i\to}^2}\right)$$

where μ_i, $\mu_{i\to}$, c and ρ_δ are layer cushion, interlayer cushion, activation contraction and interlayer smoothness defined in Definitions 4,5,6 and 7 respectively.

Proof. (of Lemma 10) If $\gamma^2 > 2\max_{x\in S}\|f_A(x)\|_2^2$, for any pair (x, y) in the training set we have $|f_A(x)[y] - \max_{i\neq y} f_A(x)[j]|^2 \leq 2\max_{x\in S}\|f_A(x)\|_2^2 \leq \gamma$ which means the output margin cannot be greater than γ and therefore $\hat{L}_\gamma(f_A) = 1$ which proves the statement. If $\gamma^2 \leq 2\max_{x\in S}\|f_A(x)\|_2^2$, by setting $\varepsilon^2 = \gamma^2/2\max_{x\in S}\|f_A(x)\|_2^2$ in Lemma 3, we know that for any $x \in S$, $\|f_A(x) - f_{\tilde{A}}(x)\|_2 \leq \gamma/\sqrt{2}$. For any (x, y), if the margin loss on the right hand side is one then the inequality holds. Otherwise, the output margin in $f_{\tilde{A}}$ is greater than γ which means in order for classification loss of f_A to be one, we neet to have $\|f_A(x) - f_{\tilde{A}}(x)\|_2 > \gamma/\sqrt{2}$ which is not possible and that completes the proof. □

Proof. (of Theorem 4.1) We show the generalization by bounding the covering number of the network with weights $\tilde{A}$. We already demonstrated that the original network with weights A can be approximated with another network with weights $\tilde{A}$ and less number of parameters. In order to get a covering number, we need to find out the required accuracy for each parameter in the second network to cover the original network. We start by bounding the norm of the weights $\tilde{A}^i$.

Because of positive homogeneity of ReLU activations, we can assume without loss of generality that the network is balanced, i.e for any $i \neq j$, $\|A_i\|_F = \|A_j\|_F = \beta$ (otherwise, one could rebalance the network before approximation and cushion in invariant to this rebalancing). Therefore, for any $x \in S$ we have:

$$\beta^d = \prod_{i=1}^{d}\|A^i\| \leq \frac{c\|x^1\|}{\|x\|\mu_1}\prod_{i=2}^{d}\|A^i\| \leq \frac{c^2\|x^2\|}{\|x\|\mu_1\mu_2}\prod_{i=2}^{d}\|A^i\| \leq \frac{c^d\|f_A(x)\|}{\|x\|\prod_{i=1}^{d}\mu_i}$$

By Lemma 3, $\|\tilde{A}^i\|_F \leq \beta(1 + 1/d)$. We know that $\tilde{A}^i = \frac{1}{k}\sum_{k'=1}^{k}\langle A^i, M_{k'}\rangle M_{k'}$ where $\langle A^i, M_{k'}\rangle$ are the parameters. Therefore, if $\hat{A}^i$ correspond to the weights after approximating each parameter in $\tilde{A}^i$ with accuracy ν, we have: $\|\hat{A}^i - \tilde{A}^i\|_F \leq \sqrt{k}h\nu \leq \sqrt{q}h\nu$ where q is the total number of parameters. Now by Lemma 9, we get:

$$\begin{aligned}|\ell_\gamma(f_{\tilde{A}}(x), y) - \ell_\gamma(f_{\hat{A}}(x), y)| &\leq \frac{2e}{\gamma}\|x\|\left(\prod_{i=1}^{d}\|\tilde{A}^i\|\right)\sum_{i=1}^{d}\frac{\|\tilde{A}^i - \hat{A}^i\|}{\|\tilde{A}^i\|} < \frac{e^2}{\gamma}\|x\|\beta^{d-1}\sum_{i=1}^{d}\|\tilde{A}^i - \hat{A}^i\|_F\\&\leq \frac{e^2c^d\|f_A(x)\|\sum_{i=1}^{d}\|\tilde{A}^i - \hat{A}^i\|_F}{\gamma\beta\prod_{i=1}^{d}\mu_i} \leq \frac{qh\nu}{\beta}\end{aligned}$$

where the last inequality is because by Lemma 10, $\frac{e^2 d\|f_A(x)\|}{\gamma\beta\prod_{i=1}^d \mu_i} < \sqrt{q}$. Since the absolute value of each parameter in layer i is at most βh, the logarithm of number of choices for each parameter in order to get ε-cover is $\log(qh^2/\varepsilon) \le 2\log(qh/\varepsilon)$ which results in the covering number $2q\log(kh/\varepsilon)$. Bounding the Rademacher complexity by Dudley entropy integral completes the proof. □

C. Convolutional Neural Networks

In this section we give a compression algorithm for convolutional neural networks, and prove Theorem 5.1.

We start by developing some notations to work with convolutions and product of tensors. For simplicity of notation, for any $k' \le k$, we define a product operator $\times_{k'}$ that given a kth-order tensor Y and a k' order tensor Z with a matching dimensionality to the last k'-dimensions of Y, vectorizes the last k' dimensions of each tensor and returns a $k-k'$th order tensor as follows:

$$(Y \times_{k'} Z)_{i_1,\ldots,i_{k-k'}} = \langle Y_{i_1,\ldots,i_{k-k'}}, Z\rangle = \langle \text{vec}(Y_{i_1,\ldots,i_{k-k'}}), \text{vec}(Z)\rangle$$

Let $X \in \mathbb{R}^{h\times n_1\times n_2}$ be an $n\times n$ image where h is the number of features for each pixel. We denote the $\kappa\times\kappa$ sub-image of X starting from pixel (i,j) by $X_{(i,j),\kappa} \in \mathbb{R}^{h\times\kappa\times\kappa}$. Let $A \in \mathbb{R}^{h'\times h\times\kappa\times\kappa}$ be a convolutional weight tensor. Now the convolution operator with stride s can be defined as follows:

$$(A *_s X)_{i,j} = A \times_3 X_{(s(i-1)+1,s(j-1)+1),\kappa} \qquad \forall 1 \le i \le \lfloor\frac{n_1-\kappa}{s}\rfloor, 1 \le i \le \lfloor\frac{n_2-\kappa}{s}\rfloor$$

where $n_1' = \lfloor\frac{n_1-\kappa}{s}\rfloor$, $n_2' = \lfloor\frac{n_2-\kappa}{s}\rfloor$ and $A *_s X \in \mathbb{R}^{h'\times n_1'\times n_2'}$.

As we discussed in Section 5, we will actually have a different set of weights at each convolution location. Let $\hat{A}_{(i,j)} \in \mathbb{R}^{h'\times h\times\kappa\times\kappa}(i\in[n_1'], j\in[n_2'])$ be a set of weights for each location, we use the notation $\hat{A} *_s X$ to denote

$$((\hat{A} *_s X)_{i,j}) = \hat{A}_{(i,j)} \times_3 X_{(s(i-1)+1,s(j-1)+1),\kappa} \qquad \forall 1 \le i \le \lfloor\frac{n_1-\kappa}{s}\rfloor, 1 \le i \le \lfloor\frac{n_2-\kappa}{s}\rfloor.$$

The $\hat{A}_{(i,j)}$'s will be generated by Algorithm 4 and are p-wise independent.

Let κ_i be the filter size and s_i be the stride in layer i of the convolutional network. Then for any $i>1$, $x^{i+1} = \phi(A^i *_{s_i} x^i)$. Furthermore, since the activation functions are ReLU, we have $x^j = M^{ij}(x^i) = J^{ij}_{x^i} \times_3 x^i$.

In the rest of this section, we will first describe the compression algorithm Matrix-Project-Conv (Algorithm 4) and show that the output of this algorithm behaves similar to Gaussian noise (similar to Lemma 2). Then we will follow the same strategy as the feed-forward case and give the full proof.

C.1. p-wise Independent Compression

Algorithm 4 Matrix-Project-Conv(A, ε, η, $n_1'\times n_2'$)

Require: Convolution Tensor $A \in \mathbb{R}^{h'\times h\times\kappa\times\kappa}$, error parameter ε, η.
Ensure: Generate $n_1'\times n_2'$ different tensors $\hat{A}_{(i,j)}((i,j)\in[n_1']\times[n_2'])$ that satisfies Lemma 13
  Let $k = \frac{Q\lceil\kappa/s\rceil^2\log^2 1/\eta}{\epsilon^2}$ for a large enough universal constant Q.
  Let $p = \log(1/\eta)$
  Sample a uniformly random subspace $\mathcal{S}$ of $h'\times h\times\kappa\times\kappa$ of dimension $k\times p$
  for each $(i,j)\in[n_1']\times[n_2']$ **do**
    Sample k matrices $M_1, M_2, ..., M_k \in \mathcal{N}(0,1)^{h'\times h\times\kappa\times\kappa}$ with random i.i.d. entries.
    for $k'=1$ to k **do**
      Let $M'_{k'} = \sqrt{hh'\kappa^2/kp}\cdot \text{Proj}_{\mathcal{S}}(M_{k'})$.
      Let $Z_{k'} = \langle A, M'_{k'}\rangle M'_{k'}$.
    end for
    Let $\hat{A}_{(i,j)} = \frac{1}{k}\sum_{k'=1}^k Z_{k'}$
  end for

The weights in convolutional neural networks have inherent correlation due to the architecture, as the weights are shared across different locations. However, in order to randomly compress the weight tensors, we need to break this correlation and try to introduce independent perturbations at every location. The procedure is described as Algorithm 4.

The goal of Algorithm 4 is to generate different compressed filters $\hat{A}_{i,j}$ such that the total number of parameters is small, and at the same time $\hat{A}_{i,j}$'s behave very similarly to applying Algorithm 1 A for each location independently. We formalize these two properties in the following two lemmas:

Lemma 11. *Given a helper string that contains all of the M' matrices used in Algorithm 4, then it is possible to compute all of $\hat{A}_{(i,j)}$'s based on $\text{Proj}_{\mathcal{S}}(A)$. Since $\mathcal{S}$ is a kp dimensional subspace $\text{Proj}_{\mathcal{S}}(A)$ has kp parameters.*

Proof. By Algorithm 4 we know $\hat{A}_{(i,j)}$'s are average of the Z matrices, and $Z_{k'} = \langle A, M'_{k'}\rangle M'_{k'}$. Since $M'_{k'} \in \mathcal{S}$, we know $\langle A, M'_{k'}\rangle = \langle \text{Proj}_{\mathcal{S}}(A), M'_{k'}\rangle$. Hence $Z_{k'} = \langle \text{Proj}_{\mathcal{S}}(A), M'_{k'}\rangle M'_{k'}$ only depends on $\text{Proj}_{\mathcal{S}}(A)$ and $M'_{k'}$. □

Lemma 12. *The random matrices $\hat{A}_{(i,j)}$'s generated by Algorithm 4 are p-wise independent. Moreover, for any $\hat{A}_{(i,j)}$, the marginal distribution of the M' matrices are i.i.d. Gaussian with variance 1 in every direction.*

Proof. Take any subset of p random matrices $\hat{A}_{(i_1,j_1)}, ..., \hat{A}_{(i_p,j_p)}$ generated by Algorithm 4. We are going to consider the joint distribution of all the M' matrices used in generating these $\hat{A}$'s ($k \times p$ of them) and the subspace $\mathcal{S}$.

Consider the following procedure: generate $k \times p$ random matrices $M'_1, M'_2..., M'_{kp}$ from $N(0,1)^{h'\times h\times\kappa\times\kappa}$, and let $\mathcal{S}$ be the span of these kp vectors. By symmetry of Gaussian vectors, we know $\mathcal{S}$ is a uniform random subspace of dimension kp.

Now we sample from the same distribution in a different order: first sample a uniform random subspace $\mathcal{S}$ of dimension kp, then sample kp random Gaussian matrices within this subspace (which can be done by sample a Gaussian in the entire space and then project to this subspace). This is exactly the procedure described in Algorithm 4.

Therefore, the M' matrices used in generating these $\hat{A}$'s are independent, as a result the $\hat{A}_{(i,j)}$'s are also independent. The equivalence also shows that the marginal distributions of M' are i.i.d. spherical Gaussians. (Note that the reason this is limited to p-wise independence is that if we look at more than kp random matrices from the subspace $\mathcal{S}$, they do not have the same distribution as Gaussian random matrices; the latter would span a subspace of dimension higher than kp.) □

Although the $\hat{A}_{(i,j)}$'s are only p-wise independent, when $p = \log 1/\eta$ we can show that they behave similarly to fully independent random filters. We defer the technical concentration bounds to the end of this section (Section C.3).

Using this compression, we will prove that the noise generated at each layer behaves similar to a random vector. In particular it does not correlate with any fixed tensor, as long as the norms of the tensor is *well-distributed*:

Definition 10. Let $U \in \mathbb{R}^{h'\times n'_1\times n'_2\times n_u}$, we say U is β well-distributed if for any $i, j \in [n'_1] \times [n'_2]$, $\|U_{:,j,k,:}\|_F \le \frac{\beta}{\sqrt{n'_1 n'_2}}\|U\|_F$.

Intuitively, U is well-distributed if no spacial location of U has a norm that is significantly larger than the average. Now we are ready to show the noise generated by this procedure behaves very similar to a random Gaussian (this is a generalization of Lemma 2):

Lemma 13. *For any $0 < \delta, \varepsilon \le 1$, et $G = \{(U^i, V^i)\}_{i=1}^m$ be a set of matrix/vector pairs of size m where $U \in \mathbb{R}^{h'\times n'_1\times n'_2\times n_u}$[3] and $V \in \mathbb{R}^{h\times n_1\times n_2}$, let $\hat{A}_{(i,j)} \in \mathbb{R}^{h\times h'}$ be the output of Algorithm 4 with $\eta = \delta/n$ and $\Delta_{(i,j)} = \hat{A}_{(i,j)} - A$. Suppose all of U's are β-well-distributed. With probability at least $1-\delta$ we have for any $(U,V) \in G$, $\|U \times_3 (\Delta *_s V)\| \le \frac{\varepsilon\beta}{\sqrt{n'_1 n'_2}}\|A\|_F\|U\|_F\|V\|_F$.*

Proof. We will first expand out $U \times_3 (\Delta *_s V)$:

$$U \times_3 (\Delta *_s V) = \sum_{i=1}^{n'_1}\sum_{j=1}^{n'_2}(U_{:,i,j,:} \otimes V_{(s(i-1)+1,s(j-1)+1),\kappa}) \times_4 (\hat{A}_{(i,j)} - A).$$

[3] U can have more than 4-orders, here we vectorize all the remaining directions in U as it does not change the proof.

In this expression, $(U_{:,i,j,:} \otimes V_{(s(i-1)+1,s(j-1)+1),\kappa})$ generates a 5-th order tensor (2 from U and 3 from V), the order of dimensions is that V takes coordinates number 3,4,5 (with dimensions $h \times \kappa \times \kappa$), the first dimension of U takes the 2nd coordinate and the 4-th dimension of U takes the 1st coordinate. The result of $(U_{:,i,j,:} \otimes V_{(s(i-1)+1,s(j-1)+1),\kappa}) \times_4 (\hat{A}_{(i,j)} - A)$ is a vector of dimension n_u (because the first 4 dimensions are removed in the inner-product).

Now let us look at the terms in this sum, let $X_{i,j} = (U_{:,i,j,:} \otimes V_{(s(i-1)+1,s(j-1)+1),\kappa}) \times_4 \hat{A}_{(i,j)}$. Let $M'_1, ..., M'_k$ be the random matrices used when computing $\hat{A}_{(i,j)}$ (for simplicity we omit the indices for i, j), then we have

$$X_{i,j} = \frac{1}{k} \sum_{l=1}^{k} [(U_{:,i,j,:} \otimes V_{(s(i-1)+1,s(j-1)+1),\kappa}) \times_4 M'_l] \langle A, M'_l \rangle.$$

Since the marginal distribution of M'_l is a spherical Gaussian, it's easy to check that $\mathbb{E}[X_{i,j}] = (U_{:,i,j,:} \otimes V_{(s(i-1)+1,s(j-1)+1),\kappa}) \times_4 A$. Also, the first term $[(U_{:,i,j,:} \otimes V_{(s(i-1)+1,s(j-1)+1),\kappa}) \times_4 M'_l]$ is a Gaussian random vector whose expected squared norm is $\|U_{:,i,j,:}\|_F^2 \|V_{(s(i-1)+1,s(j-1)+1),\kappa}\|_F^2$; the second term $\langle A, M'_l \rangle$ is a Gaussian random variable with variance $\|A\|_F^2$. By the relationship between Gaussians and subexponential random variables, there exists a universal constant Q' such that $[(U_{:,i,j,:} \otimes V_{(s(i-1)+1,s(j-1)+1),\kappa}) \times_4 M'_l] \langle A, M'_l \rangle$ is a vector whose norm is $Q' \|U_{:,i,j,:}\|_F \|V_{(s(i-1)+1,s(j-1)+1),\kappa}\|_F \|A\|_F$-subexponential. The average of k independent copies lead to a random vector $X_{i,j}$ whose norm is $\sigma_{i,j}$-subexponential, where $\sigma_{i,j} = \frac{Q'}{\sqrt{k}} \|U_{:,i,j,:}\|_F \|V_{(s(i-1)+1,s(j-1)+1),\kappa}\|_F \|A\|_F$[4].

By Lemma 12 we know $X_{i,j}$'s are p-wise independent. Now we can apply Corollary C.2 to the sum of $X_{i,j}$'s. Let $\sigma = \sqrt{\sum_{i=1}^{n'_1} \sum_{j=1}^{n'_2} \sigma_{i,j}^2}$, then we know

$$\Pr[\|U \times_3 (\Delta *_s V)\| \geq 12\sigma p] \leq 2^{-p} = \eta = \delta/m.$$

Union bound over all (U, V) pairs, we know with probability at least $1 - \delta$, we have $\|U \times_3 (\Delta *_s V)\| \leq 12\sigma p$ for all (U, V).

Finally, we will try to relate $12\sigma p$ with $\frac{\varepsilon\beta}{\sqrt{n'_1 n'_2}} \|A\|_F \|U\|_F \|V\|_F$.

$$\begin{aligned}
\sigma &= \sqrt{\sum_{i=1}^{n'_1} \sum_{j=1}^{n'_2} \sigma_{i,j}^2} \\
&= \sqrt{\sum_{i=1}^{n'_1} \sum_{j=1}^{n'_2} \frac{(Q')^2}{k} \|U_{:,i,j,:}\|_F^2 \|V_{(s(i-1)+1,s(j-1)+1),\kappa}\|_F^2 \|A\|_F^2} \\
&= \frac{Q'}{\sqrt{k}} \|A\|_F \sqrt{\sum_{i=1}^{n'_1} \sum_{j=1}^{n'_2} \|U_{:,i,j,:}\|_F^2 \|V_{(s(i-1)+1,s(j-1)+1),\kappa}\|_F^2} \\
&\leq \frac{Q'\beta}{\sqrt{n'_1 n'_2 k}} \|A\|_F \|U\|_F \sqrt{\sum_{i=1}^{n'_1} \sum_{j=1}^{n'_2} \|V_{(s(i-1)+1,s(j-1)+1),\kappa}\|_F^2} \\
&\leq \frac{Q'\beta\lceil\kappa/s\rceil}{\sqrt{n'_1 n'_2 k}} \|A\|_F \|U\|_F \|V\|_F.
\end{aligned}$$

Here the first inequality is by the assumption that all U's are β-well-distributed. The second inequality is true because each entry in V appears in at most $\lceil\kappa/s\rceil^2$ entries of $V_{(s(i-1)+1,s(j-1)+1),\kappa}$. Therefore, when k is set to $144(Q')^2\lceil\kappa/s\rceil^2 p^2/\epsilon^2 = O(\frac{\lceil\kappa/s\rceil^2 \log^2 1/\eta}{\epsilon^2})$, we have $12\sigma p \leq \frac{\varepsilon\beta}{\sqrt{n'_1 n'_2}} \|A\|_F \|U\|_F \|V\|_F$ as desired. □

[4]Notice that here this average over k independent copies actually has a better tail than a subexponential random variable. However for simplicity we are not trying to optimize the dependencies on log factors here.

C.2. Generalization Bounds for Convolutional Neural Networks

Next we will use Algorithm 4 to compress the neural network and prove generalization bounds. Similar to the feed-forward case, our first step is to show bound the perturbation of the output based on the noise introduced at each layer. This is captured by the following lemma (generalization of Lemma 3)

Lemma 14. *For any convolutional neural network f_A with $\rho_\delta \geq 3d$, any probability $0 < \delta \leq 1$ and any error $0 < \varepsilon \leq 1$, Algorithm 4 generates weights $\tilde{A}^i_{(a,b)}$ for each layer i and each convolution location (a,b) with $\tilde{O}\left(\frac{c^2d^2\beta^2}{\varepsilon^2} \cdot \sum_{i=1}^{d} \frac{\lceil \kappa_i/s_i \rceil^2}{\mu_i^2 \mu_{i\to}^2}\right)$ total parameters such that with proability $1-\delta/2$ over the generated weights $\tilde{A}_{(i,j)}$, for any $x \in S$:*

$$\|f_A(x) - f_{\tilde{A}}(x)\| \leq \varepsilon \|f_A(x)\|.$$

where μ_i, $\mu_{i\to}$, c, ρ_δ and β are layer cushion, interlayer cushion, activation contraction, interlayer smoothness and well-distributedness of Jacobian defined in Definitions 4,8,6, 7 and 9 respectively.

Proof. We will prove this by induction. For any layer $i \geq 0$, let $\hat{x}^j_i$ be the output at layer j if the weights $A^1, \ldots, A^i$ in the first i layers are replaced with $\{\tilde{A}^1_{(a,b)}\}, \ldots, \{\tilde{A}^i_{(a,b)}\}$. The induction hypothesis is then the following:

Consider any layer $i \geq 0$ and any $0 < \varepsilon \leq 1$. The following is true with probability $1 - \frac{i\delta}{2d}$ over $\tilde{A}^1, \ldots, \tilde{A}^i$ for any $j \geq i$:

$$\|\hat{x}^j_i - x^j\| \leq (i/d)\varepsilon \|x^j\|.$$

(Note that although x is now a 3-tensor, we still use $\|x\|$ to denote $\|x\|_F$ as we never use any other norm of x.)

For the base case $i = 0$, since we are not perturbing the input, the inequality is trivial. Now assuming that the induction hypothesis is true for $i-1$, we consider what happens at layer i. Let $\tilde{A}^i$ be the result of Algorithm 1 on A^i with $\varepsilon_i = \frac{\varepsilon \mu_i \mu_{i\to}}{4cd\beta}$ and $\eta = \frac{\delta}{6d^2h^2m}$. We can now apply Lemma 2 on the set $G = \{(J^{i,j}_{x^i}, x^i) | x \in S, j \geq i\}$ which has size at most dm. Let $\Delta^i_{(a,b)} = \tilde{A}^i_{(a,b)} - A^i$ $((a,b) \in [n^i_1] \times [n^i_2])$, for any $j \geq i$ we have

$$\|\hat{x}^j_i - x^j\| = \|(\hat{x}^j_i - \hat{x}^j_{i-1}) + (\hat{x}^j_{i-1} - x^j)\| \leq \|(\hat{x}^j_i - \hat{x}^j_{i-1})\| + \|\hat{x}^j_{i-1} - x^j\|.$$

The second term can be bounded by $(i-1)\varepsilon\|x^j\|/d$ by induction hypothesis. Therefore, in order to prove the induction, it is enough to show that the first term is bounded by ε/d. We decompose the error into two error terms one of which corresponds to the error propagation through the network if activation were fixed and the other one is the error caused by change in the activations:

$$\begin{aligned}
\|(\hat{x}^j_i - \hat{x}^j_{i-1})\| &= \|M^{i,j}(\tilde{A}^i *_s \phi(\hat{x}^{i-1})) - M^{i,j}(A^i *_s \phi(\hat{x}^{i-1}))\| \\
&= \|M^{i,j}(\tilde{A}^i *_s \phi(\hat{x}^{i-1})) - M^{i,j}(A^i *_s \phi(\hat{x}^{i-1})) + J^{i,j}_{x^i} \times_3 (\Delta^i *_s \phi(\hat{x}^{i-1})) - J^{i,j}_{x^i} \times_3 (\Delta^i *_s \phi(\hat{x}^{i-1}))\| \\
&\leq \|J^{i,j}_{x^i} \times_3 (\Delta^i *_s \phi(\hat{x}^{i-1}))\| + \|M^{i,j}(\tilde{A}^i *_s \phi(\hat{x}^{i-1})) - M^{i,j}(A^i *_s \phi(\hat{x}^{i-1})) - J^{i,j}_{x^i} \times_3 (\Delta^i *_s \phi(\hat{x}^{i-1}))\|
\end{aligned}$$

The first term can be bounded as follows:

$$
\begin{aligned}
&\|J_{x^i}^{i,j} \times_3 (\Delta^i *_s \phi(\hat{x}^{i-1}))\| \\
&\leq (\varepsilon\mu_i\mu_{i\to}/6cd) \cdot \frac{1}{\sqrt{n_1^i n_2^i}} \|J_{x^i}^{i,j}\|_F \|A^i\|_F \|\phi(\hat{x}^{i-1})\| && \text{Lemma 13} \\
&\leq (\varepsilon\mu_i\mu_{i\to}/6cd) \cdot \frac{1}{\sqrt{n_1^i n_2^i}} \|J_{x^i}^{i,j}\|_F \|A^i\|_F \|\hat{x}^{i-1}\| && \text{Lipschitzness of the activation function} \\
&\leq (\varepsilon\mu_i\mu_{i\to}/3cd) \cdot \frac{1}{\sqrt{n_1^i n_2^i}} \|J_{x^i}^{i,j}\|_F \|A^i\|_F \|x^{i-1}\| && \text{Induction hypothesis} \\
&\leq (\varepsilon\mu_i\mu_{i\to}/3d) \cdot \frac{1}{\sqrt{n_1^i n_2^i}} \|J_{x^i}^{i,j}\|_F \|A^i\| \|\phi(x^{i-1})\| && \text{Activation Contraction} \\
&\leq (\varepsilon\mu_{i\to}/3d) \cdot \frac{1}{\sqrt{n_1^i n_2^i}} \|J_{x^i}^{i,j}\|_F \|A^i *_s \phi(x^{i-1})\| && \text{Layer Cushion} \\
&= (\varepsilon\mu_{i\to}/3d) \cdot \frac{1}{\sqrt{n_1^i n_2^i}} \|J_{x^i}^{i,j}\|_F \|x^i\| && x^i = A^i *_s \phi(x^{i-1}) \\
&\leq (\varepsilon/3d)\|x^j\| && \text{Interlayer Cushion}
\end{aligned}
$$

The second term can be bounded as:

$$
\begin{aligned}
&\|M^{i,j}(\tilde{A}^i *_s \phi(\hat{x}^{i-1})) - M^{i,j}(A^i *_s \phi(\hat{x}^{i-1})) - J_{x^i}^{i,j} \times_3 (\Delta^i *_s \phi(\hat{x}^{i-1}))\| \\
&= \|(M^{i,j} - J_{x^i}^{i,j}) \times_3 (\tilde{A}^i *_s \phi(\hat{x}^{i-1})) - (M^{i,j} - J_{x^i}^{i,j}) \times_3 (A^i *_s \phi(\hat{x}^{i-1}))\| \\
&= \|(M^{i,j} - J_{x^i}^{i,j}) \times_3 (\tilde{A}^i *_s \phi(\hat{x}^{i-1}))\| + \|(M^{i,j} - J_{x^i}^{i,j}) \times_3 (A^i *_s \phi(\hat{x}^{i-1})\|.
\end{aligned}
$$

Both terms can be bounded using interlayer smoothness condition of the network. First, notice that $A^i *_s \phi(\hat{x}^{i-1}) = \hat{x}_{i-1}^i$. Therefore by induction hypothesis $\|A^i *_s \phi(\hat{x}^{i-1}) - x^i\| \leq (i-1)\varepsilon\|x^i\|/d \leq \varepsilon\|x^i\|$. Now by interlayer smoothness property, $\|(M^{i,j} - J_{x^i}^{i,j}) \times_3 (A^i *_s \phi(\hat{x}^{i-1}))\| \leq \frac{\|x^j\|\varepsilon}{\rho_\delta} \leq (\varepsilon/3d)\|x^j\|$. On the other hand, we also know $\tilde{A}^i *_s \phi(\hat{x}^{i-1}) = \hat{x}_{i-1}^i + \Delta^i *_s \phi(\hat{x}^{i-1})$, therefore $\|\tilde{A}^i *_s \phi(\hat{x}^{i-1}) - x^i\| \leq \|A^i *_s \phi(\hat{x}^{i-1}) - x^i\| + \|\Delta^i *_s \phi(\hat{x}^{i-1})\| \leq (i-1)\varepsilon/d + \varepsilon/3d \leq \varepsilon$, so again we have $\|(M^{i,j} - J_{x^i}^{i,j}) \times_3 (\tilde{A}^i *_s \phi(\hat{x}^{i-1}))\| \leq (\varepsilon/3d)\|x^j\|$. Putting everything together completes the induction. □

Now we are ready to prove Theorem 5.1

Proof. We show the generalization by bounding the covering number of the network with weights $\tilde{A}$. We already demonstrated that the original network with weights A can be approximated with another network with weights $\tilde{A}$ and less number of parameters. In order to get a covering number, we need to find out the required accuracy for each parameter in the second network to cover the original network. We start by bounding the norm of the weights $\tilde{A}^i$.

Because of positive homogeneity of ReLU activations, we can assume without loss of generality that the network is balanced, i.e for any $i \neq j$, $\|A_i\|_F = \|A_j\|_F = \tau$ (otherwise, one could rebalance the network before approximation and cushion in invariant to this rebalancing). Therefore, for any $x \in S$ we have:

$$
\tau^d = \prod_{i=1}^d \|A^i\|_F \leq \frac{c\|x^1\|}{\|x\|\mu_1} \prod_{i=2}^d \|A^i\|_F \leq \frac{c^2\|x^2\|}{\|x\|\mu_1\mu_2} \prod_{i=2}^d \|A^i\|_F \leq \frac{c^d\|f_A(x)\|}{\|x\| \prod_{i=1}^d \mu_i}
$$

By Lemma 14 and Lemma 11, we know $\text{Proj}_S A^i$ are the parameter. Therefore, if $\hat{A}^i$ correspond to the weights after approximating each parameter in $\tilde{A}^i$ with accuracy ν, we have: $\|\hat{A}^i - \tilde{A}^i\|_F \leq \sqrt{k}h\nu \leq \sqrt{q}h\nu$ where q is the total number of parameters. Now by Lemma 9, we get:

$$|\ell_\gamma(f_{\tilde{A}}(x),y)-\ell_\gamma(f_{\hat{A}}(x),y)| \le \frac{2e}{\gamma}\|x\|\left(\prod_{i=1}^{d}\|\tilde{A}^i\|\right)\sum_{i=1}^{d}\frac{\|\tilde{A}^i-\hat{A}^i\|}{\|\tilde{A}^i\|} < \frac{e^2}{\gamma}\|x\|\tau^{d-1}\sum_{i=1}^{d}\|\tilde{A}^i-\hat{A}^i\|_F$$
$$\le \frac{e^2c^d\|f_A(x)\|\sum_{i=1}^{d}\|\tilde{A}^i-\hat{A}^i\|_F}{\gamma\tau\prod_{i=1}^{d}\mu_i} \le \frac{qh\nu}{\tau}$$

where the last inequality is because by Lemma 10, $\frac{e^2d\|f_A(x)\|}{\gamma\tau\prod_{i=1}^{d}\mu_i} < \sqrt{q}$. Since the absolute value of each parameter in layer i is at most τh, the logarithm of number of choices for each parameter in order to get ε-cover is $\log(qh^2/\varepsilon) \le 2\log(qh/\varepsilon)$ which results in the covering number $2q\log(kh/\varepsilon)$. Bounding the Rademacher complexity by Dudley entropy integral completes the proof. □

Similar to the discussions at the end of Section 4, we can use distance to initialization and remove outliers. More concretely, we can get the following corollary

Corollary C.1. *For any convolutional neural network f_A with $\rho_\delta \ge 3d$,any probability $0<\delta\le 1$ and any margin γ, Algorithm 4 generates weights $\tilde{A}$ for the network $f_{\tilde{A}}$ such that with probability $1-\delta$ over the training set and $f_{\tilde{A}}$:*

$$L_0(f_{\tilde{A}}) \le \hat{L}_\gamma(f_A)+\zeta+\tilde{O}\left(\sqrt{\frac{c^2d^2\max_{x\in S}\|f_A(x)\|_2^2\sum_{i=1}^{d}\frac{\beta^2(\lceil\kappa_i/s_i\rceil)^2}{\mu_i^2\mu_{i\to}^2}}{\gamma^2 m}}\right)$$

where μ_i, $\mu_{i\to}$, c and ρ_δ are layer cushion, interlayer cushion, activation contraction and interlayer smoothness defined in Definitions 4,8,6 and 7 respectively and measured on a $1-\zeta$ fraction of the training set S.

C.3. Concentration Inequalities for Sum of p-wise Independent Variables

In this section we prove a technical lemma that shows the sum of p-wise independent subexponential random variables have strong concentration properties. Previously similar results were known for Bernoulli random variables (Pelekis and Ramon, 2015), the approach we take here is very similar.

Definition 11. A random variable X is σ-subexponential if for all $k>0$, $\mathbb{E}[|X-\mathbb{E}[X]|^k] \le \sigma^k k^k$.

The following lemma will imply concentration

Lemma 15. *Let $X_1, X_2, ..., X_n$ be random variables where X_i is σ_i-subexponential. Let $\sigma^2=\sum_{i=1}^{n}\sigma_i^2$, $X=\sum_{i=1}^{n}X_i$. If X_i's are p-wise independent*

$$\mathbb{E}[(X-\mathbb{E}[X])^p] \le (3\sigma)^p\cdot(2p)^p.$$

In particular, for all $t>1$,

$$\Pr[|X-\mathbb{E}[X]| \ge 6\sigma pt] \le 1/t^p.$$

Proof. Let $Y_i=X_i-\mathbb{E}[X_i]$ and $Y=X-\mathbb{E}[X]$, we will compute $\mathbb{E}[Y^p]$.

$$\mathbb{E}[Y^p]=\sum_{a,a_i\in\mathcal{N},\sum a_i=p}\frac{p!}{\prod_{i=1}^{n}a_i!}\mathbb{E}[\prod_{i=1}^{n}Y_i^{a_i}]=\sum_{a,a_i\in\mathbb{N},\sum a_i=p}\frac{p!}{\prod_{i=1}^{n}a_i!}\prod_{i=1}^{n}\mathbb{E}[Y_i^{a_i}]$$

Here the last step is because Y_i's are p-wise independent. Now, notice that $\mathbb{E}[Y_i]=0$. Therefore, as long as one of the a_i's is equal to 1, we have $\prod_{i=1}^{n}\mathbb{E}[Y_i^{a_i}]=0$. All the remaining terms are terms with a_i's either equal to 0 or at least 2. Let $\mathcal{A}$ be the set of such a's, then we have

$$\mathbb{E}[Y^p]=\sum_{a\in\mathcal{A}}\frac{p!}{\prod_{i=1}^{n}a_i!}\prod_{i=1}^{n}\mathbb{E}[Y_i^{a_i}] \le (2p)^p\sum_{a\in\mathcal{A}}\prod_{i=1}^{n}\sigma_i^{a_i}.$$

By Claim 1 below, we know this expectation is bounded by $p^p(3\sigma)^p$. The second part of the lemma follows immediately from Markov's inequality. □

Claim 1. *Let $\mathcal{A}_{n,p}$ be the set of vectors $a \in \mathbb{N}^n$ where $a_i = 0$ or $a_i \geq 2$, $\sum_{i=1}^n a_i = p$. For any $n, p \geq 0$ and for any $\sigma_1, ..., \sigma_n > 0$, we have*

$$\sum_{a \in \mathcal{A}_{n,p}} \prod_{i=1}^n \sigma_i^{a_i} \leq (9 \sum_{i=1}^n \sigma_i^2)^{p/2}.$$

Proof. We do induction on n. When $n \leq 1$ this is clearly correct. Let $F(n,p) = \sum_{a \in \mathcal{A}_{n,p}} \prod_{i=1}^n \sigma_i^{a_i}$, then we have

$$F(n,p) = F(n-1,p) + \sum_{a=2}^{p} F(n-1, p-a)\sigma_n^a.$$

Suppose the claim is true for all $n < z$, let $\sigma' = \sqrt{\sum_{i=1}^{z-1} \sigma_i^2}$, when $n = z$ we have

$$\begin{aligned} F(z,p) &= F(z-1,p) + \sum_{a=2}^{p} F(n-1, p-a)\sigma_n^a \\ &\leq (3\sigma')^p + \sum_{a=2}^{p} (3\sigma')^{p-a}\sigma_n^a. \end{aligned}$$

When $\sigma_n \leq 2\sigma'$, we know $\sum_{a=2}^{p} (3\sigma')^{p-a}\sigma_n^a \leq 3(3\sigma')^{p-2}\sigma_n^2$, hence by Binomial expansion we have

$$(9(\sigma')^2 + 9\sigma_n^2)^{p/2} \geq (3\sigma')^p + (3\sigma')^{p-2} \cdot 9\sigma_n^2 \geq F(z,p).$$

On the other hand, if $\sigma_n \geq 2\sigma'$, then we know all the terms in the summation $\sum_{a=2}^{p} (3\sigma')^{p-a}\sigma_n^a$ and $(3\sigma')^p$ are bounded by $(1.5\sigma_n)^p$, therefore

$$(9(\sigma')^2 + 9\sigma_n^2)^{p/2} \geq (3\sigma_n)^p \geq (p-1)(2\sigma_n)^p \geq F(z,p).$$

In both cases we prove $F(z,p) \leq (9\sum_{i=1}^n \sigma_i^2)^{p/2}$, which finishes the induction. □

We also remark that Lemma 15 can be generalized to vectors

Corollary C.2. *Let $X_1, X_2, ..., X_n$ be random vectors where $\|X_i\|$ is σ_i-subexponential. Let $\sigma^2 = \sum_{i=1}^n \sigma_i^2$, $X = \sum_{i=1}^n X_i$. If X_i's are p-wise independent, for any even p*

$$\mathbb{E}[\|X - \mathbb{E}[X]\|^p] \leq (3\sigma)^p \cdot (2p)^p.$$

In particular, for all $t > 1$,

$$\Pr[\|X - \mathbb{E}[X]\| \geq 6\sigma p t] \leq 1/t^p.$$

Proof. The proof is exactly the same as the proof of Lemma 15. When X_i's are vectors we get exactly the same terms, except the terms have pair-wise inner-products. However, the inner-products $\langle X_i, X_j \rangle \leq \|X_i\|\|X_j\|$ so we only need to argue about the same inequality for $\|X_i\|$'s. □

D. Extended experiment

D.1. Verification of conditions

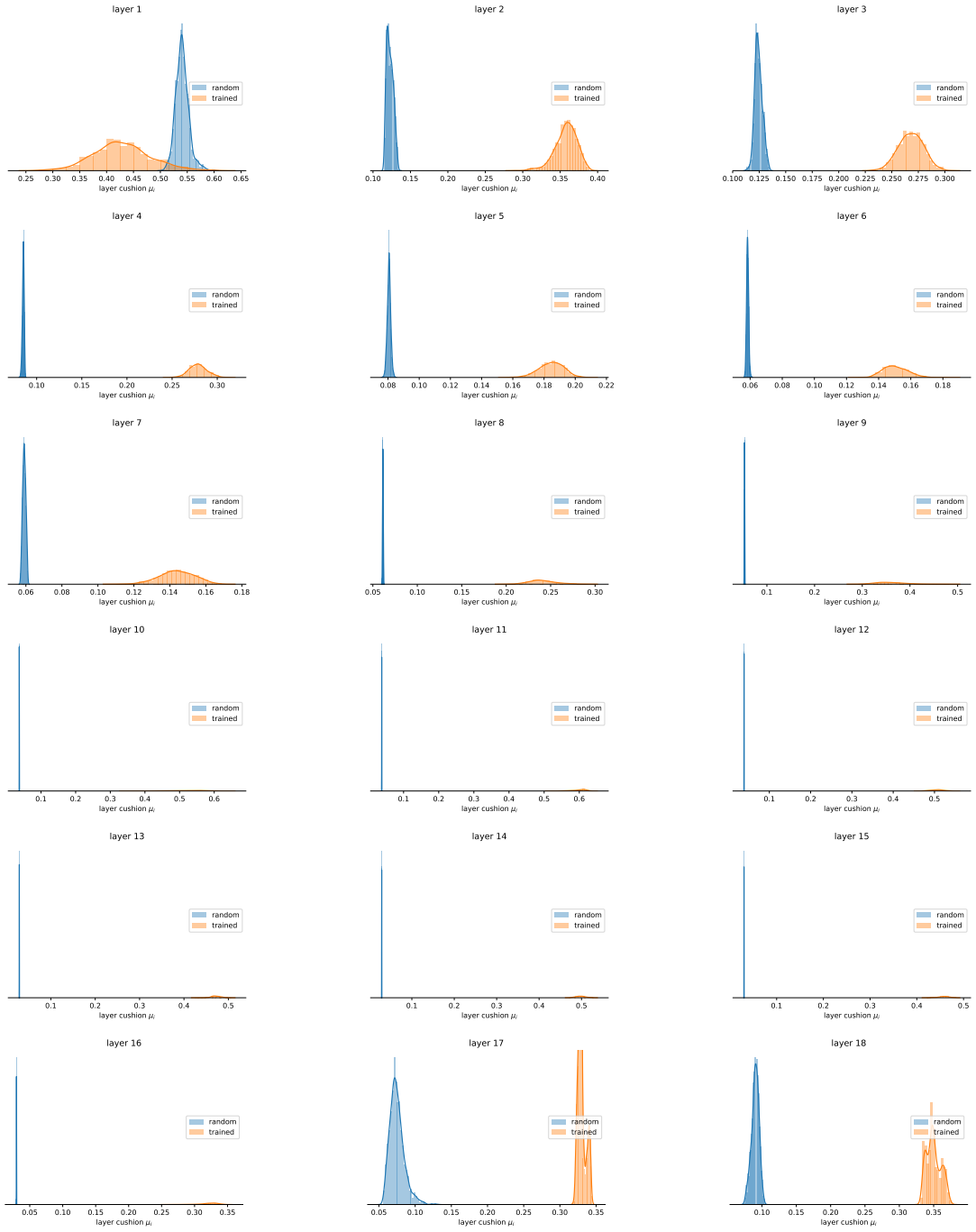

Figure A.1. Verification of layer cushion condition on the VGG-19 net

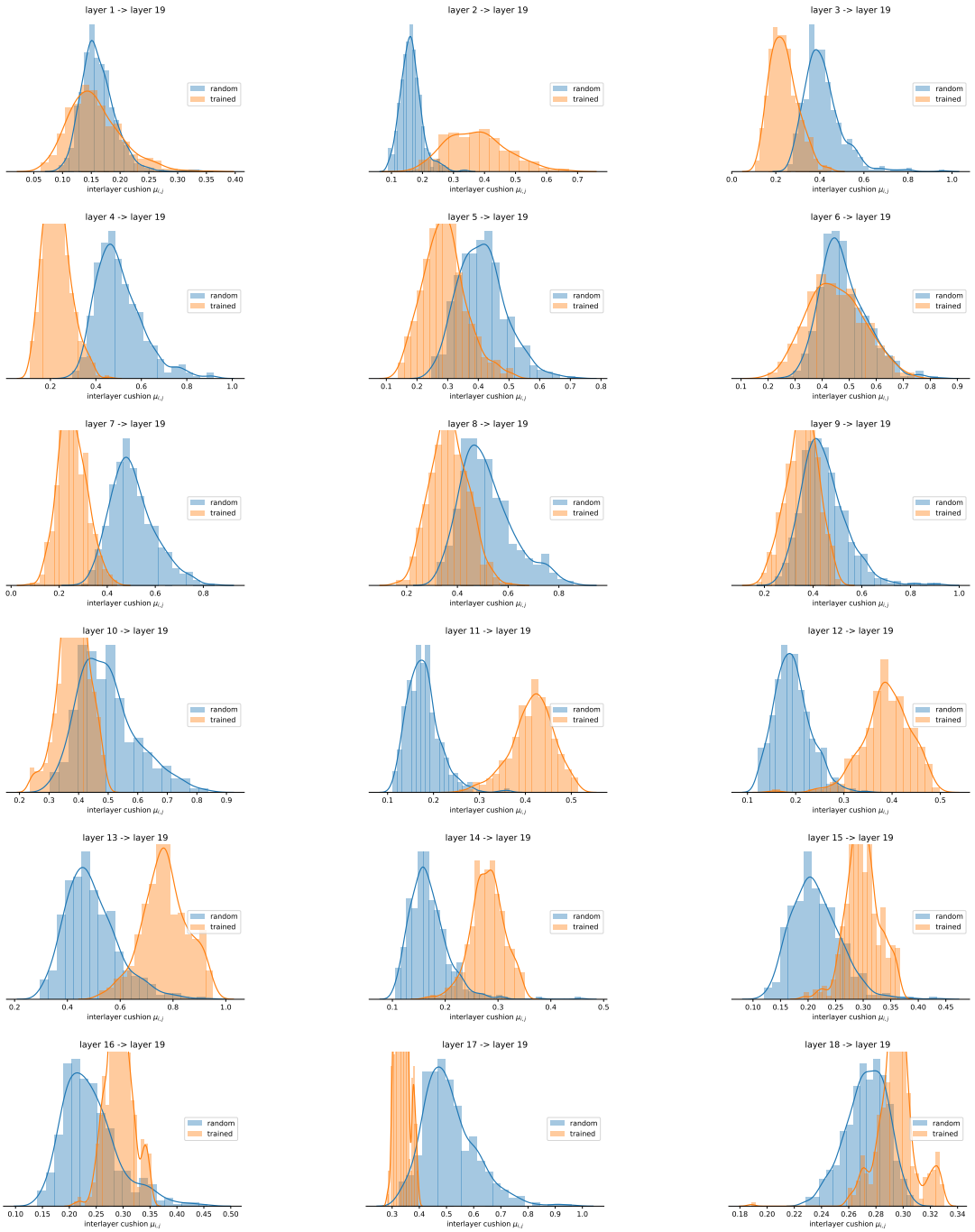

Figure A.2. Verification of interlayer cushion condition on the VGG-19 net

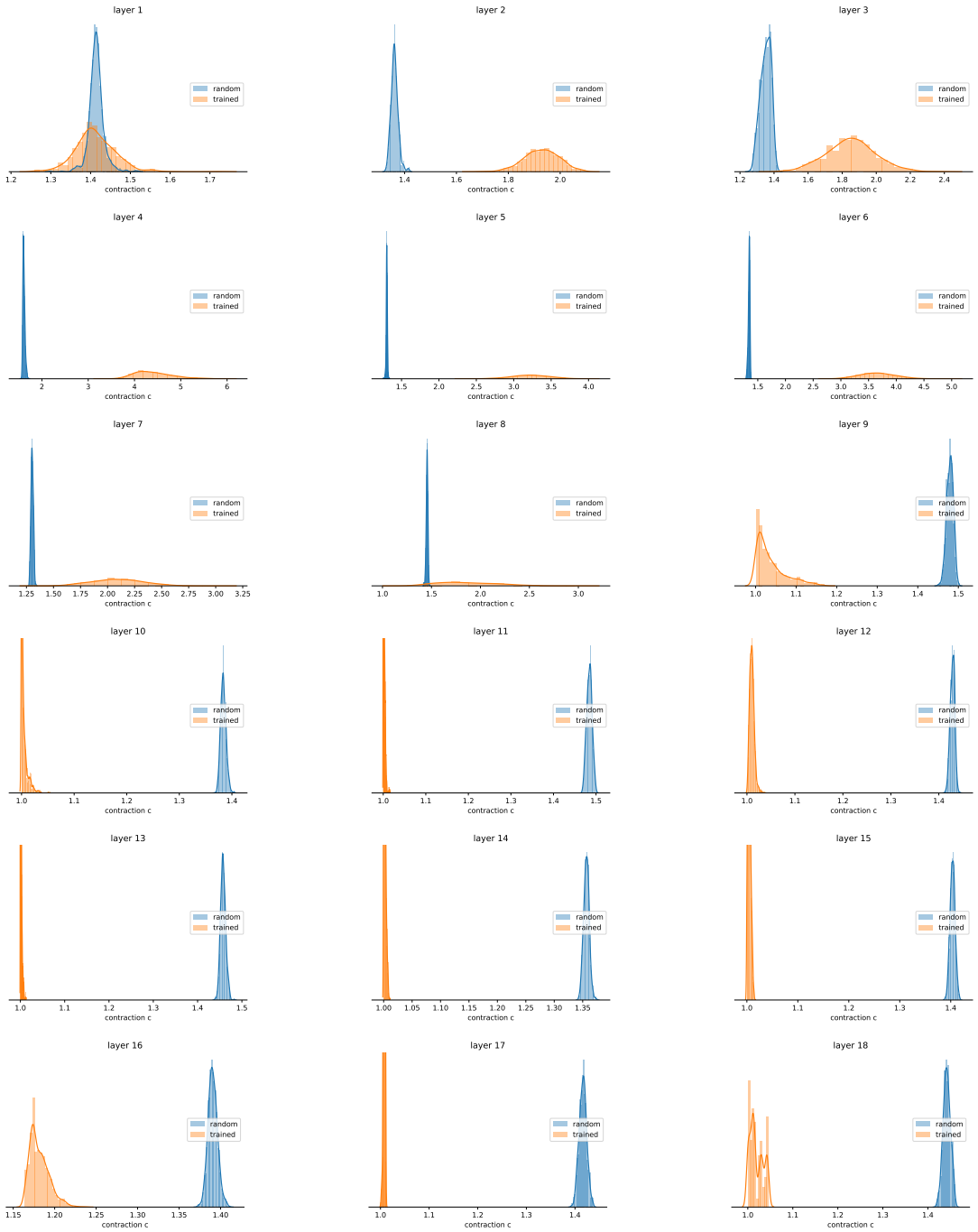

Figure A.3. Verification of activation contraction condition on the VGG-19 net

D.1.1. VERIFICATION OF INTERLAYER SMOOTHNESS CONDITION

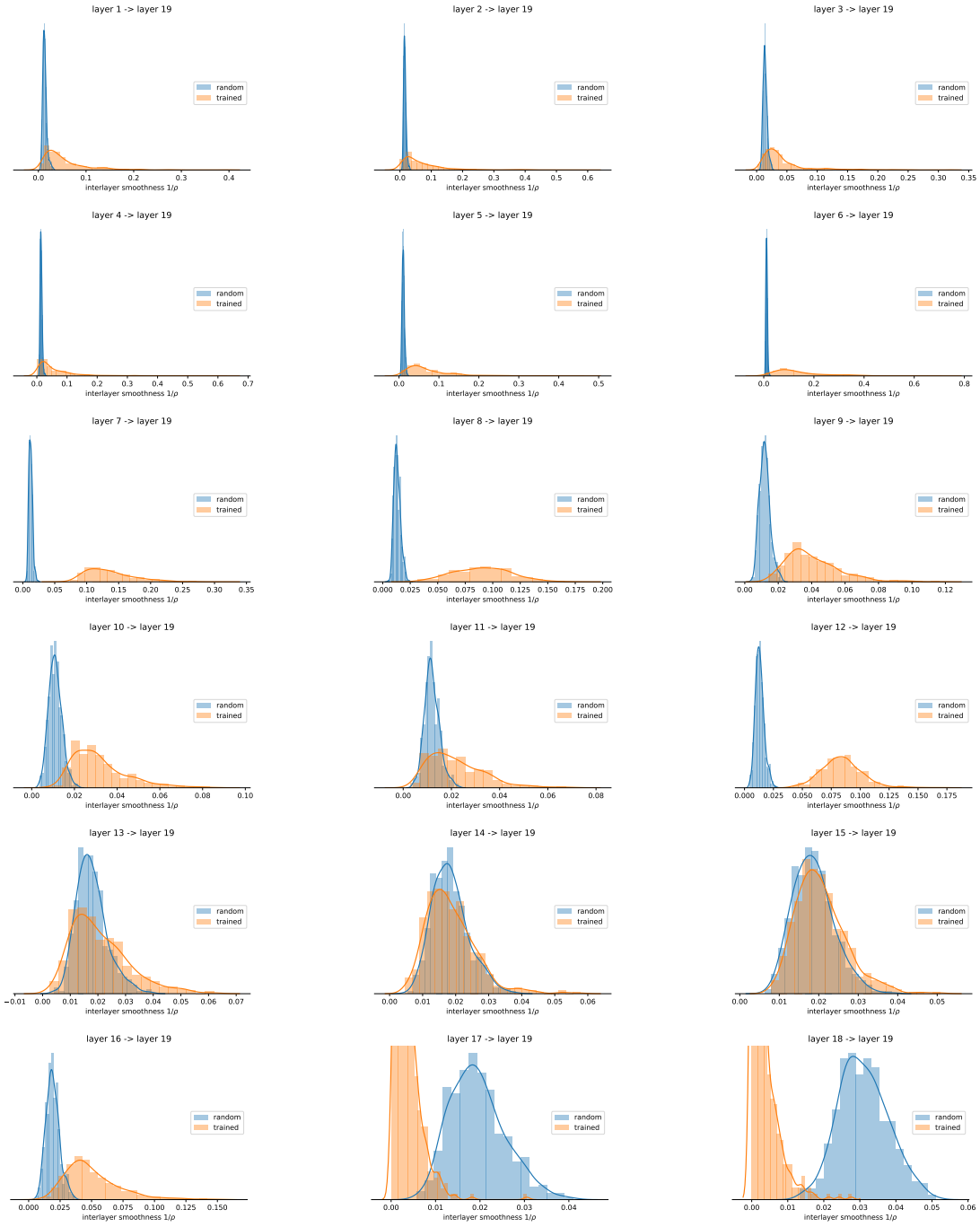

Figure A.4. Verification of interlayer smoothness condition on the VGG-19 net

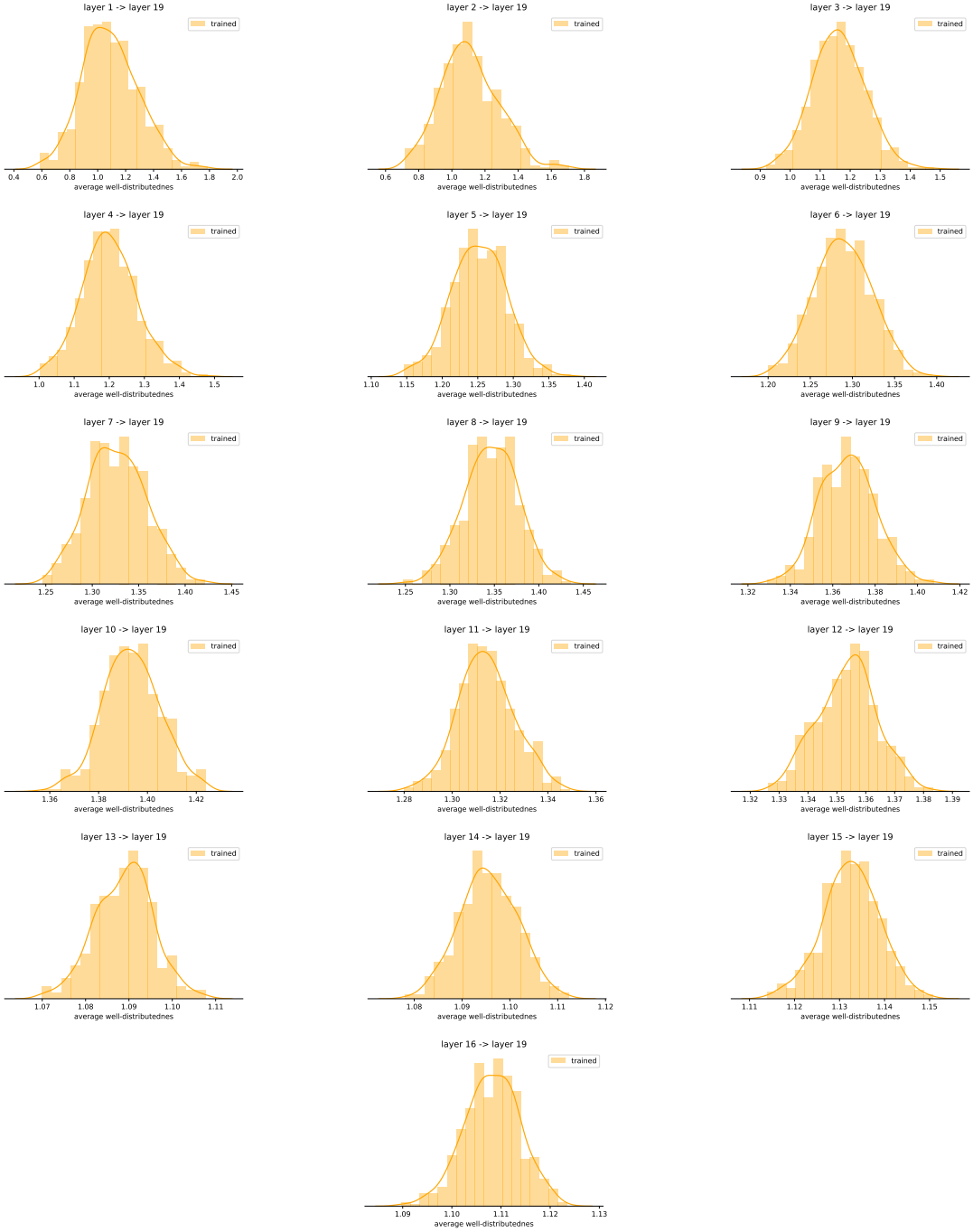

Figure A.5. Verification of well-distributedness of Jacobian condition on convolutional layers of the VGG-19 net. The histograms are generated by estimating the Frobenius norm of the Jacobians of the maps from certain layers to the final layer, restricted on randomly sampled pixels of the input feature maps. Since the well-distributedness parameter β is defined to be the largest over all the pixels, β should be read off from the upper tails of the histograms. Note for almost all layers, $\beta \approx 1$.

D.2. Effect of training on corrupted dataset

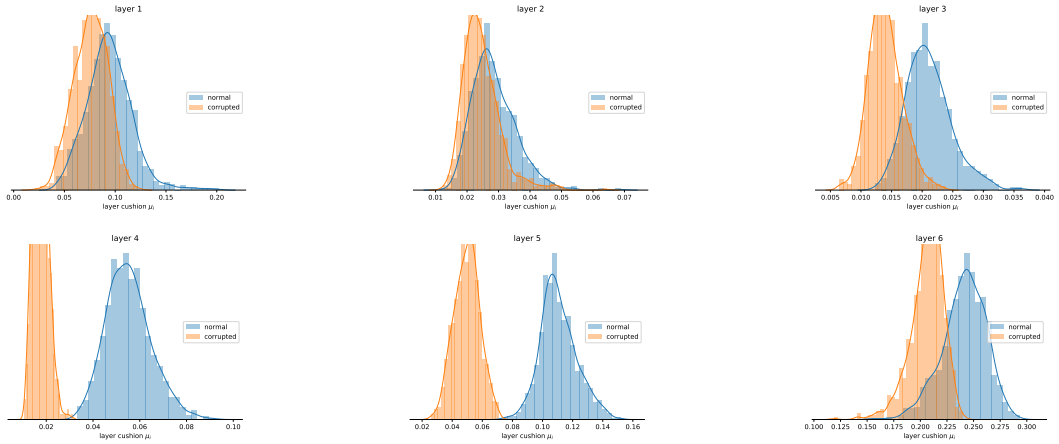

Figure A.6. Distribution of layer cushion of AlexNets trained on normal CIFAR-10 and corrupted CIFAR-10.

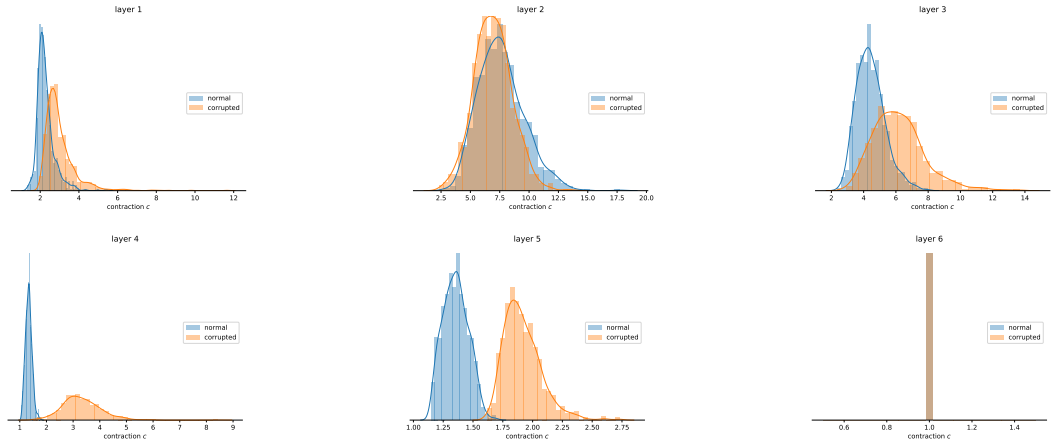

Figure A.7. Distribution of activation contraction of AlexNets trained on normal CIFAR-10 and corrupted CIFAR-10.

D.3. Comparing neural net generalization bounds

In Figure 3, we compare the following simplified generalization bounds:

- $\ell_{1,\infty}$: $\frac{B^2}{\gamma^2}\prod_{i=1}^{d}||A^i||_{1,\infty}$ (Bartlett and Mendelson, 2002)
- Frobenius: $\frac{B^2}{\gamma^2}\prod_{i=1}^{d}||A^i||_F^2$ (Neyshabur et al., 2015b; Golowich et al., 2017)
- spec $\ell_{1,2}$: $\frac{B^2}{\gamma^2}\prod_{i=1}^{d}||A_i||_2^2\sum_{i=1}^{d}\frac{||A^i||_{1,2}^2}{||A^i||_2^2}$ (Bartlett et al., 2017; Golowich et al., 2017)
- spec-fro: $\frac{B^2}{\gamma^2}\prod_{i=1}^{d}||A^i||_2^2\sum_{i=1}^{d}h_i\frac{||A^i||_F^2}{||A^i||_2^2}$ (Neyshabur et al., 2017a)
- ours: $\frac{B^2}{\gamma^2}\max_{x\in S}||f(x)||_2^2\sum_{i=1}^{d}\frac{\beta^2c_i^2\lceil\kappa/s\rceil^2}{\mu_i^2\mu_{i\rightarrow}^2}$

We further report the number of parameters as a simplified approximation of VC-dimension.