Supplementary Material

Supplementary material for the paper: "Understanding Generalization in Deep Learning via Tensor Methods". This appendix is organized as follows:

- Appendix A: Experimental details and additional results
- Appendix B: Technical definitions and propositions
- Appendix C: Main technical contributions
- Appendix D, E, and F: Generalization bounds on three types of neural networks: convolutional neural networks, fully-connected neural networks, and neural networks with residual connections
- Appendix G: Additional algorithms and algorithmic details

A Additional Experimental Results

A.1 Architecture and optimization setting

We train these four models (VGG-16, CP-VGG-16, WRN-28-10 and CP-WRN-28-10) using standard optimization settings with no dropouts and default initializations provided by PyTorch (Paszke et al., 2017). We use a SGD optimizer with momentum=0.9, weight decay=5e-4, and initial learning rate=0.05 to start the training process. The learning rate is scheduled to be divided by 2 every 30 epochs for VGG-16 and CP-VGG-16. While for WRN-28-10 and CP-WRN-28-10, the learning rate is scheduled to be divided by 5 at the $60^{\text{th}}, 120^{\text{th}}$ and 160^{th} epoch. We run 300 epochs to train each VGG-16 and CP-VGG-16, and we run 200 epochs to train each WRN-28-10 and CP-WRN-28-10.

A.2 Generalization bounds comparison with (Arora et al., 2018)

The generalization bound we calculated for a well-trained CP-VGG-16 (with the same # of parameters as VGG-16) on CIFAR10 dataset is around 12 (thus, of order 10^1) according to the transformation $f(x) = x/20 - 0.5$ applied in Figure 2b. Our evaluated bound is much better than naive counting of # parameters. Although we may not be able to directly compare our calculated bound with that in (Arora et al., 2018), which is roughly of order 10^5 as (Arora et al., 2018) uses a VGG-19 to evaluate their generalization bound while our evaluation is done using a CP-VGG-16, we present in Table 4 the effective number of parameters identified by our proposed bound. Compared with the effective number of parameters in (Arora et al., 2018) (Table 1 of (Arora et al., 2018)), we can see that **(1)** our effective number of parameters is upper bounded by the total number of parameters in original network (thus, the compression ratio is bounded by 1), while the effective number identified by (Arora et al., 2018) could be several times larger than the original number of parameters (e.g. based on Table 1 of (Arora et al., 2018), their effective number of parameters in layer 4 and 6 are more than 4 times of the original number of parameters); **(2)** the effective number of parameters in (Arora et al., 2018) ignores the dependence on depth, log factors and constants, while our effective number of parameters in Table 4 is exactly the actual number of parameters in the compressed network without these dependences.

A.3 Neural networks with CPL are natural for compression

The compression results in Table 5 are obtained directly without any fine tuning.

A.4 Improved Generalization Achieved by CPL

We provide additional experimental details in the improved generalization ability achieved by CPL under label noise setting. Our CPL combined with co-teaching (CT) (Han et al., 2018) outperforms SOTA method. Co-teaching (Han et al., 2018) is a training procedure for defeating label noise: it avoids overfitting to noisy labels by selecting clean samples out of the noisy ones and using them to update the network. Given the experimental results that neural networks with CPL tend to overfit less to noisy labels (Table 3), we combine Co-teaching to train networks with CPL on three different types of corrupted data (Table 3). The hyperparameters we use in these experiments are the same as the ones in Co-teaching [2].

As shown in Table 3, we compare our method CT+CPL against various label-noise methods (Han et al., 2018) under standard label noise setting (Han et al., 2018). **(1)** As shown in Figure 5, our method (CT+CPL) consistently outperforms the SOTA method with various choices of the number of components. **(1.1)** Specifically, according to Table 3, we see that combining CPL with co-teaching achieves the SOTA results on MNIST for

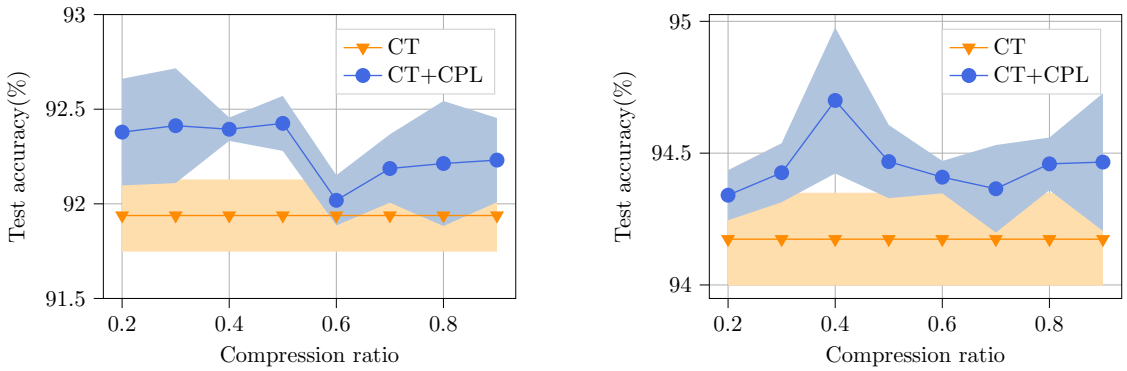
Table 4: Effective number of parameters identified by our proposed bound in Theorem 4.5.

layer	original # of params	our effective # of params	our compression ratio	effective # of params in Arora et al. (2018)	compression ratio in Arora et al. (2018)
1	1728	1694	0.980324	10728.622	6.208693
2	36864	36984	1.003255	63681.09	1.727460
3	73728	73932	1.002767	116967.945	1.586479
4	147456	147630	1.001180	910160.75	6.172423
5	294912	295106	1.000658	817337.9	2.771464
6	589824	590904	1.001831	3913927.2	6.635754
7	589824	590904	1.001831	15346982.0	26.019596
8	1179648	1177892	0.998511	367775.12	0.311767
9	2359296	2288242	0.969883	95893.41	0.040645
10	2359296	1774344	0.752065	87476.836	0.037078
11	2359296	350526	0.148572	42480.465	0.018006
12	2359296	42394	0.017969	40184.535	0.017032
13	2359296	124080	0.052592	137974.52	0.058481

Table 5: The compression results of a 28-layer Wide-ResNet equipped with CPL (CP-WRN-28-10) on CIFAR10 dataset. The compression is done via normalizing the CP spectrum and then deleting the components in CPL which have amplitudes smaller than the given cut-off-threshold.

Cut-off threshold	Compression ratio	# params	Test acc %
0	1×	36.5M	95.09
1e-4	0.229 (4×)	8.36M	95.08
1e-3	0.164 (5×)	6.90M	95.05
1e-2	0.124 (8×)	4.52M	**94.53**

PairFlip[3] with corruption rate 45% and Symmetric[4] with corruption rate 50%. **(1.2)** We also investigate the learning curve of our method compared with the SOTA (see Figure 6.). The models first reach best test accuracy early in the training, and then the test accuracy deteriorates as training goes on due to memorization effect. We see that our method always dominates the vanilla CT method when generalizability of the model starts to deteriorate due to memorization effect. This clearly shows that a neural network with CPL has better generalizability property than the plain neural network under this label noise setting. **(2)** For the Symmetric-20% in Table 3, as the label corruption rate is low, our method has a low effect in improving the generalization, which is expected.

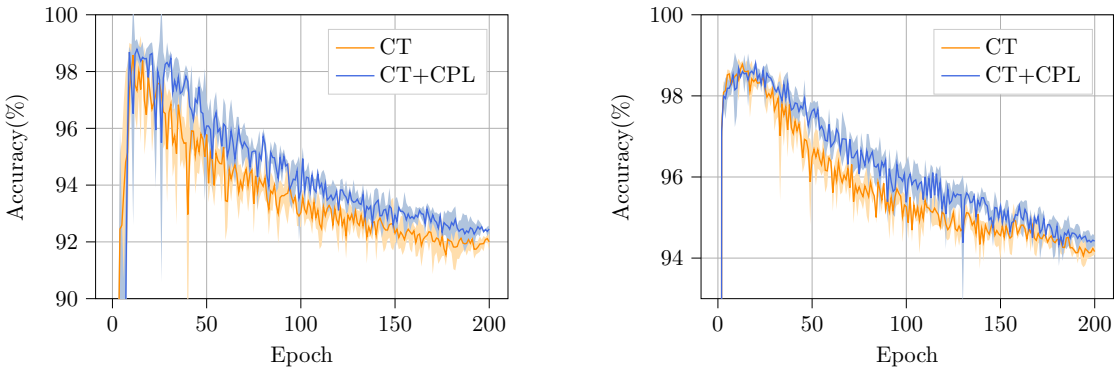


(a) PairFlip-45% **(b)** Symmetric-50%

Figure 5: Test accuracy vs. different compression ratios

Remark. The results displayed in Figure 5 and Figure 6 are based on our implementation of the CT method in

[3]PairFlip denotes that the label mistakes can only happen within very similar classes (Han et al., 2018)
[4]Symmetric denotes that the label mistakes may happen across different classes uniformly (Han et al., 2018)

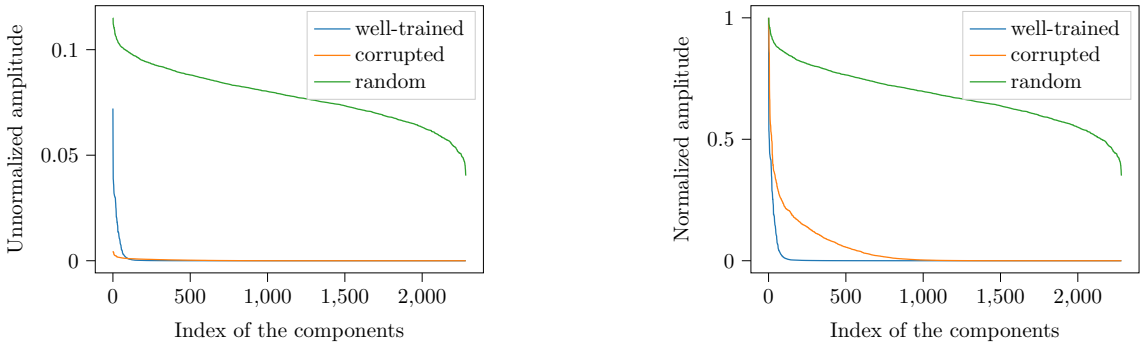


(a) PairFlip-45% **(b)** Symmetric-50%

Figure 6: Convergence plots of test accuracy vs. number of epochs on MNIST data

order to achieve a fair comparison, while the results displayed in Table 3 are based on the reported accuracies by (Han et al., 2018) as we would like to compare our CT+CPL with other different label-noise methods as well.

A.5 Compressibility of CPL: Property Evaluation CPL

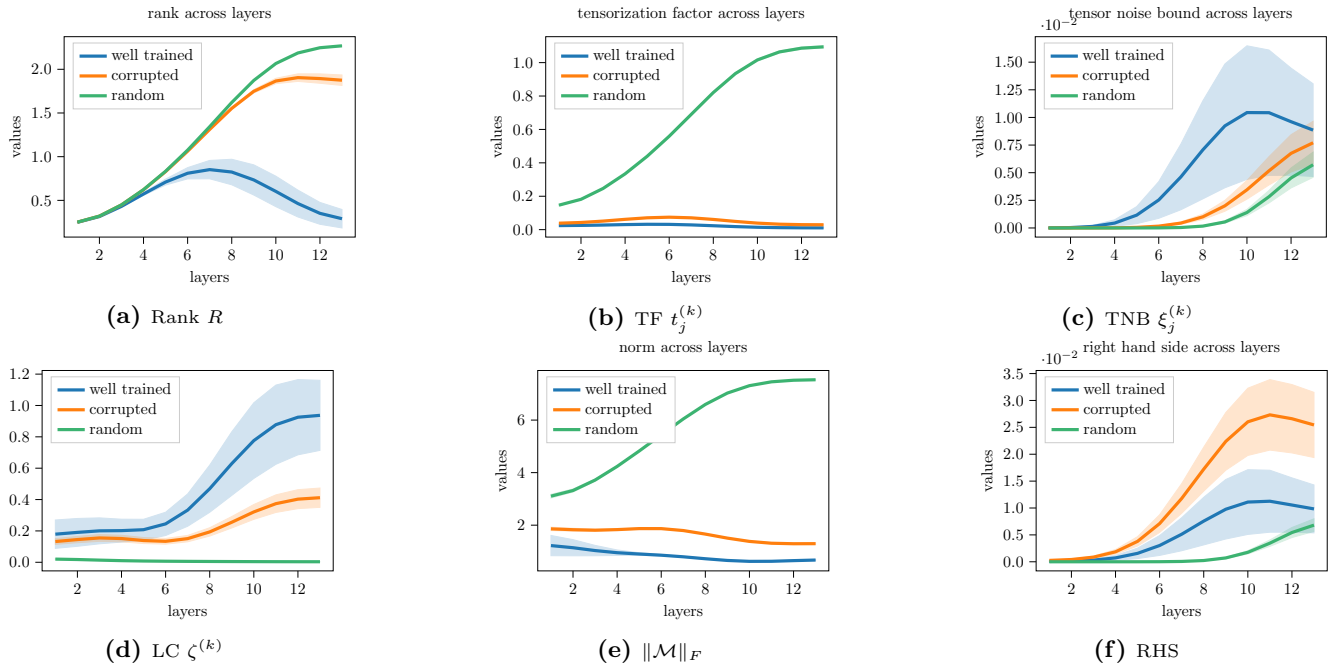


(a) CP spectrum with unnormalized amplitudes **(b)** CP spectrum with normalized amplitudes

Figure 7: Comparison of the CP spectra of a well-trained, a corrupted, and a randomly initialized CP-VGG-16 a the 13^{th} convolutional layer

Figure 7b displays the CP spectral of a well-trained, a corrupted, and a randomly initialized CP-VGG-16 (at the 13^{th} convolutional layer). For the unnormalized CP spectra of three models in Figure 7b(a), we can see that the largest amplitude in the CP spectrum of the corrupted CP-VGG-16 is much smaller than that of well-trained and random models. Yet, a smaller leading value in the CP spectrum does not necessarily mean that the corrupted is more low rank. As shown in Figure 7b(b), after normalizing the CP spectrum of each model by its largest amplitude, well-trained CP-VGG-16 still has the most low-rank CP spectrum (the blue curve) than that of corrupted or random models. Notice that the random model has the least low rankness since its weight tensors are the closest to random noise and thus it is hard to compress them.

We also compare our proposed properties among the three different sets of CP-VGG-16: well-trained, corrupted, and randomly initialized. As shown in Figure 8, since random models have the least compressibility as their weight tensors are closest to random noise, properties that focus more on the compressibility of the model are larger on random models (e.g tensor noise bound), which will lead to larger generalization bounds. In the meantime, properties that focus more on measuring the information loss after compression as well as the expressive power of the models (e.g. Fourier factors) are smaller for random models. The reason why well-trained models have the largest tensorization factor is in Figure 7b as the corrupted model usually has a very small leading value in its CP spectrum of later layers; yet as explained before, this does not necessarily indicate that corrupted models have more compressibility or low-rankness. The reason why the CP spectrum of corrupted models tend to have a small leading value is still a interesting question to study and we defer this to future work.



(a) Rank R **(b)** TF $t_j^{(k)}$ **(c)** TNB $\xi_j^{(k)}$

(d) LC $\zeta^{(k)}$ **(e)** $\|\mathcal{M}\|_F$ **(f)** RHS

Figure 8: Comparison of proposed properties among well-trained, corrupted and randomly-initialized CP-VGG-16 models

Optimization settings for obtaining the well-trained, corrupted, and randomly initialized models of CP-VGG-16. We obtain well-trained CP-VGG-16 using the same hyperparameter settings as mentioned in Appendix Section A.1. For corrupted CP-VGG-16, we train the model under 50% of label noise but using the same set of hyperparameters as the well-trained models. For CP-VGG-16 with random initialization, we just train the models for less than 1 epoch. For each set of these models, we obtain 200 instances using different random seeds.

B Common Definitions and Propositions

In this section, we will briefly review three key concepts underlying all analysis in this work, including *(multidimensional discrete Fourier transform)*, *CP decomposition* and *2D-convolutional layer* in neural networks.

B.1 Multidimensional Discrete Fourier Transform (MDFT)

Definition B.1. (Multidimensional discrete Fourier transform, MDFT) An m-dimensional MDFT $\mathcal{F}_m$ defines a mapping from an m-order tensor $\mathcal{X} \in \mathbb{R}^{N_1 \times \cdots \times N_m}$ to another *complex* m-order tensor $\tilde{\mathcal{X}} \in \mathbb{C}^{N_1 \times \cdots \times N_m}$ such that

$$\tilde{\mathcal{X}}_{f_1,\ldots,f_m} = \left(\prod_{l=1}^{m} N_l\right)^{-\frac{1}{2}} \sum_{n_1=1}^{N_1} \cdots \sum_{n_m=1}^{N_m} \mathcal{X}_{n_1,\cdots,n_m} \left(\prod_{l=1}^{m} \omega_{N_l}^{f_l n_l}\right) \tag{6}$$

where $\omega_{N_l} = \exp(-\mathrm{j}2\pi/N_l)$ and $(\prod_{l=1}^{m} N_l)^{-\frac{1}{2}}$ is the *normalization factor* that makes $\mathcal{F}_m$ unitary. Through out the paper, we will use symbols with tilde (e.g. $\tilde{\mathcal{X}}$) to denote tensors after MDFT.

MDFT can also be applied on a subset of the dimensions $\mathcal{I} \subseteq [m]$, and in this case we denote the mapping as $\mathcal{F}_m^{\mathcal{I}}$.

$$\tilde{\mathcal{X}}_{i_1,\ldots,i_m} = \left(\prod_{l\in\mathcal{I}} N_l\right)^{-\frac{1}{2}} \sum_{\forall l\in\mathcal{I}} \mathcal{X}_{n_1,\cdots,n_m} \left(\prod_{l\in\mathcal{I}} \omega_{N_l}^{f_l n_l}\right) \tag{7}$$

where $i_l = f_l$ if $l \in \mathcal{I}$ and $i_l = n_l$ for $l \notin \mathcal{I}$.

Fact B.2. (Separability of MDFT) An m-dimensional MDFT $\mathcal{F}_m$ is equivalent to a composition of m unidimensional DFTs, i.e.

$$\mathcal{F}_m = \mathcal{F}_m^1 \circ \mathcal{F}_m^2 \circ \cdots \circ \mathcal{F}_m^m \tag{8}$$

Similarly, $\mathcal{F}_m^{\mathcal{I}}$ is identical to a composition of $|\mathcal{I}|$ unidimensional DFTs over corresponding dimensions.

Fact B.3. (MDFT is unitary) For an MDFT $\mathcal{F}$, its adjoint $\mathcal{F}^*$ is equal to its inverse $\mathcal{F}^{-1}$, i.e. $\mathcal{F}^* = \mathcal{F}^{-1}$. An immediate corollary of this property is that the operator norm is invariant to MDFT: Given an operator $\mathcal{A}$, its operator norm of $\mathcal{A}$ is equal to $\mathcal{F}^*\mathcal{A}\mathcal{F}$, i.e. $\|\mathcal{A}\| = \|\mathcal{F}^*\mathcal{A}\mathcal{F}\|$.

B.2 CP decomposition

Definition B.4. (CP decomposition) Given an m-order tensor $\mathcal{T} \in \mathbb{R}^{N_1 \times \cdots \times N_m}$, a CP decomposition factorizes $\mathcal{T}$ into m *core factors* $\{\boldsymbol{K}^l\}_{l=1}^m$ with $\boldsymbol{K}^l \in \mathbb{R}^{R \times N_l}$ (with its r^{th} column as $\boldsymbol{k}_r^l \in \mathbb{R}^{N_l}$) such that

$$\mathcal{T} = \sum_{r=1}^{R} \lambda_r \boldsymbol{k}_r^1 \otimes \cdots \otimes \boldsymbol{k}_r^m \tag{9a}$$

$$\mathcal{T}_{n_1,\cdots,n_m} = \sum_{r=1}^{R} \lambda_r \boldsymbol{K}_{r,n_1}^1 \cdots \boldsymbol{K}_{r,n_m}^m \tag{9b}$$

where each column $\boldsymbol{k}_r^l$ has unit ℓ_2 norm, i.e. $\|\boldsymbol{k}_r^l\|_2 = 1, \forall r \in [R], l \in [m]$. Without loss of generality, we assume the *CP eigenvalues* are positive and sorted in decreasing order, i.e. $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m > 0$. If the columns in $\boldsymbol{K}^l$ are orthogonal, i.e. $\langle \boldsymbol{k}_r^l, \boldsymbol{k}_{r'}^l \rangle = 1$ for $r \neq r'$, the factorization is further named as *orthogonal CP decomposition*.

Lemma B.5. (MDFT of CP decomposition) If an m-order tensor $\mathcal{T} \in \mathbb{R}^{N_1 \times \cdots \times N_m}$ takes a CP decomposition as in Eq. (9a), its (all-dimensional) MDFT $\tilde{\mathcal{T}} = \mathcal{F}_m(\mathcal{T}) \in \mathbb{C}^{N_1 \times \cdots \times N_m}$ also takes a CP format as

$$\tilde{\mathcal{T}} = \sum_{r=1}^{R} \lambda_r \tilde{\boldsymbol{k}}_r^1 \otimes \cdots \otimes \tilde{\boldsymbol{k}}_r^m \tag{10}$$

$$\tilde{\mathcal{T}}_{f_1,\cdots,f_m} = \sum_{r=1}^{R} \lambda_r \tilde{\boldsymbol{K}}_{r,f_1}^1 \cdots \tilde{\boldsymbol{K}}_{r,f_m}^m \tag{11}$$

where $\tilde{\boldsymbol{K}}^l = \mathcal{F}_2^2(\boldsymbol{K}^l), \forall l \in [m]$. The result can be extended to MDFT where a subset of dimensions are transformed.

Proof. (of Lemma B.5) According to the definition of multidimensional discrete Fourier transform, we have

$$\tilde{\mathcal{T}}_{n_1,\cdots,n_m} = \left(\prod_{l=1}^{m} N_l\right)^{-\frac{1}{2}} \sum_{n_1=1}^{N_1} \cdots \sum_{n_m=1}^{N_m} \mathcal{T}_{n_1,\cdots,n_m} \left(\prod_{l=1}^{m} \omega_{N_l}^{f_l n_l}\right) \tag{12}$$

$$= \left(\prod_{l=1}^{m} N_l\right)^{-\frac{1}{2}} \sum_{n_1=1}^{N_1} \cdots \sum_{n_m=1}^{N_m} \left(\sum_{r=1}^{R} \lambda_r \tilde{\boldsymbol{K}}_{r,n_1}^1 \cdots \tilde{\boldsymbol{K}}_{r,n_m}^m\right) \left(\prod_{l=1}^{m} \omega_{N_l}^{f_l n_l}\right) \tag{13}$$

$$= \sum_{r=1}^{R} \lambda_r \left(N_1^{-1/2} \sum_{n_1=1}^{N_1} \boldsymbol{K}_{r,n_1}^1 \omega_{N_1}^{f_1 n_1}\right) \cdots \left(N_m^{-1/2} \sum_{n_m=1}^{N_m} \boldsymbol{K}_{r,n_m}^m \omega_{N_m}^{f_m n_m}\right) \tag{14}$$

$$= \sum_{r=1}^{R} \lambda_r \tilde{\boldsymbol{K}}_{r,f_1}^1 \cdots \tilde{\boldsymbol{K}}_{r,f_m}^m \tag{15}$$

which completes the proof. □

B.3 2D-Convolutional Layer in Neural Networks

Definition B.6. (2D-convolutional layer) In CNNs, a 2D-convolutional layer is parametrized by a 4^{th}-order tensor $\mathcal{M} \in \mathbb{R}^{k_x \times k_y \times T \times S}$ (with $k_x \times k_y$ kernels). It defines a mapping from a 3^{rd}-order input tensor $\mathcal{X} \in \mathbb{R}^{H \times W \times S}$ (with S channels) to another 3^{rd}-order output tensor $\mathcal{Y} \in \mathbb{R}^{H \times W \times T}$ (with T channels).

$$\mathcal{Y}_{:,:,t} = \sum_{s=1}^{S} \mathcal{M}_{:,:,t,s} * \mathcal{X}_{:,:,s} \tag{16}$$

$$\mathcal{Y}_{i,j,t} = \sum_{s=1}^{S} \sum_{p,q} \mathcal{M}_{i-p,j-q,t,s} \mathcal{X}_{p,q,s} \tag{17}$$

where $*$ represents a 2D-convolution operator.

Lemma B.7. (Convolutional theorem of 2D-convolutional layer) Suppose $\tilde{\mathcal{X}} = \mathcal{F}_3^{1,2}(\mathcal{X}) \in \mathbb{C}^{H\times W\times S}$, $\tilde{\mathcal{M}} = \mathcal{F}_4^{1,2}(\mathcal{M}) \in \mathbb{C}^{H\times W\times T\times S}$ and $\tilde{\mathcal{Y}} = \mathcal{F}_3^{1,2}(\mathcal{Y}) \in \mathbb{C}^{H\times W\times T}$ are the MDFT of input, weights and outputs tensors $\mathcal{X}$, $\mathcal{W}$ and $\mathcal{Y}$ respectively, then these three tensors satisfy the following equation:

$$\tilde{\mathcal{Y}}_{f,g,t} = \sqrt{HW}\sum_{s=1}^{S}\tilde{\mathcal{M}}_{f,g,t,s}\tilde{\mathcal{X}}_{f,g,s} \tag{18}$$

Notice that the equation has a constant $\sqrt{HW}$ since we use a normalized MDFT.

Proof. (of Lemma B.7) The theorem can be easily proved by applying MDFT on both sides of Eq. (17).

$$\begin{aligned}
\tilde{\mathcal{Y}}_{f,g,t} &= \frac{1}{\sqrt{HW}}\sum_{i,j}\mathcal{Y}_{i,j,t}\omega_H^{if}\omega_W^{jg} && (19)\\
&= \frac{1}{\sqrt{HW}}\sum_{i,j}\left(\sum_{s=1}^{S}\sum_{p,q}\mathcal{M}_{i-p,j-q,t,s}\mathcal{X}_{p,q,s}\right)\omega_H^{if}\omega_W^{jg} && (20)\\
&= \sqrt{HW}\sum_{s=1}^{S}\left(\frac{1}{\sqrt{HW}}\sum_{i,j}\mathcal{M}_{i-p,j-q,t,s}\omega_H^{(i-p)f}\omega_W^{(j-q)g}\right)\left(\frac{1}{\sqrt{HW}}\sum_{p,q}\mathcal{X}_{p,q,s}\omega_H^{pf}\omega_W^{qg}\right) && (21)\\
&= \sqrt{HW}\sum_{s=1}^{S}\tilde{\mathcal{M}}_{f,g,t,s}\tilde{\mathcal{X}}_{f,g,s} && (22)
\end{aligned}$$

□

Lemma B.8. (Operator norm of 2D-convolutional layer) Suppose we rewrite the tensors in matrix/vector form, i.e. $\tilde{\mathcal{X}}_{f,g,s} = \tilde{\boldsymbol{x}}_s^{(f,g)}$, $\tilde{\mathcal{M}}_{f,g,t} = \tilde{\boldsymbol{M}}_{t,s}^{(f,g)}$, $\tilde{\mathcal{Y}}_{f,g,t} = \tilde{\boldsymbol{y}}_t^{(f,g)}$, then Eq. (18) can be written using matrix/vector products:

$$\tilde{\boldsymbol{y}}_t^{(f,g)} = \sum_{s=1}^{S}\tilde{\boldsymbol{M}}_{t,s}^{(f,g)}\tilde{\boldsymbol{y}}_t^{(f,g)},\ \forall f,g \tag{23}$$

The operator norm of $\mathcal{M}$, defined as $\|\mathcal{M}\| = \max_{\|\mathcal{X}\|_F=1}\|\mathcal{Y}\|_F$, can be obtained by spectral norms of $\tilde{\boldsymbol{M}}^{(f,g)}$ as:

$$\|\mathcal{M}\| = \sqrt{HW}\max_{f,g}\left\|\boldsymbol{M}^{(f,g)}\right\|_2 \tag{24}$$

Remarks. The bound is first given by Sedghi et al. (2018). In this work, we provide a much simpler proof compared to the original one in Sedghi et al. (2018). In the next section, we show that the bound can be computed without evaluating the spectral norm if the weights tensor $\mathcal{M}$ takes a CP format similar to Eq. (9a).

Proof. (of Lemma B.8) From Fact B.3, we know that $\|\mathcal{M}\| = \|\tilde{\mathcal{M}}\|$, where $\|\tilde{\mathcal{M}}\| = \max_{\|\tilde{\mathcal{X}}\|_F=1}\|\tilde{\mathcal{Y}}\|_F$. Next, we bound $\|\tilde{\mathcal{Y}}\|_F^2$ (i.e. $\sum_{f,g}\left\|\tilde{\boldsymbol{y}}^{(f,g)}\right\|_F^2$) assuming $\|\tilde{\mathcal{X}}\|_2^2 = 1$ (i.e. $\sum_{f,g}\left\|\tilde{\boldsymbol{x}}^{(f,g)}\right\|_2^2 = 1$).

$$\begin{aligned}
\|\tilde{\mathcal{Y}}\|_F^2 &= \sum_{f,g}\left\|\tilde{\boldsymbol{y}}^{(f,g)}\right\|_2^2 && (25)\\
&\le HW\sum_{f,g}\left\|\tilde{\boldsymbol{M}}^{(f,g)}\right\|^2\left\|\tilde{\boldsymbol{x}}^{(f,g)}\right\|_2^2 && (26)\\
&\le HW\max_{f,g}\left\|\tilde{\boldsymbol{M}}^{(f,g)}\right\|^2\sum_{f,g}\left\|\tilde{\boldsymbol{x}}^{(f,g)}\right\|_2^2 && (27)\\
&= HW\max_{f,g}\left\|\tilde{\boldsymbol{M}}^{(f,g)}\right\|^2 && (28)\\
\|\tilde{\mathcal{Y}}\|_F &\le \sqrt{HW}\max_{f,g}\left\|\tilde{\boldsymbol{M}}^{(f,g)}\right\| && (29)
\end{aligned}$$

We complete the proof by observing all inequalities can achieve equality simultaneously. □

Definition B.9. (Tensor product) For vectors $\boldsymbol{a} \in \mathbb{R}^n$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\boldsymbol{c} \in \mathbb{R}^p$, their tensor product $\boldsymbol{a} \otimes \boldsymbol{b} \otimes \boldsymbol{c}$ is a 3-way tensor in $\mathbb{R}^{m \times n \times p}$, with the $(i,j,k)^{\text{th}}$ entry being $\boldsymbol{a}_i \boldsymbol{b}_j \boldsymbol{c}_k$. Similarly, for a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ and a vector $\boldsymbol{c} \in \mathbb{R}^p$, their tensor product $\boldsymbol{A} \otimes \boldsymbol{c}$ is a $m \times n \times p$ tensor with the $(i,j,k)^{\text{th}}$ entry being $\boldsymbol{A}_{ij} \boldsymbol{c}_k$.

Definition B.10. (Kronecker product). Let $\boldsymbol{A}$ be an $n \times p$ matrix and $\boldsymbol{B}$ an $m \times q$ matrix. The $mn \times pq$ matrix

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{1,1}\boldsymbol{B} & a_{1,2}\boldsymbol{B} & \cdots & a_{1,p}\boldsymbol{B} \\ a_{2,1}\boldsymbol{B} & a_{2,2}\boldsymbol{B} & \cdots & a_{2,p}\boldsymbol{B} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n,1}\boldsymbol{B} & a_{n,2}\boldsymbol{B} & \cdots & a_{n,p}\boldsymbol{B} \end{bmatrix}$$

is called the *Kronecker product* of A and B. The outer product is an instance of *Kronecker products*.

C CP Layers in Tensorial Neural Networks

In this section, we will introduce three types of neural network layers, whose parameters are factorized in CP format as in Eq. (9a) (with small variations). For brevity, we omit the layer superscript and denote the input, layer parameters and output as $\mathcal{X}$, $\mathcal{M}$ and $\mathcal{Y}$, and we use $\mathcal{Y} = \mathcal{M}(\mathcal{X})$ to denote the relations between $\mathcal{X}$, $\mathcal{M}$ and $\mathcal{Y}$.

C.1 CP 2D-convolutional Layer

Definition C.1. (CP 2D-convolutional layer) For a given 2D-convolutional layer in Eq. (17), a CP decomposition factorizes the weights tensor $\mathcal{M} \in \mathbb{R}^{H \times W \times T \times S}$ into three *core factors* $\mathcal{C} \in \mathbb{R}^{R \times k_x \times k_y}$, $\boldsymbol{U} \in \mathbb{R}^{R \times T}$, $\boldsymbol{V} \in \mathbb{R}^{R \times S}$ and a vector of *CP eigenvalues* $\lambda \in \mathbb{R}^R$ such that

$$\mathcal{M} = \sum_{r=1}^{R} \lambda_r \mathcal{C}_r \otimes \boldsymbol{u}_r \otimes \boldsymbol{v}_r \tag{30}$$

$$\mathcal{M}_{i,j,t,s} = \sum_{r=1}^{R} \lambda_r \mathcal{C}_{r,i,j} \boldsymbol{U}_{r,t} \boldsymbol{V}_{r,s} \tag{31}$$

where $\lambda_r > 0$, $\|\mathcal{C}_r\|_F = 1$, $\|\boldsymbol{u}_r\|_2 = 1$ and $\|\boldsymbol{v}_r\|_2 = 1$ for all $r \in [R]$.

Lemma C.2. (Operator norm of CP 2D-convolutional layer) For a 2D-convolutional layer whose weights tensor takes a CP format as in Eq. (31), the operator norm $\|\mathcal{M}\|$ is bound by the CP eigenvalues λ as

$$\|\mathcal{M}\| \leq \sqrt{HW} \sum_{r=1}^{R} |\lambda_r| \max_{f,g} \left| \tilde{\mathcal{C}}_r^{(f,g)} \right| \tag{32}$$

Proof. (of Lemma C.2) From Fact B.3, the operator norm of $\mathcal{M}$ is equal to the one of its MDFT $\tilde{\mathcal{M}} = \mathcal{F}_4^{1,2}(\mathcal{M})$, i.e. $\|\mathcal{M}\| = \|\tilde{\mathcal{M}}\|$. According to Lemma B.8, it is sufficient to compute the spectral norm for each matrix $\tilde{\boldsymbol{M}}^{(f,g)}$ individually. Notice that if $\mathcal{M}$ takes a CP format, each $\tilde{\boldsymbol{M}}^{(f,g)}$ has a decomposed form as follows

$$\tilde{\boldsymbol{M}}^{(f,g)} = \sum_{r=1}^{R} \lambda_r \tilde{\mathcal{C}}_r^{(f,g)} \boldsymbol{u}_r \boldsymbol{v}_r^\top \tag{33a}$$

$$\tilde{\boldsymbol{M}}_{t,s}^{(f,g)} = \sum_{r=1}^{R} \lambda_r \tilde{\mathcal{C}}_r^{(f,g)} \boldsymbol{U}_{r,t} \boldsymbol{V}_{r,s} \tag{33b}$$

where $\tilde{\mathcal{C}} = \mathcal{F}_3^{2,3}(\mathcal{C})$ and $\tilde{\mathcal{C}}_r^{(f,g)} = \tilde{\mathcal{C}}_{r,f,g}$. The rest of the proof follows the definition of spectral norm of $\tilde{\boldsymbol{M}}$, i.e.

$\|\tilde{\boldsymbol{M}}^{(f,g)}\|_2 = \max_{\|\boldsymbol{a}\|=1} \|\tilde{\boldsymbol{M}}^{(f,g)}\boldsymbol{a}\|$. Let $\boldsymbol{b} = \tilde{\boldsymbol{M}}^{(f,g)}\boldsymbol{a}$, we can bound the ℓ_2 norm of $\boldsymbol{b}$:

$$\|\boldsymbol{b}\|_2 = \left\|\tilde{\boldsymbol{M}}^{(f,g)}\boldsymbol{a}\right\|_2 = \left\|\sum_{r=1}^{R} \lambda_r \tilde{\mathcal{C}}_r^{(f,g)} \boldsymbol{u}_r \boldsymbol{V}_r^\top \boldsymbol{a}\right\|_2 \tag{34}$$

$$\leq \sum_{r=1}^{R} \left|\lambda_r \tilde{\mathcal{C}}_r^{(f,g)} \left(\boldsymbol{v}_r^\top \boldsymbol{a}\right)\right| \|\boldsymbol{u_r}\|_2 \tag{35}$$

$$= \sum_{r=1}^{R} \left|\lambda_r \tilde{\mathcal{C}}_r^{(f,g)} \left(\boldsymbol{v}_r^\top \boldsymbol{a}\right)\right| \tag{36}$$

$$\leq \sum_{r=1}^{R} |\lambda_r| \left|\tilde{\mathcal{C}}_r^{(f,g)}\right| \tag{37}$$

Therefore, $\|\mathcal{M}\| = \|\tilde{\mathcal{M}}\| = \sqrt{HW}\max_{f,g}\|\tilde{\boldsymbol{M}}^{(f,g)}\| \leq \sqrt{HW}\sum_{r=1}^{R}|\lambda_r|\max_{f,g}\left|\tilde{\mathcal{C}}_r^{(f,g)}\right|$. □

C.2 Higher-order CP Fully-connected Layer

Definition C.3. (Higher-order fully-connected layer) The layer is parameterized by a $2m^{\text{th}}$-order tensor $\mathcal{M} \in \mathbb{R}^{T_1\times\cdots\times T_m\times S_1\times\cdots\times S_m}$. It maps an m^{th}-order input tensor $\mathcal{X} \in \mathbb{R}^{S_1\times\cdots\times S_m}$ to another m^{th}-order output tensor $\mathcal{Y} \in \mathbb{R}^{T_1\times\cdots\times S_m}$ with the following equation:

$$\mathcal{Y}_{t_1,\cdots,t_m} = \sum_{\forall l: S_l} \mathcal{M}_{t_1,\cdots,t_m,s_1,\cdots,s_m}\mathcal{X}_{s_1,\cdots,s_m} \tag{38}$$

Definition C.4. (Higher-order CP fully-connected layer) Given a higher-order fully-connected layer in Eq. (38), a CP decomposition factorizes the weights tensor $\mathcal{M} \in \mathbb{R}^{T_1\times\cdots\times T_m\times S_1\times\cdots\times S_m}$ into m core factors $\mathcal{K}^m \in \mathbb{R}^{R\times T_m\times S_m}$.

$$\mathcal{M}_{t_1,\cdots,t_m,s_1,\cdots,s_m} = \sum_{r=1}^{R} \lambda_r \mathcal{K}^1_{r,t_1,s_1}\cdots\mathcal{K}^m_{r,t_m,s_m} \tag{39}$$

For simplicity, we denote the r^{th} slice of $\mathcal{K}^l$ as $\boldsymbol{K}_r^l = \mathcal{K}^l_{r,:,:}$. We assume $\boldsymbol{K}_r^l$ has unit Frobenius norm, i.e. $\|\boldsymbol{K}_r^l\|_F = 1$ and $\lambda_r > 0$ for all $r \in [R]$.

Lemma C.5. (Operator norm of higher-order CP fully-connected layer) For a higher-order fully layer whose weights tensor takes a CP format as in Eq. (39), the operator norm $\|\mathcal{M}\|$ is bound by the CP eigenvalues λ as

$$\|\mathcal{M}\| \leq \sum_{r=1}^{R} |\lambda_r| \tag{40}$$

Proof. (of Lemma C.5) The proof follows directly the definition of operator norm $\|\mathcal{M}\| = \max_{\|\mathcal{X}\|_F=1}\|\mathcal{Y}\|_F$.

$$\|\mathcal{Y}\|_F \leq \sum_{r=1}^{R} |\lambda_r| \left\|\boldsymbol{K}_m^l\right\|_2 \cdots \left\|\boldsymbol{K}_1^l\right\|_2 \|\mathcal{X}\|_F \tag{41}$$

$$\leq \sum_{r=1}^{R} |\lambda_r| \left\|\boldsymbol{K}_m^l\right\|_F \cdots \left\|\boldsymbol{K}_1^l\right\|_F \|\mathcal{X}\|_F \tag{42}$$

$$= \sum_{r=1}^{R} |\lambda_r| \|\mathcal{X}\|_F = \sum_{r=1}^{R} |\lambda_r| \tag{43}$$

□

C.3 Higher-order 2D-convolutional layer

Definition C.6. (Higher-order 2D-convolutional layer) The layer is parameterized by a $(2m+2)^{\text{th}}$-order tensor $\mathcal{M} \in \mathbb{R}^{k\times k\times T_1\times\cdots\times T_m\times S_1\times\cdots\times S_m}$. It maps an $(m+2)^{\text{th}}$-order input tensor $\mathcal{X} \in \mathbb{R}^{H\times W\times S_1\times\cdots\times S_m}$ to another $(m+2)^{\text{th}}$-order output tensor $\mathcal{Y} \in \mathbb{R}^{H\times W\times T_1\times\cdots\times S_m}$ as:

$$\mathcal{Y}_{:,:,t_1,\cdots,t_m} = \sum_{\forall l: s_l=1}^{S_l} \mathcal{M}_{:,:,t_1,\cdot,t_m,s_1,\cdots,s_m} * \mathcal{X}_{:,:,s_1,\cdots,s_m} \tag{44a}$$

$$\mathcal{Y}_{i,j,t_1,\cdots,t_m} = \sum_{\forall l: s_l=1}^{S_l} \sum_{p,q} \mathcal{M}_{i-p,j-q,t_1,\cdots,t_m,s_1,\cdots,s_m} \mathcal{X}_{p,q,s_1,\cdots,s_m} \tag{44b}$$

Definition C.7. (CP decomposition of higher-order 2D-convolutional layer) Given a higher-order 2D-convolutional layer in Eq. (38), a CP decomposition factorizes the weights tensor $\mathcal{M} \in \mathbb{R}^{H\times W\times T_1\times\cdots\times T_m\times S_1\times\cdots\times S_m}$ into $(m+1)$ core factors $\mathcal{C} \in \mathbb{R}^{R\times H\times W}$ and $\mathcal{K}^l \in \mathbb{R}^{R\times T_l\times S_l}, \forall l \in [m]$.

$$\mathcal{M}_{i,j,t_1,\cdots,t_m,s_1,\cdots,s_m} = \sum_{r=1}^{R} \lambda_r \mathcal{C}_{r,i,j} \mathcal{K}^1_{r,t_1,s_1} \cdots \mathcal{K}^m_{r,t_m,s_m} \tag{45}$$

where we we assume $\mathcal{C}_r$ and $\boldsymbol{K}^l_r = \mathcal{K}^l_{r,:,:}$ has unit Frobenius norm, i.e. $\|\boldsymbol{K}^l_r\|_F = 1$ and $\|\mathcal{C}_r\|_F = 1$

Lemma C.8. (Operator norm of Higher-order CP 2D-convolutional layer) For a higher-order 2D-convolutional layer layer whose weights tensor takes a CP format as in Eq. (45), the operator norm $\|\mathcal{M}\|$ is bound by the CP eigenvalues λ as

$$\|\mathcal{M}\| \leq \sqrt{HW} \sum_{r=1}^{R} |\lambda_r| \max_{f,g} \left|\tilde{\mathcal{C}}^{(f,g)}_r\right| \tag{46}$$

Proof. (of Lemma C.8) The proof is a combination of Lemmas C.2 and C.5. Let $\tilde{\mathcal{M}} = \mathcal{F}^{1,2}_m(\mathcal{M})$, we have

$$\|\mathcal{M}\| = \|\tilde{\mathcal{M}}\| = \sqrt{HW} \max_{f,g} \|\tilde{\mathcal{W}}^{(f,g)}\| \tag{47}$$

$$\tilde{\mathcal{M}}^{(f,g)} = \sum_{r=1}^{R} \lambda_r \tilde{\mathcal{C}}^{(f,g)}_r \mathcal{K}^1_{r,t_1,s_1} \cdots \mathcal{K}^m_{r,t_m,s_m} \tag{48}$$

The operator norm is bounded using Lemma C.5: $\|\tilde{\mathcal{M}}^{(f,g)}\| \leq \sum_{r=1}^{R} |\lambda_r| \max_{f,g} \left|\tilde{\mathcal{C}}^{(f,g)}_r\right|$. □

D Convolutional Neural Networks: Compressibility and Generalization

D.1 Complete Proofs of Convolutional Neural Networks

Definition D.1. [tensorization factor $t^{(k)}_j$] The *tensorization factors* $\left\{t^{(k)}_j\right\}_{j=1}^{R^{(k)}}$ of the k^{th} layer is defined as

$$t^{(k)}_j := \sum_{r=1}^{j} \left|\lambda^{(k)}_r\right| \max_{f,g} \left|\tilde{C}^{(f,g)}_r\right| \tag{49}$$

where $\lambda^{(k)}_r$ is the r^{th} largest value in the CP spectrum of $\mathcal{M}^{(k)}$.

Definition D.2. [tensor noise bound $\xi^{(k)}_j$] The *tensor noise bound* $\left\{\xi^{(k)}_j\right\}_{j=1}^{R^{(k)}}$ of the k^{th} layer measures the amplitudes of the remaining components after pruning the ones with amplitudes smaller than the $\lambda^{(k)}_j$:

$$\xi^{(k)}_j := \sum_{r=j+1}^{R^{(k)}} \left|\lambda^{(k)}_r\right| \max_{f,g} \left|\tilde{C}^{(f,g)}_r\right| \tag{50}$$

Definition D.3. [layer cushion $\zeta^{(k)}$] As introduced in Arora et al. (2018), the layer cushion of the k^{th} layer is defined to be the largest value $\zeta^{(k)}$ such that for any $\mathcal{X}^{(k)} \in S$,

$$\zeta^{(k)} \frac{\left\|\mathcal{M}^{(k)}\right\|_{\mathsf{F}}}{\sqrt{H^{(k)}W^{(k)}}} \left\|\mathcal{X}^{(k)}\right\|_{\mathsf{F}} \leq \left\|\mathcal{M}^{(k+1)}\right\|_{\mathsf{F}} \tag{51}$$

Following Arora et al. (2018), the layer cushion considers how much smaller the output $\left\|\mathcal{M}^{(k+1)}\right\|_{\mathsf{F}}$ of the k^{th} layer (after activation) compared with the product between the weight tensor $\left\|M^{(k)}\right\|_{\mathsf{F}}$ and the input $\left\|\mathcal{X}^{(k)}\right\|_{\mathsf{F}}$. Note that $H^{(k)}$ and $W^{(k)}$ are constants and will not influence the results of the theorem and the lemmas. For simplicity, we use H and W to denote the maximum $H^{(k)}$ and $W^{(k)}$ over the n layers for the following proofs where upper bounds are desired.

Given these definitions, we can bound the difference of outputs from a given model and its compressed counterpart. The following lemma characterizes the relation between the difference and the factors $t_j^{(k)}$, $\xi_j^{(k)}$, $\zeta^{(k)}$.

Lemma D.4. (Compression bound of convolutional neural networks) Suppose a convolutional neural network $\mathbb{M}$ has n layers, and each convolutional layer takes a CP format as in Eq. (31) with rank $R^{(k)}$. If an algorithm generates a compressed network $\hat{\mathbb{M}}$ such that only $\hat{R}^{(k)}$ components with largest $\lambda_r^{(k)}$'s are retained at the k^{th} layer, the difference of their outputs at the m^{th} is bounded by $\mathcal{X}^{(m+1)}$ as

$$\left\|\mathcal{X}^{(m)} - \hat{\mathcal{X}}^{(m)})\right\|_{\mathsf{F}} \leq \left(\sum_{k=1}^{m-1} \frac{\xi^{(k)}}{\zeta^{(k)} \left\|\mathcal{M}^{(k)}\right\|_{\mathsf{F}}} \prod_{l=k+1}^{m-1} \frac{t^{(l)}}{\zeta^{(l)} \left\|\mathcal{M}^{(l)}\right\|_{\mathsf{F}}}\right) \left\|\mathcal{X}^{(m)})\right\|_{\mathsf{F}} \tag{52}$$

Therefore for the whole network with n layers, the difference between $\mathbb{M}(\mathcal{X})$ and $\hat{\mathbb{M}}(\mathcal{X})$ is bounded by

$$\left\|\mathbb{M}(\mathcal{X}) - \hat{\mathbb{M}}(\mathcal{X})\right\|_{\mathsf{F}} \leq \left(\sum_{k=1}^{n} \frac{\xi^{(k)}}{\zeta^{(k)} \left\|\mathcal{M}^{(k)}\right\|_{\mathsf{F}}} \prod_{l=k+1}^{n} \frac{t^{(l)}}{\zeta^{(l)} \left\|\mathcal{M}^{(l)}\right\|_{\mathsf{F}}}\right) \left\|\mathbb{M}(\mathcal{X})\right\|_{\mathsf{F}} \tag{53}$$

Proof. (of Lemma D.4) We prove this lemma by induction. For $m = 2$, the lemma holds since

$$\left\|\mathcal{X}^{(2)} - \hat{\mathcal{X}}^{(2)}\right\|_{\mathsf{F}} = \left\|\mathsf{ReLU}\left(\mathcal{Y}^{(1)}\right) - \mathsf{ReLU}\left(\hat{\mathcal{Y}}^{(1)}\right)\right\|_{\mathsf{F}} \tag{54}$$

$$\leq \left\|\mathcal{Y}^{(1)} - \hat{\mathcal{Y}}^{(1)}\right\|_{\mathsf{F}} = \left\|\left(\mathcal{M}^{(1)} - \hat{\mathcal{M}}^{(1)}\right)\left(\mathcal{X}^{(1)}\right)\right\|_{\mathsf{F}} \tag{55}$$

$$\leq \sqrt{HW}\xi^{(1)} \left\|\mathcal{X}^{(1)}\right\|_{\mathsf{F}} \leq \frac{\xi^{(1)}}{\zeta^{(1)} \left\|\mathcal{M}^{(1)}\right\|_{\mathsf{F}}} \left\|\mathcal{M}^{(2)}\right\|_{\mathsf{F}} \tag{56}$$

where $\mathcal{Y} = \mathcal{M}(\mathcal{X})$ denotes the computation of a convolutional layer. (1) The first inequality follows the Lipschitzness of the ReLU activations; (2) The second inequality uses Lemma C.2; and (3) the last inequality holds by the definition of $\zeta^{(1)}$. For $m + 1 > 2$, we assume the lemma already holds for m

$$\left\|\mathcal{X}^{(m+1)} - \hat{\mathcal{X}}^{(m+1)}\right\|_{\mathsf{F}} = \left\|\mathsf{ReLU}\left(\mathcal{Y}^{(m)}\right) - \mathsf{ReLU}\left(\hat{\mathcal{Y}}^{(m)}\right)\right\|_{\mathsf{F}} \tag{57}$$

$$\leq \left\|\mathcal{Y}^{(m)} - \hat{\mathcal{Y}}^{(m)}\right\|_{\mathsf{F}} = \left\|\mathcal{M}^{(m)}\left(\mathcal{X}^{(m)}\right) - \hat{\mathcal{M}}^{(m)}\left(\hat{\mathcal{X}}^{(m)}\right)\right\|_{\mathsf{F}} \tag{58}$$

$$= \left\|\hat{\mathcal{M}^{(m)}}\left(\mathcal{X}^{(m)} - \hat{\mathcal{X}}^{(m)}\right) + \left(\mathcal{M}^{(m)} - \hat{\mathcal{M}}^{(m)}\right)\left(\mathcal{X}^{(m)}\right)\right\|_{\mathsf{F}} \tag{59}$$

$$\leq \sqrt{HW}\left(t^{(m)} \left\|\mathcal{X}^{(m)} - \hat{\mathcal{X}}^{(m)}\right\|_{\mathsf{F}} + \xi^{(m)} \left\|\mathcal{X}^{(m)}\right\|_{\mathsf{F}}\right) \tag{60}$$

$$\leq t^{(m)} \left(\sum_{k=1}^{m-1} \frac{\xi^{(k)}}{\zeta^{(k)} \left\|\mathcal{M}^{(k)}\right\|_{\mathsf{F}}} \prod_{l=k+1}^{m-1} \frac{t^{(l)}}{\zeta^{(l)} \left\|\mathcal{M}^{(l)}\right\|_{\mathsf{F}}}\right) \left\|\mathcal{X}^{(m)}\right\|_{\mathsf{F}} + \frac{\xi^{(m)}}{\zeta^{(m)} \left\|\mathcal{M}^{(m)}\right\|_{\mathsf{F}}} \left\|\mathcal{X}^{(m)}\right\|_{\mathsf{F}} \tag{61}$$

$$\leq \left(\sum_{k=1}^{m} \frac{\xi^{(k)}}{\zeta^{(k)} \left\|\mathcal{M}^{(k)}\right\|_{\mathsf{F}}} \prod_{l=k+1}^{m} \frac{t^{(l)}}{\zeta^{(l)} \left\|\mathcal{M}^{(l)}\right\|_{\mathsf{F}}}\right) \left\|\mathcal{M}^{(m+1)}\right\|_{\mathsf{F}} \tag{62}$$

which completes the induction. □

Lemma D.5. For any convolutional neural network $\mathbb{M}$ of n layers satisfying the assumptions in section 3 and any error $0 \leq \epsilon \leq 1$, Algorithm 1 generates a compressed tensorial neural network $\hat{\mathbb{M}}$ such that for any $\mathcal{X} \in S$:

$$\left\|\mathbb{M}(\mathcal{X}) - \hat{\mathbb{M}}(\mathcal{X})\right\|_{\mathsf{F}} \leq \epsilon \left\|\mathbb{M}(\mathcal{X})\right\|_{\mathsf{F}} \tag{63}$$

The compressed convolutional neural network $\hat{\mathbb{M}}$ has $\sum_{k=1}^{n} \hat{R}^{(k)}(s^{(k)} + o^{(k)} + k_x^{(k)} k_y^{(k)} + 1)$ total parameters, where each $\hat{R}^{(k)}$ satisfies:

$$\hat{R}^{(k)} = \min\left\{j \in [R^{(k)}] | \xi_j^{(k)} \Pi_{i=k+1}^{n} t_j^{(i)} \leq \frac{\epsilon}{n} \Pi_{i=k}^{n} \zeta^{(i)} \left\|\mathcal{M}^{(i)}\right\|_{\mathsf{F}}\right\} \tag{64}$$

Remark. Equation (64) is slightly different with equation 5, as the margin γ is replaced by a perturbation error ϵ. Therefore, how well the compressed tensorial neural network can approximate the original network is related to the choice of $\hat{R}^{(k)}$. Notice that when $\hat{R^{(k)}} = R^{(k)}$, the inequality for the k^{th} layer will be automatically satisfied as $\theta^{(k)} = 0$ in this case by definition.

Proof. (of Lemma D.5) The proof is trivial by observing

$$\frac{\xi^{(k)}}{\zeta^{(k)} \left\|\mathcal{M}^{(k)}\right\|_{\mathsf{F}}} \leq \frac{\epsilon}{n} \prod_{i=k+1}^{n} \frac{\zeta^{(i)} \left\|\mathcal{M}^{(i)}\right\|_{\mathsf{F}}}{t^{(i)}} \tag{65}$$

$$\implies \frac{\xi^{(k)}}{\zeta^{(k)} \left\|\mathcal{M}^{(k)}\right\|_{\mathsf{F}}} \prod_{i=k+1}^{n} \frac{t^{(i)}}{\zeta^{(i)} \left\|\mathcal{M}^{(i)}\right\|_{\mathsf{F}}} \leq \frac{\epsilon}{n} \tag{66}$$

$$\implies \sum_{k=1}^{n} \frac{\xi^{(k)}}{\zeta^{(k)} \left\|\mathcal{M}^{(k)}\right\|_{\mathsf{F}}} \prod_{i=k+1}^{n} \frac{t^{(i)}}{\zeta^{(l)} \left\|\mathcal{M}^{(i)}\right\|_{\mathsf{F}}} \leq \epsilon \tag{67}$$

□

Before proving Theorem 4.5, Lemma D.6 (introduced below) is needed.

Lemma D.6. For any convolutional neural network $\mathbb{M}$ of n layers satisfying the assumptions in section 3 and any margin $\gamma \geq 0$, $\mathbb{M}$ can be compressed to a tensorial convolutional neural network $\hat{\mathbb{M}}$ with $\sum_{k=1}^{n} \hat{R}^{(k)}(s^{(k)} + t^{(k)} + k_x^{(k)} \times k_y^{(k)} + 1)$ total parameters such that for any $\mathcal{X} \in S$, $\hat{L_0}(\hat{\mathbb{M}}) \leq \hat{L}_\gamma(\mathbb{M})$. Here, for each layer k,

$$\hat{R}^{(k)} = \min\left\{j \in [R^{(k)}] | \xi_j^{(k)} \Pi_{i=k+1}^{n} t_j^{(i)} \leq \frac{\epsilon}{n} \Pi_{i=k}^{n} \zeta^{(i)} \left\|\mathcal{M}^{(i)}\right\|_{\mathsf{F}}\right\} \tag{68}$$

Proof. (of Lemma D.6)

If $\gamma \geq 2 \max_{\mathcal{X} \in S} \left\|\mathbb{M}(\mathcal{X})\right\|_{\mathsf{F}}$, for any pair $(\mathcal{X}, y) \in S$, we have

$$\begin{aligned} |\mathbb{M}(\mathcal{X})[y] - \max_{j \neq y} \mathbb{M}(\mathcal{X})[j]|^2 &\leq (|\mathbb{M}(\mathcal{X})[y]| + |\max_{j \neq y} \mathbb{M}(\mathcal{X})[j]|)^2 \\ &\leq 4 \max_{\mathcal{X} \in S} \left\|\mathbb{M}(\mathcal{X})\right\|_{\mathsf{F}}^2 \\ &\leq \gamma^2 \end{aligned}$$

Then the output margin of $\mathbb{M}$ cannot be greater than γ for any $\mathcal{X} \in S$. Thus $\hat{L}_\gamma(\mathbb{M}) = 1$.

If $\gamma < 2 \max_{\mathcal{X} \in S} \left\|\mathbb{M}(\mathcal{X})\right\|_{\mathsf{F}}$, setting

$$\epsilon = \frac{\gamma}{2 \max_{\mathcal{X} \in S} \left\|\mathbb{M}(\mathcal{X})\right\|_{\mathsf{F}}}$$

in Lemma D.5, we obtain a compressed fully-connected tensorial neural network $\hat{\mathbb{M}}$ with the desired number of parameters and

$$\left\|\mathbb{M}(\mathcal{X}) - \hat{\mathbb{M}}(\mathcal{X})\right\|_{\mathsf{F}} < \frac{\gamma}{2} \Rightarrow \forall j, |\mathbb{M}(\mathcal{X})[j] - \hat{\mathbb{M}}(\mathcal{X})[j]| < \frac{\gamma}{2}$$

Then for any pair $(\mathcal{X}, y) \in S$, if $\mathbb{M}(\mathcal{X})[y] > \gamma + \max_{j \neq y} \mathbb{M}(\mathcal{X})[j]$, $\hat{\mathbb{M}}$ classifies $\mathcal{X}$ correctly as well because:

$$\hat{\mathbb{M}}(\mathcal{X})[y] > \mathbb{M}(\mathcal{X})[y] - \frac{\gamma}{2} > \max_{j \neq y} \mathbb{M}(\mathcal{X})[j] + \frac{\gamma}{2} > \max_{j \neq y} \hat{\mathbb{M}}(\mathcal{X})[j]$$

Thus, $\hat{L_0}(\hat{\mathbb{M}}) \leq \hat{L}_\gamma(\mathbb{M})$. □

Now we prove the main theorem 4.5 by bounding the covering number given any ϵ.

D.1.1 Covering Number Analysis for Convolutional Neural Network

Proof. (of Theorem 4.5) To be more specific, let us bound the covering number of the compressed network $\hat{\mathbb{M}}$ by approximating each parameter with accuracy μ.

Lemma D.7. For any given constant accuracy μ, the covering number of the compressed convolutional network $\hat{\mathbb{M}}$ is of order $\tilde{O}(d)$ where d denotes the total number of parameters in $\hat{\mathbb{M}}$: $d := \sum_{k=1}^{n} \hat{R}^{(k)}(s^{(k)} + o^{(k)} + k_x^{(k)} \times k_y^{(k)} + 1)$.

Let $\tilde{\mathcal{M}}$ denote the network after approximating each parameter in $\hat{\mathbb{M}}$ with accuracy μ (and $\tilde{\mathcal{M}}^{(k)}$ denote its weight tensor on the k^{th} layer). Based on the given accuracy, we know that $\forall k,\ |\hat{\lambda}_r^{(k)} - \tilde{\lambda}_r^{(k)}| \leq \mu$, $\left\|\hat{\boldsymbol{a}}_r^{(k)} - \tilde{\boldsymbol{a}}_r^{(k)}\right\| \leq \sqrt{s^{(k)}}\mu$, $\left\|\hat{\boldsymbol{b}}_r^{(k)} - \tilde{\boldsymbol{b}}_r^{(k)}\right\| \leq \sqrt{o^{(k)}}\mu$, $\left\|\hat{\boldsymbol{C}}_r^{(k)} - \tilde{\boldsymbol{C}}_r^{(k)}\right\| \leq \sqrt{k_x^{(k)} k_y^{(k)}}\mu$, where s, o, k_x and k_y are the number of input channels, the number of output channels, the height of the kernel and the width of the kernel, as defined in Section 3. For simplicity, in this proof, let us just use $\mathcal{X}^{(k)}, \mathcal{Y}^{(k)}, \boldsymbol{a}_r^{(k)}, \boldsymbol{b}_r^{(k)}, \boldsymbol{C}_r^{(k)}$ to denote $\hat{\mathcal{X}}^{(k)}, \hat{\mathcal{Y}}^{(k)}, \hat{\boldsymbol{a}}_r^{(k)}, \hat{\boldsymbol{b}}_r^{(k)}, \hat{\boldsymbol{C}}_r^{(k)}$. $\mathcal{X}^{(k)} \in \mathbb{R}^{H \times W \times s^{(k)}}$, $\mathcal{Y}^{(k)} \in \mathbb{R}^{H \times W \times o^{(k)}}$.

We have

$$\mathcal{F}_3^{1,2}(\mathcal{Y}^{(k)})_{fgj} = \sqrt{HW} \sum_i [\mathcal{F}_3^{1,2}(\mathcal{X}^{(k)})_{fgi} \sum_{r=1}^{\hat{R}^{(k)}} \lambda_r^{(k)} a_{ri}^{(k)} b_{rj}^{(k)} \mathcal{F}_2(C_r^{(k)})_{fg}]$$

$$\mathcal{F}_3^{1,2}(\tilde{\mathcal{Y}}^{(k)})_{fgj} = \sqrt{HW} \sum_i [\mathcal{F}_3^{1,2}(\tilde{\mathcal{X}}^{(k)})_{fgi} \sum_{r=1}^{\hat{R}^{(k)}} \tilde{\lambda}_r^{(k)} \tilde{a}_{ri}^{(k)} \tilde{b}_{rj}^{(k)} \mathcal{F}_2(\tilde{C}_r^{(k)})_{fg}]$$

where $\sqrt{HW}$ is a normalization factor defined in Lemma B.7

Let $\epsilon^{(k)} = \left\|\tilde{\boldsymbol{Y}}^{(k)} - \boldsymbol{Y}^{(k)}\right\|_{\mathsf{F}}$. Then for each k, let $\varphi = \sum_{f,g,i,j} \left(\sum_r^{\hat{R}^{(k)}} \lambda_r^{(k)} a_{ri}^{(k)} b_{rj}^{(k)} \left(\mathcal{F}_2(C_r^{(k)})_{fg} - \mathcal{F}_2(\tilde{C}_r^{(k)})_{fg}\right) \right)^2$ and $\psi = \sum_{f,g,i,j} \left(\sum_r^{\hat{R}^{(k)}} \left(\lambda_r^{(k)} a_{ri}^{(k)} b_{rj}^{(k)} - \tilde{\lambda}_r^{(k)} \tilde{a}_{ri}^{(k)} \tilde{b}_{rj}^{(k)}\right) \mathcal{F}_2(\tilde{C}_r^{(k)})_{fg} \right)^2$. We first bound φ and ψ as follows.

Bound $\varphi = \sum_{f,g,i,j} \left(\sum_r^{\hat{R}^{(k)}} \lambda_r^{(k)} a_{ri}^{(k)} b_{rj}^{(k)} \left(\mathcal{F}_2(C_r^{(k)})_{fg} - \mathcal{F}_2(\tilde{C}_r^{(k)})_{fg}\right) \right)^2$: All calculations are based on the k^{th} layer, we remove the layer number (k) for ease of reading. So $a = a^{(k)}$ (the same for b, c, and R). Then

$$\begin{aligned}
&\sum_{f,g,i,j} \Big(\sum_r^{\hat{R}^{(k)}} \lambda_r^{(k)} a_{ri}^{(k)} b_{rj}^{(k)} \left(\mathcal{F}_2(C_r^{(k)})_{fg} - \mathcal{F}_2(\tilde{C}_r^{(k)})_{fg}\right) \Big)^2 \\
&\leq \sum_{f,g,i,j} \Big(\sum_r^{\hat{R}} (\lambda_r a_{ri} b_{rj})^2 \sum_r^{\hat{R}} \left(\mathcal{F}_2(C_r)_{fg} - \mathcal{F}_2(\tilde{C}_r)_{fg}\right)^2 \Big) \\
&\leq \sum_r^{\hat{R}} (\lambda_r^2 \sum_i a_{ri}^2 \sum_j b_{rj}^2) \sum_r^{\hat{R}} \sum_{f,g} \left(\mathcal{F}_2(C_r)_{fg} - \mathcal{F}_2(\tilde{C}_r)_{fg}\right)^2 \\
&\leq \sum_r^{\hat{R}} \lambda_r^2 \hat{R} k_x k_y \mu^2
\end{aligned}$$

Bound $\psi = \sum_{f,g,i,j} \left(\sum_r^{\hat{R}^{(k)}} \left(\lambda_r^{(k)} a_{ri}^{(k)} b_{rj}^{(k)} - \tilde{\lambda}_r^{(k)} \tilde{a}_{ri}^{(k)} \tilde{b}_{rj}^{(k)}\right) \mathcal{F}_2(\tilde{C}_r^{(k)})_{fg} \right)^2$: Similarly, we remove the layer number (k) for ease of reading. Then we have

$$
\begin{aligned}
&\sum_{f,g,i,j} \Big(\sum_{r}^{\hat{R}^{(k)}} (\lambda_r^{(k)} a_{ri}^{(k)} b_{rj}^{(k)} - \tilde{\lambda}_r^{(k)} \tilde{a}_{ri}^{(k)} \tilde{b}_{rj}^{(k)}) \mathcal{F}_2(\tilde{C}_r^{(k)})_{fg} \Big)^2 \\
&\quad \leq \sum_{f,g,i,j} (\sum_{r}^{\hat{R}} (\lambda_r a_{ri} b_{rj} - \tilde{\lambda}_r \tilde{a}_{ri} \tilde{b}_{rj})^2 \sum_{r}^{\hat{R}} \mathcal{F}_2(\tilde{C}_r)_{fg}^2) \\
&\quad = \sum_{f,g,i,j} \Big(\sum_{r}^{\hat{R}} \big(\lambda_r (a_{ri} b_{rj} - \tilde{a}_{ri} \tilde{b}_{rj}) + (\lambda_r - \tilde{\lambda}_r) \tilde{a}_{ri} \tilde{b}_{rj}\big)^2 \sum_{r}^{\hat{R}} \mathcal{F}_2(\tilde{C}_r)_{fg}^2 \Big) \\
&\quad \leq \sum_{f,g,i,j} \Big((2 \sum_{r}^{\hat{R}} \lambda_r^2 (a_{ri} b_{rj} - \tilde{a}_{ri} \tilde{b}_{rj})^2 + 2 \sum_{r}^{\hat{R}} (\lambda_r - \tilde{\lambda}_r)^2 \tilde{a}_{ri}^2 \tilde{b}_{rj}^2) \sum_{r}^{\hat{R}} \mathcal{F}_2(\tilde{C}_r)_{fg}^2 \Big) \\
&\quad = \sum_{f,g,i,j} \left(\left(2 \sum_{r}^{\hat{R}} \lambda_r^2 \big(a_{ri}(b_{rj} - \tilde{b}_{rj}) + (a_{ri} - \tilde{a}_{ri}) \tilde{b}_{rj}\big)^2 + 2 \sum_{r}^{\hat{R}} (\lambda_r - \tilde{\lambda}_r)^2 \tilde{a}_{ri}^2 \tilde{b}_{rj}^2 \right) \sum_{r}^{\hat{R}} \mathcal{F}_2(\tilde{C}_r)_{fg}^2 \right) \\
&\quad \leq \sum_{f,g,i,j} \left(\left(4 \sum_{r}^{\hat{R}} \lambda_r^2 \big(a_{ri}^2 (b_{rj} - \tilde{b}_{rj})^2 + (a_{ri} - \tilde{a}_{ri})^2 \tilde{b}_{rj}^2\big)^2 + 2 \sum_{r}^{\hat{R}} (\lambda_r - \tilde{\lambda}_r)^2 \tilde{a}_{ri}^2 \tilde{b}_{rj}^2 \right) \sum_{r}^{\hat{R}} \mathcal{F}_2(\tilde{C}_r)_{fg}^2 \right) \\
&\quad = \left(4 \sum_{r}^{\hat{R}} \lambda_r^2 \big(\sum_{i} a_{ri}^2 \sum_{j} (b_{rj} - \tilde{b}_{rj})^2 + \sum_{i} (a_{ri} - \tilde{a}_{ri})^2 \sum_{j} \tilde{b}_{rj}^2 \big)^2 + 2 \sum_{r}^{\hat{R}} (\lambda_r - \tilde{\lambda}_r)^2 \sum_{i} \tilde{a}_{ri}^2 \sum_{j} \tilde{b}_{rj}^2 \right) \sum_{r}^{\hat{R}} \sum_{f.g} \mathcal{F}_2(\tilde{C}_r)_{fg}^2 \\
&\quad \leq \big(4 \sum_{r}^{\hat{R}} \lambda_r^2 (o\mu^2 + s\mu^2) + 2\hat{R}\mu^2\big) \hat{R} \\
&\quad = \big(4 \sum_{r}^{\hat{R}} \lambda_r^2 (o + s) \hat{R} + 2\hat{R}^2\big) \mu^2
\end{aligned}
$$

Bound $\epsilon^{(k)} = \left\| \tilde{\mathcal{Y}}^{(k)} - \mathcal{Y}^{(k)} \right\|_{\mathbf{F}}$: Similarly, we remove the layer number (k). And we let $w_i = \mathcal{F}_3^{1,2}(\mathcal{X}^{(k)})_{fgi}$, $\tilde{w}_i = \mathcal{F}_3^{1,2}(\tilde{\mathcal{X}}^{(k)})_{fgi}$, $u_i = \sum_r^{\hat{R}} \lambda_r^{(k)} a_{ri}^{(k)} b_{rj}^{(k)} \mathcal{F}_2(C_r^{(k)})_{fg}$ and $\tilde{u}_i = \sum_r^{\hat{R}} \tilde{\lambda}_r^{(k)} \tilde{a}_{ri}^{(k)} \tilde{b}_{rj}^{(k)} \mathcal{F}_2(\tilde{C}_r^{(k)})_{fg}$.

$$
\begin{aligned}
&\left\|\tilde{\mathcal{Y}}^{(k)}-\mathcal{Y}^{(k)}\right\|_{\mathsf{F}}^2 \\
&\quad=\left\|\mathcal{F}_3^{1,2}(\tilde{\mathcal{Y}}^{(k)})-\mathcal{F}_3^{1,2}(\mathcal{Y}^{(k)})\right\|_{\mathsf{F}}^2 \\
&\quad=\sum_{f,g,j}\left|[\mathcal{F}_3^{1,2}(\tilde{\mathcal{Y}}^{(k)})]_{fgj}-[\mathcal{F}_3^{1,2}(\mathcal{Y}^{(k)})]_{fgj}\right|^2 \\
&\quad=\sum_{f,g,j}HW(\sum_i w_iu_i-\sum_i\tilde{w}_i\tilde{u}_i)^2 \\
&\quad=HW\sum_{f,g,j}\big(\sum_i w_i(u_i-\tilde{u}_i)+\sum_i(w_i-\tilde{w}_i)\tilde{u}_i\big)^2 \\
&\quad\le 2HW\sum_{f,g,j}\big(\sum_i w_i(u_i-\tilde{u}_i)\big)^2+2\sum_{f,g,j}\big(\sum_i(w_i-\tilde{w}_i)\tilde{u}_i\big)^2 \\
&\quad\le 2HW\sum_{f,g,j}\big((\sum_i w_i^2)\sum_i(u_i-\tilde{u}_i)^2\big)+2\sum_{f,g,j}\big(\sum_i(w_i-\tilde{w}_i)^2(\sum_i\tilde{u}_i)^2\big) \\
&\quad\le 2HW(\sum_{f,g,i}w_i^2)\sum_{f,g,i,j}(u_i-\tilde{u}_i)^2+2\sum_{f,g,i}(w_i-\tilde{w}_i)^2\sum_{f,g,i,j}\tilde{u}_i^2 \\
&\quad\le 2HW(\sum_{f,g,i}w_i^2)(2\varphi+2\psi)+2\sum_{f,g,i}(w_i-\tilde{w}_i)^2\sum_{f,g,i,j}\tilde{u}_i^2 \\
&\quad\le 4HW\left\|\mathcal{X}^{(k)}\right\|_{\mathsf{F}}^2\mu^2\big(\sum_r^{\hat{R}}\lambda_r^2\hat{R}k_xk_y+4\sum_r^{\hat{R}}\lambda_r^2(o+s)\hat{R}+2\hat{R}^2\big)+2\sum_{f,g,i}(w_i-\tilde{w}_i)^2\sum_{f,g,i,j}\tilde{u}_i^2 \\
&\quad\le 4HW\left\|\mathcal{X}^{(k)}\right\|_{\mathsf{F}}^2\mu^2\big(\sum_r^{\hat{R}}\lambda_r^2\hat{R}k_xk_y+4\sum_r^{\hat{R}}\lambda_r^2(o+s)\hat{R}+2\hat{R}^2\big)+2(\left\|\mathcal{X}^{(k)}-\tilde{\mathcal{X}}^{(k)}\right\|_{\mathsf{F}}^2\left\|\tilde{\mathcal{M}}\right\|_{\mathsf{F}}^2)
\end{aligned}
$$

When $k=1$, we know that $\mathcal{X}^{(1)}=\tilde{\mathcal{X}}^{(1)}$, so

$$
\begin{aligned}
&\left\|\tilde{\mathcal{Y}}^{(1)}-\mathcal{Y}^{(1)}\right\|_{\mathsf{F}}^2 \\
&\qquad\le 4HW\left\|\mathcal{X}^{(1)}\right\|_{\mathsf{F}}^2\mu^2\big(\sum_r^{\hat{R}}\lambda_r^2\hat{R}k_xk_y+4\sum_r^{\hat{R}}\lambda_r^2(o+s)\hat{R}+2\hat{R}^2\big)
\end{aligned}
$$

When $k>1$, we have

$$
\begin{aligned}
&\left\|\tilde{\mathcal{Y}}^{(k)}-\mathcal{Y}^{(k)}\right\|_{\mathsf{F}}^2 \\
&\quad\le 4HW\left\|\mathcal{X}^{(k)}\right\|_{\mathsf{F}}^2\mu^2\big(\sum_r^{\hat{R}}\lambda_r^2\hat{R}k_xk_y+4\sum_r^{\hat{R}}\lambda_r^2(o+s)\hat{R}+2\hat{R}^2\big)+2(\left\|\mathcal{X}^{(k)}-\tilde{\mathcal{X}}^{(k)}\right\|_{\mathsf{F}}^2\left\|\tilde{\mathcal{M}}\right\|_{\mathsf{F}}^2) \\
&\quad\le 4HW\left\|\mathcal{X}^{(k)}\right\|_{\mathsf{F}}^2\mu^2\big(\sum_r^{\hat{R}}\lambda_r^2\hat{R}k_xk_y+4\sum_r^{\hat{R}}\lambda_r^2(o+s)\hat{R}+2\hat{R}^2\big)+2(\left\|\mathsf{ReLU}\left(\mathcal{Y}^{(k-1)}\right)-\mathsf{ReLU}\left(\tilde{\mathcal{Y}}^{(k-1)}\right)\right\|_{\mathsf{F}}^2\left\|\tilde{\mathcal{M}}\right\|_{\mathsf{F}}^2) \\
&\quad\le 4HW\left\|\mathcal{X}^{(k)}\right\|_{\mathsf{F}}^2\mu^2\big(\sum_r^{\hat{R}}\lambda_r^2\hat{R}k_xk_y+4\sum_r^{\hat{R}}\lambda_r^2(o+s)\hat{R}+2\hat{R}^2\big)+2(\left\|\mathcal{Y}^{(k-1)}-\tilde{\mathcal{Y}}^{(k-1)}\right\|_{\mathsf{F}}^2\left\|\tilde{\mathcal{M}}\right\|_{\mathsf{F}}^2) \\
&\quad\le 4HW\left\|\mathcal{X}^{(k)}\right\|_{\mathsf{F}}^2\mu^2\big(\sum_r^{\hat{R}}\lambda_r^2\hat{R}k_xk_y+4\sum_r^{\hat{R}}\lambda_r^2(o+s)\hat{R}+2\hat{R}^2\big)+2((\epsilon^{(k-1)})^2\left\|\tilde{\mathcal{M}}\right\|_{\mathsf{F}}^2)
\end{aligned}
$$

Let $\alpha^{(k)} = 4HW \left\|\mathcal{X}^{(k)}\right\|_{\mathsf{F}}^2 \big(\sum_r^{\hat{R}^{(k)}} (\lambda_r^{(k)})^2 \hat{R}^{(k)} k_x^{(k)} k_y^{(k)} + 4 \sum_r^{\hat{R}^{(k)}} (\lambda_r^{(k)})^2 (o^{(k)} + s^{(k)}) \hat{R}^{(k)} + 2(\hat{R}^{(k)})^2 \big) \mu^2$, and $\beta^{(k)} = 2 \left\|\tilde{\mathcal{M}}^{(k)}\right\|_{\mathsf{F}}^2$. Then the difference between the final output of the two networks are bounded by:

$$\begin{aligned}
&\left\|\hat{\mathbb{M}}(\mathcal{X}) - \tilde{\mathbb{M}}(\mathcal{X})\right\|_{\mathsf{F}}^2 \\
&\quad = \left\|\mathsf{ReLU}\left((\hat{\mathcal{Y}})\right) - \mathsf{ReLU}\left((\tilde{\mathcal{Y}})\right)\right\|_{\mathsf{F}}^2 \\
&\quad \leq \left\|\hat{\mathcal{Y}} - \tilde{\mathcal{Y}}\right\|_{\mathsf{F}}^2 \\
&\quad \leq \sum_{k=1}^{n} \alpha^{(k)} \prod_{i=k+1}^{n} \beta^{(i)}
\end{aligned}$$

Since $\forall k \in [n], \left\|\mathcal{X}^{(k)}\right\| \leq \Pi_{i=k}^{n} \frac{\left\|\mathcal{X}^{(n+1)}\right\|_{\mathsf{F}}}{\zeta^{(i)} \left\|\mathcal{M}^{(i)}\right\|_{\mathsf{F}}}$, to obtain an ϵ-cover of the compressed network, we can first assume $\beta^{(k)} \geq 1 \ \forall k \in [n]$. Then μ need to satisfy:

$$\mu \leq \frac{\epsilon}{2\sqrt{HW} n \left\|\mathcal{X}^{(n+1)}\right\|_{\mathsf{F}} \hat{R}^{(*)} (\frac{\sqrt{2}\left\|\tilde{\mathcal{M}}^{(*)}\right\|_{\mathsf{F}}}{\zeta^{(*)} \left\|\mathcal{M}^{(*)}\right\|_{\mathsf{F}}})^n \sqrt{(\lambda^{(*)})^2 k_x^{(*)} k_y^{(*)} + 4(\lambda^{(*)})^2 (o^{(*)} + s^{(*)}) + 2}}$$

where $\hat{R}^{(*)} = \max_k r^{(k)}$ $\lambda^{(*)} = \max_{r,k} \lambda_r^{(k)}$, $s^{(*)} = \max_k s^{(k)}$, $o^{(*)} = \max_k o^{(k)}$, $k_x^{(*)} = \max_k k_x^{(k)}$, $k_y^{(*)} = \max_k k_y^{(k)}$ and $\frac{\left\|\tilde{\mathcal{M}}^{(*)}\right\|_{\mathsf{F}}}{\mu^{(*)} \left\|\mathcal{M}^{(*)}\right\|_{\mathsf{F}}} = \max_k \frac{\left\|\tilde{\mathcal{M}}^{(k)}\right\|_{\mathsf{F}}}{\mu^{(k)} \left\|\mathcal{M}^{(*)}\right\|_{\mathsf{F}}}$

As when μ is fixed, the number of networks in our cover will at most be $(\frac{1}{\mu})^d$ where d denote the number of parameters in the compressed network. Hence, the covering number w.r.t to a given ϵ is $\tilde{O}(nd)$ (n is the number of layers in the given neural network). As for practical neural networks, the number of layers n is usually much less than $O(\log(d))$, thus the covering number we obtained w.r.t to a given ϵ is just $\tilde{O}(d)$ for practical neural networks.

□

E Fully Connected Networks: Compressibility and Generalization

In this section, we derive generalization bounds for fully connected (FC) neural networks (denoted as $\mathbb{M}$) using tensor methods.

E.1 Compression of a FC Network with CPL

Original Fully Connected Neural Network: Let $\mathbb{M}$ denote an n-layer fully connected network with ReLU activations, where $\boldsymbol{A}^{(k)} \in \mathbb{R}^{h^{(k)} \times h^{(k+1)}}$ denotes the weight matrix of the k^{th} layer, $\boldsymbol{x}^{(k)} \in \mathbb{R}^{h^{(k)}}$ denotes the input to k^{th} layer, and $\boldsymbol{y}^{(k)}$ denotes the output of the k^{th} layer before activation in $\mathbb{M}$. **Transform original FCN to a CP-FCN:** We transform the original fully connected network $\mathbb{M}$ to a network $\tilde{\mathbb{M}}$ with CPL. The k^{th} layer of $\tilde{\mathbb{M}}$ is denoted by $\mathcal{M}^{(k)} \in \mathbb{R}^{s_1^{(k)} \times s_2^{(k)} \times s_1^{(k+1)} \times s_2^{(k+1)}}$ is a 4-dimensional tensor reshaped from $\boldsymbol{A}^{(k)}$ where $s_1^{(k)} \times s_2^{(k)} = h_k, \forall k \in [n]$.

Input and Output of $\tilde{\mathbb{M}}$: The original input and output vectors of $\mathbb{M}$ are reshaped into matrices. The input to the k^{th} layer of the $\tilde{\mathbb{M}}$, denoted by $\boldsymbol{X}^{(k)} \in \mathbb{R}^{s_1^{(k)} \times s_2^{(k)}}$, is a matrix reshaped from the input vector $\boldsymbol{x}^{(k)}$ of the k^{th} layer in the original network $\mathbb{M}$. Similarly, the output of the k^{th} layer before activation in $\tilde{\mathbb{M}}$, denoted by $\boldsymbol{Y}^{(k)} \in \mathbb{R}^{s_1^{(k)} \times s_2^{(k)}}$, is a matrix reshaped from the output vector $\boldsymbol{y}^{(k)}$ of the k^{th} layer in the original network $\mathbb{M}$. For prediction purposes, we reshape the output $\boldsymbol{Y}^{(n)}$ of the last layer in $\tilde{\mathbb{M}}$ back into a vector. So the final outputs of $\mathbb{M}$ and $\tilde{\mathbb{M}}$ are of the same dimension.

Assumption E.1 (**Polyadic Form of $\mathbb{M}$**). For each layer k, assume the weight tensor $\mathcal{M}^{(k)}$ of $\mathbb{M}$ has a Polyadic form with rank $R^{(k)} \leq \min\{s_1^{(k)}, s_2^{(k)}, s_1^{(k+1)}, s_2^{(k+1)}\}$:

$$\mathcal{M}^{(k)} = \sum_{i=1}^{R^{(k)}} \lambda_i^{(k)} a_i^{(k)} \otimes b_i^{(k)} \otimes c_i^{(k)} \otimes d_i^{(k)} \tag{69}$$

where $\forall i, a_i, b_i, c_i, d_i$ are unit vectors in $\mathbb{R}^{s_1^{(k)}}, \mathbb{R}^{s_2^{(k)}}, \mathbb{R}^{s_1^{(k+1)}}, \mathbb{R}^{s_2^{(k+1)}}$ respectively, and $\forall 1 \leq i \leq R^{(k)}, \langle a_i, a_i \rangle = 1, \langle b_i, b_i \rangle = 1, \langle c_i, c_i \rangle = 1, \langle d_i, d_i \rangle = 1$. Moreover, for each $\mathcal{M}^{(k)}$, $\lambda_i^{(k)} \geq \lambda_{i+1}^{(k)}, \forall i$, and the absolute value of the smallest $|\lambda_{R^{(k)}}^{(k)}|$ can be arbitrarily small.

The total number of parameters in $\mathbb{M}$ is $(s_1^{(k)} + s_2^{(k)} + s_1^{(k+1)} + s_2^{(k+1)} + 1)R^{(k)}$ and a smaller $R^{(k)}$ renders fewer number of parameters and thus leads to compression. We introduce a compression mechanism that prunes out the smaller components of weight tensor of $\mathbb{M}$, i.e., a low rank approximation of each weight tensor $\mathcal{M}^{(k)}$ of the k^{th} layer, and generates a compressed CP-FCN $\hat{\mathbb{M}}$. The algorithm is depicted in Algorithm 2.

Compression of a FC Network with CPL: In Li and Huang (2018), a tensor decomposition algorithm (procedure 1 in Li and Huang (2018)) on tensors with asymmetric orthogonal components is guaranteed to recover the top-r components with the largest singular values. To compress $\mathbb{M}$, we apply top-$\hat{R}^{(k)}$ ($\hat{R}^{(k)} \leq R^{(k)}$) CP decomposition algorithm on each $\mathcal{M}^{(k)}$, obtaining the components from CP decomposition $(\hat{\lambda}_i^{(k)}, \hat{a}_i^{(k)}, \hat{b}_i^{(k)}, \hat{c}_i^{(k)}, \hat{d}_i^{(k)})$, $i \in [\hat{R}^{(k)}]$. Therefore, we achieve **a compressed network** $\hat{\mathbb{M}}$ of $\mathbb{M}$, and the j^{th} layer of the compressed network $\hat{\mathbb{M}}$ has weight tensor as follows

$$\hat{\mathcal{T}}^{(k)} = \sum_{i=1}^{\hat{R}^{(k)}} \hat{\lambda}_i^{(k)} \hat{a}_i^{(k)} \otimes \hat{b}_i^{(k)} \otimes \hat{c}_i^{(k)} \otimes \hat{d}_i^{(k)}. \tag{70}$$

As each $\mathcal{M}^{(k)}$ has a low rank orthogonal CP decomposition by our assumption, the returned results $\{\hat{\lambda}_i^{(k)}, \hat{a}_i^{(k)}, \hat{b}_i^{(k)}, \hat{c}_i^{(k)}, \hat{d}_i^{(k)}\}_{i=1}^{\hat{R}^{(k)}}$ from procedure 1 in Li and Huang (2018) are perfect recoveries of $\{\lambda_i^{(k)}, a_i^{(k)}, b_i^{(k)}, c_i^{(k)}, d_i^{(k)}\}_{i=1}^{\hat{R}^{(k)}}$ according to the robustness theorem in Li and Huang (2018). Our compression procedure is depicted in Algorithm 2.

Algorithm 2 Compression of Fully Connected Neural Networks

$^{\square}$FBR (in Appendix G) denotes a sub-procedure which calculates $\hat{R}^{(k)}$ such that $\left\|\mathbb{M}(\boldsymbol{X}) - \hat{\mathbb{M}}(\boldsymbol{X})\right\|_{\mathsf{F}} \leq \epsilon \left\|\mathbb{M}(\boldsymbol{X})\right\|_{\mathsf{F}}$ holds for any input $\boldsymbol{X}$ in the training dataset and for any given ϵ.

$^{\triangle}$TNN-Project (in Appendix G) denotes a sub-procedure which returns a compressed network $\hat{\mathbb{M}}$ by pruning out the smaller components in the Polyadic form of the weight tensors in the original CNN.

More intuitions of the sub-procedures FBR and TNN-Project are described in Section E.2.

Input: A FCN $\mathbb{M}$ of n layers and a margin γ

Output: A compressed $\hat{\mathbb{M}}$ whose expected error $L_0(\hat{\mathbb{M}}) \leq \hat{L}_\gamma(\mathbb{M}) + \tilde{O}\left(\sqrt{\frac{\sum_{k=1}^n \hat{R}^{(k)}(2s^{(k)}+2s^{(k+1)}+1)}{m}}\right)$

1: Calculate all layer cushions $\{\zeta^{(k)}\}_{k=1}^n$ based on definition E.4

2: Pick $R^{(k)} = \min\{s^{(k)}, s^{(k+1)}\}$ for each layer k

3: If $\mathbb{M}$ does not have CPL, apply a CP-decomposition to the weight tensor of each layer k

4: Set the perturbation parameter $\epsilon := \frac{\gamma}{2\max_{\boldsymbol{X}} \|\mathbb{M}(\boldsymbol{X})\|_{\mathsf{F}}}$

5: Compute number of components needed for each layer of the compressed network $\{\hat{R}^{(k)}\}_{k=1}^n \leftarrow \mathsf{FBR}^{\square}\left(\{\mathcal{M}^{(k)}\}_{k=1}^n, \{R^{(k)}\}_{k=1}^n, \{\zeta^{(k)}\}_{k=1}^n, \epsilon\right)$

6: $\hat{\mathbb{M}} \leftarrow \mathsf{TNN\text{-}Project}^{\triangle}\left(\mathbb{M}, \{\hat{R}^{(k)}\}_{i=1}^n\right)$

7: Return the compressed convolutional neural network $\hat{\mathbb{M}}$

We denote the input matrix of the k^{th} layer in $\hat{\mathbb{M}}$ as $\hat{\boldsymbol{X}}^{(k)}$, and the output matrix before activation as $\hat{\boldsymbol{Y}}^{(k)}$. Note that $\boldsymbol{X}^{(1)} = \hat{\boldsymbol{X}}^{(1)}$ as the input data is not being modified.

Algorithm 2 is desigend for general neural networks. For neural networks with *CPLayer*, line 3 can be done by pruning out small components from CP decomposition, and only keeping top-$\hat{R}^{(k)}$ components. For notation simplicity, assume for each layer in $\mathbb{M}$, the width of the k^{th} layer is a square of some integer $s^{(k)}$. Then the input to the k^{th} layer of $\mathbb{M}$ is a ReLu transformation of the output of the $k-1^{\text{th}}$ layer as in equation (71). The

output of the k^{th} layer of $\mathbb{M}$ is illustrated in equation (72) as the weight tensor which permits a CP forms as in equation (69).

$$\boldsymbol{X}^{(k)} = \mathsf{ReLU}\left(\boldsymbol{Y}^{(k-1)}\right) \tag{71}$$

$$\boldsymbol{Y}^{(k)} = \sum_{i=1}^{\hat{R}^{(k)}} \lambda_i^{(k)} {a_i^{(k)}}^\top \boldsymbol{X}^{(k)} b_i^{(k)} c_i^{(k)} \otimes d_i^{(k)} + \phi^{(k)}(\boldsymbol{X}^{(k)}) \tag{72}$$

where $\phi^{(k)} = \sum_{i=\hat{R}^{(k)}+1}^{R^{(k)}} \lambda_i^{(k)} a_i^{(k)} \otimes b_i^{(k)} \otimes c_i^{(k)} \otimes d_i^{(k)}$, $\phi^{(k)}(\boldsymbol{X}^{(k)})$ denotes the multilinear operation of the tensor $\phi^{(k)}$ on $\boldsymbol{X}^{(k)}$, i.e., $\{\phi^{(k)}(\boldsymbol{X}^{(k)})\}_{i,j} = \sum_{k,l} \phi_{i,j,k,l}^{(k)} \boldsymbol{X}_{k,l}^{(k)}$ and $a_i^{(k)}, b_i^{(k)}, \hat{a}_i^{(k)}, \hat{b}_i^{(k)} \in \mathbb{R}^{s_k}$. Similarly, the input and output of the k^{th} layer of the compressed neural nets $\hat{\mathbb{M}}$ satisfy

$$\hat{\boldsymbol{X}}^{(k)} = \mathsf{ReLU}\left(\hat{\boldsymbol{Y}}^{(k-1)}\right) \tag{73}$$

$$\hat{\boldsymbol{Y}}^{(k)} = \sum_{i=1}^{\hat{R}^{(k)}} \hat{\lambda}_i^{(k)} (\hat{a}_i^{(k)})^\top \hat{\boldsymbol{X}}^{(k)} \hat{b}_i^{(k)} \hat{c}_i^{(k)} \otimes \hat{d}_i^{(k)}. \tag{74}$$

E.2 Characterizing Compressibility of FC Networks with CPL

Now we characterize the compressibility of the fully connected network with CPL $\mathbb{M}$ through properties defined in the following, namely reshaping factor, tensorization factor, layer cushion and tensor noise bound.

Definition E.2. (reshaping factor). The *reshaping factor* $\rho^{(k)}$ of layer k is defined to be the smallest value $\rho^{(k)}$ such that for any $\boldsymbol{x} \in S$,

$$\left\|\boldsymbol{X}^{(\boldsymbol{k})}\right\| \leq \rho^{(k)} \left\|\boldsymbol{X}^{(\boldsymbol{k})}\right\|_{\mathsf{F}} \tag{75}$$

The reshaping factor upper bounds the ratio between the spectral norm and Frobenius norm of the reshaped input in the k^{th} layer over any data example in the training dataset. Reshaping the vector examples into matrix examples improves the compressibility of the network (i.e., renders smaller $\rho^{(k)}$) as illustrated and empirically verified in Su et al. (2018). Note that $\hat{\boldsymbol{X}}^{(k)}$ is the input to the k^{th} layer of the compressed network $\hat{\mathbb{M}}$, and $\rho^{(k)} \leq 1, \forall k$.

Definition E.3. (tensorization factor) The *tensorization factor* $\{t_j^{(k)}\}_{j=1}^{R^{(k)}}$ of the k^{th} layer regarding the network with CPL $\mathbb{M}$ and the original network $\mathbb{M}$ is defined as:

$$t_j^{(k)} = \sum_{r=1}^{j} |\lambda_r^{(k)}|, \forall j. \tag{76}$$

The tensorization factor measures the amplitudes of the leading components. By Lemma C.5, the tensorization factor is the upper bound of operator norm of the weight tensor.

Definition E.4. (layer cushion). Our definition of *layer cushion* for each layer k is similar to Arora et al. (2018). The layer cushion $\zeta^{(k)}$ of layer k is defined to be the largest value $\zeta^{(k)}$ such that for any $\boldsymbol{x} \in S$, $\zeta^{(k)} \left\|\boldsymbol{A}^{(\boldsymbol{k})}\right\|_{\mathsf{F}} \left\|\boldsymbol{x}^{(\boldsymbol{k})}\right\| \leq \left\|\boldsymbol{x}^{(\boldsymbol{k+1})}\right\|$.

The layer cushion defined in Arora et al. (2018) is sligntly larger than ours since our RHS is $\left\|\boldsymbol{x}^{(\boldsymbol{k+1})}\right\| = \mathsf{ReLU}\left(\boldsymbol{A}^{(\boldsymbol{k})}\boldsymbol{x}^{(\boldsymbol{k})}\right)$ while the RHS of the inequality in the definition of layer cushion in Arora et al. (2018) is $\boldsymbol{A}^{(\boldsymbol{k})}\boldsymbol{x}^{(\boldsymbol{k})}$. The layer cushion under our settings also considers how much smaller the output $\left\|\boldsymbol{x}^{(\boldsymbol{k+1})}\right\|$ is compared to is compared to the upper bound $\left\|\boldsymbol{A}^{(\boldsymbol{k})}\right\|_{\mathsf{F}} \left\|\boldsymbol{x}^{(\boldsymbol{k})}\right\|$.

Definition E.5. (tensor noise bound). The *tensor noise bound* $\{\xi^{(k)}\}_{j=1}^{R^{(k)}}$ of the the k^{th} layer measures the amplitudes of the remaining components after pruning out the ones with amplitudes smaller than the j^{th} component:

$$\xi_j^{(k)} := \sum_{r=j+1}^{R^{(k)}} |\lambda_r^{(k)}| \tag{77}$$

The tensor noise bound measures the amplitudes of the CP components that are pruned out by the compression algorithm, and the smaller it is, the more low-rank the weight matrix is. We will see that a network equipped with CPL will be much more low-rank than standard networks.

E.3 Generalization Guarantee of Fully Connected Neural Networks

We have introduced the compression mechanism in Algorithm 2. For a fully connected network with CPL $\mathbb{M}$ that is characterized by the properties such as reshaping factor, tensorization factor, layer cushion and tensor noise bound, in section E.2, we derive the generalization error bound of a compression network with any chosen ranks $\{\hat{R}^{(k)}\}_{k=1}^{n}$ as follows.

Theorem E.6. For any fully connected network $\mathbb{M}$ of n layers satisfying the Assumptions E.1, Algorithm 2 generates a compressed network $\hat{\mathbb{M}}$ such that with high probability over the training set , the expected error $L_0(\hat{\mathbb{M}})$ is bounded by

$$L_0(\hat{\mathbb{M}}) \leq \hat{L}_\gamma(\mathbb{M}) + \tilde{O}\Big(\sqrt{\frac{\sum_{k=1}^{n} \hat{R}^{(k)}(2s^{(k)} + 2s^{(k+1)} + 1)}{m}}\Big) \tag{78}$$

for any margin $\gamma \geq 0$, and the rank of the k^{th} layer, $\hat{R}^{(k)}$, satisfies that

$$\hat{R}^{(k)} = \min\Big\{ j \in [R^{(k)}] \;\Big|\; n\rho^{(k)}\xi_j^{(k)}\Pi_{i=k+1}^{n} t^{(i)} \leq \frac{\gamma}{2\max_{\boldsymbol{x}\in S}\|\mathbb{M}(\boldsymbol{x})\|_{\mathsf{F}}} \Big\|\boldsymbol{A}^{(k)}\Big\|_{\mathsf{F}} \Pi_{i=k}^{n}\zeta^{(i)} \Big\}$$

and $\rho^{(k)}, t^{(k)}, \zeta^{(k)}$ are reshaping factor, tensorization factor, layer cushion and tensor noise bound of the k^{th} layer in Definitions E.2, E.3, and E.4 respectively. $\xi_j^{(k)}$ is defined in the same way with $\xi^{(k)}$, where $\hat{R}^{(k)}$ is replaced by j.

The generalization error of the compressed network $L_0(\hat{\mathbb{M}})$ depends on the compressibility of the $\mathbb{M}$. The compressibility of the $\mathbb{M}$ determines the rank that the compression mechanism should select according to Theorem E.6, which depends on reshaping factor $\rho^{(k)}$, tensorization factor $t^{(k)}$, layer cushion $\zeta^{(k)}$ and tensor noise bound $\xi_j^{(k)}$.

Proof sketch of Theorem E.6: To prove this theorem, we introduce the following Lemma E.7, which reveals that the difference between the output of the original fully connected network $\mathbb{M}$ and that of the compressed $\mathbb{M}$ is bounded by $\epsilon\|\mathbb{M}(\boldsymbol{x})\|_{\mathsf{F}}$. Then we show the covering number of the compressed network $\mathbb{M}$ by approximating each parameter with some certain accuracy is $\tilde{O}(d)$ w.r.t to a given ϵ. After bounding the covering number, the rest of the proof follows from conventional learning theory.

Lemma E.7. For any fully connected network $\mathbb{M}$ of n layers satisfying Assumption E.1 , Algorithm 2 generates a compressed $Tensorial-FC$ $\hat{\mathbb{M}}$ where for any $\boldsymbol{x} \in S$ and any error $0 \leq \epsilon \leq 1$:

$$\Big\|\mathbb{M}(\boldsymbol{x}) - \hat{\mathbb{M}}(\boldsymbol{X})\Big\|_{\mathsf{F}} \leq \epsilon\|\mathbb{M}(\boldsymbol{x})\|_{\mathsf{F}} \tag{79}$$

The compressed $Tensorial-FC$ $\hat{\mathbb{M}}$ consists of $\sum_{k=1}^{n}\hat{R}^{(k)}[2(s^{(k)}+s^{(k+1)})+1)]$ number of parameters, where each $\hat{R}^{(k)}$ is defined as what is stated in Algorithm 4: for each layer $k \in [n]$,

$$\hat{R}^{(k)} = \min\Big\{ j \in [R^{(k)}] \;\Big|\; \rho^{(k)}\xi_j^{(k)}\Pi_{i=k+1}^{n} t^{(i)} \leq \frac{\epsilon}{n} \Big\|\boldsymbol{A}^{(k)}\Big\|_{\mathsf{F}} \Pi_{i=k}^{n}\zeta^{(i)} \Big\}$$

The complete proofs are in E.4

E.4 Complete Proofs of Fully Connected Neural Networks

To prove Lemma E.7, Lemma E.8 (introduced below) is needed.

Lemma E.8. For any fully connected network $\mathbb{M}$ of n layers satisfying the assumptions in section 3, given a list of ranks $\{\hat{R}^{(k)}\}_{i=1}^{n}(\forall k, \hat{R}^{(k)} \leq R^{(k)})$, after tensorizing each layer in $\mathbb{M}$ and making $\mathbb{M}$ into $\mathbb{M}$, Algorithm 6 generates a compressed tensorial neural network $\hat{\mathbb{M}}$ with $\sum_{k=1}^{n} r^{(k)}[2(s^{(k)}+s^{(k+1)})+1)]$ total parameters where for any $\boldsymbol{x} \in S$:

$$\Big\|\mathbb{M}(\boldsymbol{x}) - \hat{\mathbb{M}}(\boldsymbol{X})\Big\|_{\mathsf{F}} \leq \Big(\sum_{k=1}^{n} \frac{\rho^{(k)}\xi^{(k)}}{\zeta^{(k)}\left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}}} \cdot \Pi_{i=k+1}^{n}\frac{t^{(i)}}{\zeta^{(i)}}\Big)\|\mathbb{M}(\boldsymbol{x})\|_{\mathsf{F}}$$

where $\boldsymbol{X}$ is the matricized version of $\boldsymbol{x}$, and $\rho^{(k)}, t^{(k)}, \zeta^{(k)}, \xi^{(k)}$ are reshaping factor, tensorization factor, layer cushion, and tensor noise bound of the k^{th} layer in Definitions E.2, E.3, E.4, and E.5 respectively.

Proof. (of Lemma E.8) Based on Algorithm 2, since for each layer k in the compressed network $\hat{\mathbb{M}}$, representing $\{\hat{\lambda}_i^{(k)}, \hat{a}_i^{(k)}, \hat{b}_i^{(k)}, \hat{c}_i^{(k)}, \hat{d}_i^{(k)}\}_{i=1}^{\hat{R}^{(k)}}$ only needs $\hat{R}^{(k)}[2(s^{(k)} + s^{(k+1)}) + 1)]$ parameters, the total number of parameters in $\hat{\mathbb{M}}$ is $\sum_{k=1}^{n} \hat{R}^{(k)}[2(s^{(k)} + s^{(k+1)}) + 1)]$.

Then as for any $\boldsymbol{x} \in S$, $\mathbb{M}(\boldsymbol{x}) = \mathbb{M}(\boldsymbol{X})$, and by construction, $\mathbb{M}(\boldsymbol{X}) = \boldsymbol{X}^{(n+1)}$ and $\hat{\mathbb{M}}(\boldsymbol{X}) = \hat{\boldsymbol{X}}^{(n+1)}$, we can prove the lemma by showing $\left\|\boldsymbol{X}^{(n+1)} - \hat{\boldsymbol{X}}^{(n+1)}\right\|_{\mathsf{F}}$ satisfies the above inequality, and we will prove this by induction. Notice

Induction Hypothesis: For any layer $m \geq 0$, $\left\|\boldsymbol{X}^{(m)} - \hat{\boldsymbol{X}}^{(m)}\right\|_{\mathsf{F}} \leq (\sum_{k=1}^{m-1} \frac{\rho^{(k)}\xi^{(k)}}{\zeta^{(k)}\left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}}} \cdot \Pi_{i=k+1}^{m-1} \frac{t^{(i)}}{\zeta^{(i)}}) \left\|\boldsymbol{X}^{(m)}\right\|_{\mathsf{F}}$

Base case: when $m = 1$, the above inequality hold trivially as $\boldsymbol{X}^{(1)} = \hat{\boldsymbol{X}}^{(1)}$ as we cannot modify the input, and the RHS is always ≥ 0.

Inductive Step: Now we assume show that the induction hypothesis is true for all m, let us look what happens at layer $m+1$. As we assume perfect recovery in each layer, $\forall k$, $\{\hat{\lambda}_i^{(k)}, \hat{a}_i^{(k)}, \hat{b}_i^{(k)}, \hat{c}_i^{(k)}, \hat{d}_i^{(k)}\}_{i=1}^{\hat{R}^{(k)}} = \{\lambda_i^{(k)}, a_i^{(k)}, b_i^{(k)}, c_i^{(k)}, d_i^{(k)}\}_{i=1}^{\hat{R}^{(k)}}$.

Let $\phi^{(k)} := \sum_{i=\hat{R}^{(k)}+1}^{R^{(k)}} \lambda_i^{(k)} a_i^{(k)} \otimes b_i^{(k)} \otimes c_i^{(k)} \otimes d_i^{(k)}$, and note that $\mathcal{M}^{(k)} = \hat{\mathcal{M}}^{(k)} + \phi$.

Then we have

$$
\begin{aligned}
&\left\|\boldsymbol{X}^{(m+1)} - \hat{\boldsymbol{X}}^{(m+1)}\right\|_{\mathsf{F}} \\
&\quad = \left\|\mathsf{ReLU}\left(Y^{(m)}\right) - \mathsf{ReLU}\left(\hat{Y}^{(m)}\right)\right\|_{\mathsf{F}} \\
&\quad \leq \left\|\sum_{i=1}^{\hat{R}^{(m)}} \lambda_i^{(m)} (a_i^{(m)})^\top \boldsymbol{X}^{(m)} b_i^{(m)} c_i^{(m)} \otimes d_i^{(m)} + \phi^{(m)}(\boldsymbol{X}^{(m)}) - \sum_{i=1}^{\hat{R}^{(m)}} \hat{\lambda}_i^{(m)} (\hat{a}_i^{(m)})^\top \hat{\boldsymbol{X}}^{(m)} \hat{b}_i^{(m)} \hat{c}_i^{(m)} \otimes \hat{d}_i^{(m)}\right\|_{\mathsf{F}} \\
&\quad = \left\|\sum_{i=1}^{\hat{R}^{(m)}} \lambda_i^{(m)} (a_i^{(m)})^\top (\boldsymbol{X}^{(m)} - \hat{\boldsymbol{X}}^{(m)}) b_i^{(m)} c_i^{(m)} \otimes d_i^{(m)} + \phi^{(m)}(\boldsymbol{X}^{(m)})\right\|_{\mathsf{F}}
\end{aligned}
$$

So

$$
\begin{aligned}
&\left\|\boldsymbol{X}^{(m+1)} - \hat{\boldsymbol{X}}^{(m+1)}\right\|_{\mathsf{F}} \\
&\qquad \leq \left\|\sum_{i=1}^{\hat{R}^{(m)}} \lambda_i^{(m)} (a_i^{(m)})^\top (\boldsymbol{X}^{(m)} - \hat{\boldsymbol{X}}^{(m)}) b_i^{(m)} c_i^{(m)} \otimes d_i^{(m)}\right\|_{\mathsf{F}} + \left\|\phi^{(m)}(\boldsymbol{X}^{(m)})\right\|_{\mathsf{F}}
\end{aligned}
$$

As $\phi^{(m)}(\boldsymbol{X}^{(m)}) = \sum_{i=\hat{R}^{(k)}+1}^{R^{(k)}} \lambda_i^{(k)} (a_i^{(k)})^\top \boldsymbol{X}^{(m)} b_i^{(k)} c_i^{(k)} \otimes d_i^{(k)}$. Since $\{c_i^m\}_i$ and are $\{d_i^m\}_i$ are sets of orthogonal

vectors with unit norms,

$$\begin{aligned}
\left\|\phi^{(m)}(\boldsymbol{X}^{(m)})\right\|_{\mathsf{F}} &= \sqrt{\sum_{i=\hat{R}^{(k)}+1}^{R^{(k)}} [\lambda_i^{(k)}(a_i^{(k)})^\top \boldsymbol{X}^{(m)} b_i^{(k)}]^2} \\
&\leq \sqrt{\sum_{i=\hat{R}^{(k)}+1}^{R^{(k)}} (\lambda_i^{(k)})^2 \left\|a_i^{(k)}\right\|^2 \left\|\boldsymbol{X}^{(m)} b_i^{(k)}\right\|^2} \\
&\leq \sqrt{\sum_{i=\hat{R}^{(k)}+1}^{R^{(k)}} (\lambda_i^{(k)})^2 \left\|\boldsymbol{X}^{(m)}\right\|^2 \left\|b_i^{(k)}\right\|^2} \\
&= \sqrt{\sum_{i=\hat{R}^{(k)}+1}^{R^{(k)}} (\lambda_i^{(k)})^2} \left\|\boldsymbol{X}^{(m)}\right\| \\
&= \xi^{(m)} \left\|\boldsymbol{X}^{(m)}\right\| \\
&\leq \xi^{(m)} \rho^{(m)} \left\|\boldsymbol{X}^{(m)}\right\|_{\mathsf{F}} \\
&\leq \frac{\rho^{(m)} \xi^{(m)} \left\|\boldsymbol{X}^{(m+1)}\right\|_{\mathsf{F}}}{\zeta^{(m)} \left\|\boldsymbol{A}^{(m)}\right\|_{\mathsf{F}}}
\end{aligned}$$

Similarly, we can bound $\left\|\sum_{i=1}^{\hat{R}^{(m)}} \lambda_i^{(m)}(a_i^{(m)})^\top(\boldsymbol{X}^{(m)} - \hat{\boldsymbol{X}}^{(m)})b_i^{(m)}c_i^{(m)} \otimes d_i^{(m)}\right\|_{\mathsf{F}}$ as follows:

$$\begin{aligned}
&\left\|\sum_{i=1}^{\hat{R}^{(m)}} \lambda_i^{(m)}(a_i^{(m)})^\top(\boldsymbol{X}^{(m)} - \hat{\boldsymbol{X}}^{(m)})b_i^{(m)}c_i^{(m)} \otimes d_i^{(m)}\right\|_{\mathsf{F}} \\
&= \sqrt{\sum_{i=1}^{\hat{R}^{(m)}} [\lambda_i^{(m)}(a_i^{(m)})^\top(\boldsymbol{X}^{(m)} - \hat{\boldsymbol{X}}^{(m)})b_i^{(m)}]^2} \\
&\leq \sqrt{\sum_{i=1}^{\hat{R}^{(m)}} (\lambda_i^{(m)})^2} \left\|\boldsymbol{X}^{(m)} - \hat{\boldsymbol{X}}^{(m)}\right\| \\
&\leq \sqrt{\sum_{i=1}^{\hat{R}^{(m)}} (\lambda_i^{(m)})^2} \left\|\boldsymbol{X}^{(m)} - \hat{\boldsymbol{X}}^{(m)}\right\|_{\mathsf{F}} \\
&= \sqrt{(t^{(m)})^2 \left\|\boldsymbol{A}^{(m)}\right\|_{\mathsf{F}}^2} \left\|\boldsymbol{X}^{(m)} - \hat{\boldsymbol{X}}^{(m)}\right\|_{\mathsf{F}} \\
&\leq t^{(m)} \left\|\boldsymbol{A}^{(m)}\right\|_{\mathsf{F}} \cdot \Big(\sum_{k=1}^{m-1} \frac{\rho^{(k)}\xi^{(k)}}{\zeta^{(k)} \left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}}} \cdot \Pi_{i=k+1}^{m-1} \frac{t^{(i)}}{\zeta^{(i)}}\Big) \left\|\boldsymbol{X}^{(m)}\right\|_{\mathsf{F}} \\
&\leq \rho^{(m)} t^{(m)} \left\|\boldsymbol{A}^{(m)}\right\|_{\mathsf{F}} \frac{\left\|\boldsymbol{X}^{(m+1)}\right\|_{\mathsf{F}}}{\zeta^{(m)} \left\|\boldsymbol{A}^{(\boldsymbol{m})}\right\|_{\mathsf{F}}} \times \Big(\sum_{k=1}^{m-1} \frac{\rho^{(k)}\xi^{(k)}}{\zeta^{(k)} \left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}}} \cdot \Pi_{i=k+1}^{m-1} \frac{t^{(i)}}{\zeta^{(i)}}\Big) \\
&= \Big(\sum_{k=1}^{m-1} \frac{\rho^{(k)}\xi^{(k)}}{\zeta^{(k)} \left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}}} \cdot \Pi_{i=k+1}^{m} \frac{t^{(i)}}{\zeta^{(i)}}\Big) \cdot \left\|\boldsymbol{X}^{(m+1)}\right\|_{\mathsf{F}}
\end{aligned}$$

Combining the above two terms together, we have

$$\begin{aligned}
&\left\|\boldsymbol{X}^{(m+1)}-\hat{\boldsymbol{X}}^{(m+1)}\right\|_{\mathsf{F}} \\
&\quad\leq\Big(\sum_{k=1}^{m-1}\frac{\rho^{(k)}\xi^{(k)}}{\zeta^{(k)}\left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}}}\cdot\Pi_{i=k+1}^{m}\frac{t^{(i)}}{\zeta^{(i)}}\Big)\cdot\left\|\boldsymbol{X}^{(m+1)}\right\|_{\mathsf{F}}+\frac{\rho^{(m)}\xi^{(m)}\left\|\boldsymbol{X}^{(m+1)}\right\|_{\mathsf{F}}}{\zeta^{(m)}\left\|\boldsymbol{A}^{(m)}\right\|_{\mathsf{F}}} \\
&\quad=\Big(\sum_{k=1}^{m-1}\frac{\rho^{(k)}\xi^{(k)}}{\zeta^{(k)}\left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}}}\cdot\Pi_{i=k+1}^{m}\frac{t^{(i)}}{\zeta^{(i)}}+\frac{\rho^{(m)}\xi^{(m)}}{\zeta^{(m)}\left\|\boldsymbol{A}^{(m)}\right\|_{\mathsf{F}}}\Big)\cdot\left\|\boldsymbol{X}^{(m+1)}\right\|_{\mathsf{F}} \\
&\quad=\Big(\sum_{k=1}^{m-1}\frac{\rho^{(k)}\xi^{(k)}}{\zeta^{(k)}\left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}}}\cdot\Pi_{i=k+1}^{m}\frac{t^{(i)}}{\zeta^{(i)}}+\frac{\rho^{(m)}\xi^{(m)}}{\zeta^{(m)}\left\|\boldsymbol{A}^{(m)}\right\|_{\mathsf{F}}}\cdot\Pi_{i=m+1}^{m}\frac{t^{(i)}}{\zeta^{(i)}}\Big)\cdot\left\|\boldsymbol{X}^{(m+1)}\right\|_{\mathsf{F}} \\
&\quad=\Big(\sum_{k=1}^{m}\frac{\rho^{(k)}\xi^{(k)}}{\zeta^{(k)}\left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}}}\cdot\Pi_{i=k+1}^{m}\frac{t^{(i)}}{\zeta^{(i)}}\Big)\cdot\left\|\boldsymbol{X}^{(m+1)}\right\|_{\mathsf{F}}
\end{aligned}$$

Where the second to the last equality is due to the fact that for any $\alpha_i, \beta \in \mathbb{R}$, $(\Pi_{i=k+1}^{k}\alpha_i)\times\beta=\beta$. □

Then we can proceed to prove Lemma E.7:

Proof. (of Lemma E.7) Based on the assumptions of the components from CP decomposition for each $\mathcal{M}^{(k)}$ in section 3, the $\{\hat{R}^{(k)}\}_{k=1}^{n}$ returned by Algorithm 4 will satisfy:

- $\forall k,\ \hat{R}^{(k)}\leq R^{(k)}$
- $\rho^{(k)}\xi^{(k)}\Pi_{i=k+1}^{n}t^{(i)}\leq\frac{\epsilon}{n}\left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}}\Pi_{i=k}^{n}\zeta^{(i)}$

Thus,

$$\frac{\rho^{(k)}\xi^{(k)}}{\zeta^{(k)}\left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}}}\cdot\Pi_{i=k+1}^{n}\frac{t^{(i)}}{\zeta^{(i)}}\leq\frac{\epsilon}{n}$$

Then by lemma E.8,

$$\left\|\mathbb{M}(\boldsymbol{x})-\hat{\mathbb{M}}(\boldsymbol{X})\right\|_{\mathsf{F}}\leq\epsilon\left\|\mathbb{M}(\boldsymbol{x})\right\|_{\mathsf{F}}$$

□

Before proving Theorem E.6, Lemma E.9 (introduced below) is needed.

Lemma E.9. For any fully connected network $\mathbb{M}$ of n layers satisfying the assumptions in section 3 and any margin $\gamma\geq 0$, $\mathbb{M}$ can be compressed to a fully-connected tensorial neural network $\hat{\mathbb{M}}$ with $\sum_{k=1}^{n}\hat{R}^{(k)}[2(s^{(k)}+s^{(k+1)})+1)]$ total parameters such that for any $\boldsymbol{x}\in S$, $\hat{L}_0(\hat{\mathbb{M}})\leq\hat{L}_\gamma(\mathbb{M})$. Here, for each layer k,

$$\hat{R}^{(k)}=\min\left\{j\in[R^{(k)}]\;\middle|\;\rho^{(k)}\xi_j^{(k)}\Pi_{i=k+1}^{n}t^{(i)}\leq\frac{\epsilon}{n}\left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}}\Pi_{i=k}^{n}\zeta^{(i)}\right\}$$

Proof. (of Lemma E.9) If $\gamma\geq 2\max_{\boldsymbol{x}\in S}\left\|\mathbb{M}(\boldsymbol{x})\right\|_{\mathsf{F}}$, for any pair $(x,y)\in S$, we have

$$\begin{aligned}
|\mathbb{M}(\boldsymbol{x})[y]-\max_{j\neq y}\mathbb{M}(\boldsymbol{x})[j]|^2&\leq(|\mathbb{M}(\boldsymbol{x})[y]|+|\max_{j\neq y}\mathbb{M}(\boldsymbol{x})[j]|)^2\\
&\leq 4\max_{\boldsymbol{x}\in S}\left\|\mathbb{M}(\boldsymbol{x})\right\|_{\mathsf{F}}^2\\
&\leq\gamma^2
\end{aligned}$$

Then the output margin of $\mathbb{M}$ cannot be greater than γ for any $\boldsymbol{x}\in S$. Thus $\hat{L}_\gamma(\mathbb{M})=1$.

If $\gamma<2\max_{\boldsymbol{x}\in S}\left\|\mathbb{M}(\boldsymbol{x})\right\|_{\mathsf{F}}$, setting

$$\epsilon=\frac{\gamma}{2\max_{\boldsymbol{x}\in S}\left\|\mathbb{M}(\boldsymbol{x})\right\|_{\mathsf{F}}}$$

in Lemma E.7, we obtain a compressed fully-connected tensorial neural network $\hat{\mathbb{M}}$ with the desired number of parameters and

$$\left\|\mathbb{M}(\boldsymbol{x}) - \hat{\mathbb{M}}(\boldsymbol{X})\right\|_{\mathsf{F}} < \frac{\gamma}{2} \Rightarrow \forall j, |\mathbb{M}(\boldsymbol{x})[j] - \hat{\mathbb{M}}(\boldsymbol{X})[j]| < \frac{\gamma}{2}$$

Then for any pair $(x, y) \in S$, if $\mathbb{M}(\boldsymbol{x})[y] > \gamma + \max_{j \neq y} \mathbb{M}(\boldsymbol{x})[j]$, $\hat{\mathbb{M}}$ classifies $\boldsymbol{x}$ correctly as well because:

$$\hat{\mathbb{M}}(\boldsymbol{X})[y] > \mathbb{M}(\boldsymbol{x})[y] - \frac{\gamma}{2} > \max_{j \neq y} \mathbb{M}(\boldsymbol{x})[j] + \frac{\gamma}{2} > \max_{j \neq y} \hat{\mathbb{M}}(\boldsymbol{X})[j]$$

Thus, $\hat{L_0}(\hat{\mathbb{M}}) \leq \hat{L}_\gamma(\mathbb{M})$. □

Now we prove the main theorem E.6 by bounding the covering number given any ϵ.

Proof. (of Theorem E.6) To be more specific, let us bound the covering number of the compressed network $\hat{\mathbb{M}}$ by approximating each parameter with accuracy μ.

Covering Number Analysis for Fully Connected Neural Network Let $\tilde{T}$ denote the network after approximating each parameter in $\hat{\mathbb{M}}$ with accuracy μ (and $\tilde{\mathcal{T}}^{(k)}$ denote its weight tensor on the k^{th} layer). Based on the given accuracy, we know that $\forall k,\ |\hat{\lambda}_i^{(k)} - \tilde{\lambda}_i^{(k)}| \leq \mu$ and $\left\|\hat{\boldsymbol{a}}_i^{(k)} - \tilde{\boldsymbol{a}}_i^{(k)}\right\| \leq \sqrt{s^{(k)}}\mu$ (similar inequalities also hold for $\hat{\boldsymbol{b}}_i^{(k)}, \hat{\boldsymbol{c}}_i^{(k)}, \hat{\boldsymbol{d}}_i^{(k)}$). For simplicity, in this proof, let us just use $\boldsymbol{a}_i^{(k)}, \boldsymbol{b}_i^{(k)}, \boldsymbol{c}_i^{(k)}, \boldsymbol{d}_i^{(k)}$ to denote $\hat{\boldsymbol{a}}_i^{(k)}, \hat{\boldsymbol{b}}_i^{(k)}, \hat{\boldsymbol{c}}_i^{(k)}, \hat{\boldsymbol{d}}_i^{(k)}$

Notice that

$$\boldsymbol{Y}^{(k)} = \sum_{i=1}^{r^{(k)}} \lambda_i^{(k)} (\boldsymbol{a}_i^{(k)})^\top \boldsymbol{X}^{(k)} \boldsymbol{b}_i^{(k)} \boldsymbol{c}_i^{(k)} \otimes \boldsymbol{d}_i^{(k)}$$

$$\tilde{\boldsymbol{Y}}^{(k)} = \sum_{i=1}^{r^{(k)}} \tilde{\lambda}_i^{(k)} (\tilde{\boldsymbol{a}}_i^{(k)})^\top \tilde{\boldsymbol{X}}^{(k)} \tilde{\boldsymbol{b}}_i^{(k)} \tilde{\boldsymbol{c}}_i^{(k)} \otimes \tilde{\boldsymbol{d}}_i^{(k)}$$

Let $\epsilon^{(k)} = \left\|\tilde{\boldsymbol{Y}}^{(k)} - \boldsymbol{Y}^{(k)}\right\|_{\mathsf{F}}$. Then for each k, let us first bound $|(\boldsymbol{a}_i^{(k)})^\top \boldsymbol{X}^{(k)} \boldsymbol{b}_i^{(k)} - (\tilde{\boldsymbol{a}}_i^{(k)})^\top \tilde{\boldsymbol{X}}^{(k)} \tilde{\boldsymbol{b}}_i^{(k)}|$ and $\left\|\boldsymbol{c}_i^{(k)} \otimes \boldsymbol{d}_i^{(k)} - \tilde{\boldsymbol{c}}_i^{(k)} \otimes \tilde{\boldsymbol{d}}_i^{(k)}\right\|_{\mathsf{F}}$ separately.

Bound $|(\boldsymbol{a}_i^{(k)})^\top \boldsymbol{X}^{(k)} \boldsymbol{b}_i^{(k)} - (\tilde{\boldsymbol{a}}_i^{(k)})^\top \tilde{\boldsymbol{X}}^{(k)} \tilde{\boldsymbol{b}}_i^{(k)}|$: When $k = 1$, we know that $\boldsymbol{X}^{(1)} = \tilde{\boldsymbol{X}}^{(1)}$. Let us first consider the base case where $k = 1$. For simplicity, let $\boldsymbol{a} = \boldsymbol{a}_i^{(1)}$, $\tilde{\boldsymbol{a}} = \tilde{\boldsymbol{a}}_i^{(1)}$, $\boldsymbol{b} = \boldsymbol{b}_i^{(1)}$, $\tilde{\boldsymbol{b}} = \tilde{\boldsymbol{b}}_i^{(1)}$, and $\boldsymbol{X} = \boldsymbol{X}^{(1)}$. Then

$$\begin{aligned}
&|(\boldsymbol{a}_i^{(1)})^\top \boldsymbol{X}^{(1)} \boldsymbol{b}_i^{(1)} - (\tilde{\boldsymbol{a}}_i^{(1)})^\top \tilde{\boldsymbol{X}}^{(1)} \tilde{\boldsymbol{b}}_i^{(1)}| \\
&\quad = |\boldsymbol{a}^\top \boldsymbol{X} \boldsymbol{b} - \tilde{\boldsymbol{a}}^\top \boldsymbol{X} \tilde{\boldsymbol{b}}| \\
&\quad = |\boldsymbol{a}^\top \boldsymbol{X} \boldsymbol{b} - \boldsymbol{a}^\top \boldsymbol{X} \tilde{\boldsymbol{b}} + \boldsymbol{a}^\top \boldsymbol{X} \tilde{\boldsymbol{b}} - \tilde{\boldsymbol{a}}^\top \boldsymbol{X} \tilde{\boldsymbol{b}}| \\
&\quad = |\boldsymbol{a}^\top \boldsymbol{X} (\boldsymbol{b} - \tilde{\boldsymbol{b}}) + (\boldsymbol{a} - \tilde{\boldsymbol{a}})^\top \boldsymbol{X} \tilde{\boldsymbol{b}}| \\
&\quad \leq |\boldsymbol{a}^\top \boldsymbol{X} (\boldsymbol{b} - \tilde{\boldsymbol{b}})| + |(\boldsymbol{a} - \tilde{\boldsymbol{a}})^\top \boldsymbol{X} \tilde{\boldsymbol{b}}| \\
&\quad \leq \left\|\boldsymbol{X}^\top \boldsymbol{a}\right\| \left\|\boldsymbol{b} - \tilde{\boldsymbol{b}}\right\| + \left\|\boldsymbol{a} - \tilde{\boldsymbol{a}}\right\| \left\|\boldsymbol{X} \tilde{\boldsymbol{b}}\right\| \\
&\quad \leq \mu \sqrt{s^{(1)}} \left\|\boldsymbol{X}\right\| (\left\|\boldsymbol{a}\right\| + \left\|\tilde{\boldsymbol{b}}\right\|) \\
&\quad \leq 2\mu \sqrt{s^{(1)}} \left\|\boldsymbol{X}\right\|
\end{aligned}$$

The second to the last inequality is because singular values are invariant to matrix transpose.

When $k \geq 1$, similarly, let $\boldsymbol{a} = \boldsymbol{a}_i^{(k)}$, $\tilde{\boldsymbol{a}} = \tilde{\boldsymbol{a}}_i^{(k)}$ (define b in a similar way), $\boldsymbol{X} = \boldsymbol{X}^{(k)}$, and $\tilde{\boldsymbol{X}} = \tilde{\boldsymbol{X}}^{(k)}$. Let

$\boldsymbol{Y} = \boldsymbol{Y}^{(k-1)}$, and $\tilde{\boldsymbol{Y}} = \tilde{\boldsymbol{Y}}^{(k-1)}$ (basically the output from the $(k-1)^{th}$ layer before activation). Then

$$
\begin{aligned}
&|(\boldsymbol{a}_i^{(k)})^\top \boldsymbol{X}^{(k)} \boldsymbol{b}_i^{(k)} - (\tilde{\boldsymbol{a}}_i^{(k)})^\top \tilde{\boldsymbol{X}}^{(k)} \tilde{\boldsymbol{b}}_i^{(k)}| \\
&\quad = |\boldsymbol{a}^\top \boldsymbol{X} \boldsymbol{b} - \tilde{\boldsymbol{a}}^\top \tilde{\boldsymbol{X}} \tilde{\boldsymbol{b}}| \\
&\quad = |\boldsymbol{a}^\top \boldsymbol{X} \boldsymbol{b} - \tilde{\boldsymbol{a}}^\top \boldsymbol{X} \tilde{\boldsymbol{b}} + \tilde{\boldsymbol{a}}^\top \boldsymbol{X} \tilde{\boldsymbol{b}} - \tilde{\boldsymbol{a}}^\top \tilde{\boldsymbol{X}} \tilde{\boldsymbol{b}}| \\
&\quad \leq |\boldsymbol{a}^\top \boldsymbol{X} \boldsymbol{b} - \tilde{\boldsymbol{a}}^\top \boldsymbol{X} \tilde{\boldsymbol{b}}| + |\tilde{\boldsymbol{a}}^\top \boldsymbol{X} \tilde{\boldsymbol{b}} - \tilde{\boldsymbol{a}}^\top \tilde{\boldsymbol{X}} \tilde{\boldsymbol{b}}| \\
&\quad \leq 2\mu \sqrt{s^{(k)}} \|\boldsymbol{X}\| + \left\| \boldsymbol{X} - \tilde{\boldsymbol{X}} \right\|, \text{by base case } k = 1 \\
&\quad = 2\mu \sqrt{s^{(k)}} \|\boldsymbol{X}\| + \left\| \mathsf{ReLU}\left(\boldsymbol{Y}\right) - \mathsf{ReLU}\left(\tilde{\boldsymbol{Y}}\right) \right\| \\
&\quad \leq 2\mu \sqrt{s^{(k)}} \|\boldsymbol{X}\| + \left\| \mathsf{ReLU}\left(\boldsymbol{Y}\right) - \mathsf{ReLU}\left(\tilde{\boldsymbol{Y}}\right) \right\|_{\mathsf{F}} \\
&\quad \leq 2\mu \sqrt{s^{(k)}} \|\boldsymbol{X}\| + \left\| \boldsymbol{Y} - \tilde{\boldsymbol{Y}} \right\|_{\mathsf{F}} \\
&\quad = 2\mu \sqrt{s^{(k)}} \|\boldsymbol{X}\| + \epsilon^{(k-1)}
\end{aligned}
$$

Then we can also bound $|\lambda_i^{(k)} (\boldsymbol{a}_i^{(k)})^\top \boldsymbol{X}^{(k)} \boldsymbol{b}_i^{(k)} - \tilde{\lambda}_i^{(k)} (\tilde{\boldsymbol{a}}_i^{(k)})^\top \tilde{\boldsymbol{X}}^{(k)} \tilde{\boldsymbol{b}}_i^{(k)}|$. For simplicity, let $\lambda = \lambda_i^{(k)}$, $\tilde{\lambda} = \tilde{\lambda}_i^{(k)}$, $x = (\boldsymbol{a}_i^{(k)})^\top \boldsymbol{X}^{(k)} \boldsymbol{b}_i^{(k)}$, and $\tilde{x} = (\tilde{\boldsymbol{a}}_i^{(k)})^\top \tilde{\boldsymbol{X}}^{(k)} \tilde{\boldsymbol{b}}_i^{(k)}$. Then

$$
\begin{aligned}
&|\lambda_i^{(k)} (\boldsymbol{a}_i^{(k)})^\top \boldsymbol{X}^{(k)} \boldsymbol{b}_i^{(k)} - \tilde{\lambda}_i^{(k)} (\tilde{\boldsymbol{a}}_i^{(k)})^\top \tilde{\boldsymbol{X}}^{(k)} \tilde{\boldsymbol{b}}_i^{(k)}| \\
&\quad = |\lambda x - \hat{\lambda} \hat{x}| \\
&\quad \leq |\lambda - \hat{\lambda}||x| + |\hat{\lambda}||x - \hat{x}| \\
&\quad \leq |\lambda - \hat{\lambda}||x| + |\lambda||x - \hat{x}|, \text{we can pick } |\hat{\lambda}| \leq |\lambda| \\
&\quad \leq \mu|x| + |\lambda| \times (2\mu \sqrt{s^{(k)}} \left\| \boldsymbol{X}^{(k)} \right\| + \epsilon^{(k-1)}) \\
&\quad \leq \mu \left\| \boldsymbol{X}^{(k)} \right\| + 2\mu \left\| \boldsymbol{X}^{(k)} \right\| |\lambda| \sqrt{s^{(k)}} + |\lambda| \epsilon^{(k-1)} \\
&\quad = \mu \left\| \boldsymbol{X}^{(k)} \right\| (1 + 2|\lambda| \sqrt{s^{(k)}}) + |\lambda| \epsilon^{(k-1)}
\end{aligned}
$$

Bound $\left\|\boldsymbol{c}_i^{(k)} \otimes \boldsymbol{d}_i^{(k)} - \tilde{\boldsymbol{c}}_i^{(k)} \otimes \tilde{\boldsymbol{d}}_i^{(k)}\right\|_{\mathsf{F}}$: Similarly let $\boldsymbol{c} = \boldsymbol{c}_i^{(k)}$ and $\tilde{\boldsymbol{c}} = \tilde{\boldsymbol{c}}_i^{(k)}$ (define $\boldsymbol{d}$ and $\tilde{\boldsymbol{d}}$ in a similar way). Then

$$
\begin{aligned}
&\left\|\boldsymbol{c}_i^{(k)} \otimes \boldsymbol{d}_i^{(k)} - \tilde{\boldsymbol{c}}_i^{(k)} \otimes \tilde{\boldsymbol{d}}_i^{(k)}\right\|_{\mathsf{F}}^2 \\
&\quad= \left\|\boldsymbol{c}\boldsymbol{d}^\top - \tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top\right\|_{\mathsf{F}}^2 \\
&\quad= \mathrm{Tr}((\boldsymbol{c}\boldsymbol{d}^\top - \tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top)^\top(\boldsymbol{c}\boldsymbol{d}^\top - \tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top)) \\
&\quad= \mathrm{Tr}((\boldsymbol{d}\boldsymbol{c}^\top - \tilde{\boldsymbol{d}}\tilde{\boldsymbol{c}}^\top)(\boldsymbol{c}\boldsymbol{d}^\top - \tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top)) \\
&\quad= \mathrm{Tr}(\boldsymbol{d}\boldsymbol{c}^\top\boldsymbol{c}\boldsymbol{d}^\top - \tilde{\boldsymbol{d}}\tilde{\boldsymbol{c}}^\top\boldsymbol{c}\boldsymbol{d}^\top - \boldsymbol{d}\boldsymbol{c}^\top\tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top + \tilde{\boldsymbol{d}}\tilde{\boldsymbol{c}}^\top\tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top) \\
&\quad= \mathrm{Tr}(\boldsymbol{d}\boldsymbol{c}^\top\boldsymbol{c}\boldsymbol{d}^\top) - \mathrm{Tr}(\tilde{\boldsymbol{d}}\tilde{\boldsymbol{c}}^\top\boldsymbol{c}\boldsymbol{d}^\top) - \mathrm{Tr}(\boldsymbol{d}\boldsymbol{c}^\top\tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top) + \mathrm{Tr}(\tilde{\boldsymbol{d}}\tilde{\boldsymbol{c}}^\top\tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top) \\
&\quad= \mathrm{Tr}(\boldsymbol{c}^\top\boldsymbol{c}\boldsymbol{d}^\top\boldsymbol{d}) - \mathrm{Tr}(\boldsymbol{c}^\top\tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top\boldsymbol{d}) + \mathrm{Tr}(\tilde{\boldsymbol{c}}^\top\tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top\tilde{\boldsymbol{d}}) - \mathrm{Tr}(\tilde{\boldsymbol{c}}^\top\boldsymbol{c}\boldsymbol{d}^\top\tilde{\boldsymbol{d}}) \\
&\quad= \mathrm{Tr}(\boldsymbol{c}^\top(\boldsymbol{c}\boldsymbol{d}^\top - \tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top)\boldsymbol{d} + \tilde{\boldsymbol{c}}^\top(\tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top - \boldsymbol{c}\boldsymbol{d}^\top)\tilde{\boldsymbol{d}}) \\
&\quad= \boldsymbol{c}^\top(\boldsymbol{c}\boldsymbol{d}^\top - \tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top)\boldsymbol{d} + \tilde{\boldsymbol{c}}^\top(\tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top - \boldsymbol{c}\boldsymbol{d}^\top)\tilde{\boldsymbol{d}} \\
&\quad\le \|\boldsymbol{c}\| \left\|\boldsymbol{c}\boldsymbol{d}^\top - \tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top\right\| \|\boldsymbol{d}\| + \|\tilde{\boldsymbol{c}}\| \left\|\tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top - \boldsymbol{c}\boldsymbol{d}^\top\right\| \|\boldsymbol{d}\| \\
&\quad\le 2\left\|\boldsymbol{c}\boldsymbol{d}^\top - \tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top\right\|, \text{as the norms of } \boldsymbol{c}, \boldsymbol{d}, \tilde{\boldsymbol{c}}, \tilde{\boldsymbol{d}} \text{ are} \le 1 \\
&\quad= 2\left\|\boldsymbol{c}\boldsymbol{d}^\top - \boldsymbol{c}\tilde{\boldsymbol{d}}^\top + \boldsymbol{c}\tilde{\boldsymbol{d}}^\top - \tilde{\boldsymbol{c}}\tilde{\boldsymbol{d}}^\top\right\| \\
&\quad= 2\left\|\boldsymbol{c}(\boldsymbol{d}^\top - \tilde{\boldsymbol{d}}^\top) + (\boldsymbol{c} - \tilde{\boldsymbol{c}})\tilde{\boldsymbol{d}}^\top\right\| \\
&\quad\le 2(\left\|\boldsymbol{c}(\boldsymbol{d}^\top - \tilde{\boldsymbol{d}}^\top)\right\| + \left\|(\boldsymbol{c} - \tilde{\boldsymbol{c}})\tilde{\boldsymbol{d}}^\top\right\|) \\
&\quad\le 2(\|\boldsymbol{c}\| \left\|\boldsymbol{d} - \tilde{\boldsymbol{d}}\right\| + \|\boldsymbol{d}\| \|\boldsymbol{c} - \tilde{\boldsymbol{c}}\|), \text{as they are rank 1 matrices} \\
&\quad\le 4\sqrt{s^{(k+1)}}\mu
\end{aligned}
$$

Bound $\epsilon^{(k)} = \left\|\tilde{\boldsymbol{Y}}^{(k)} - \boldsymbol{Y}^{(k)}\right\|_{\mathsf{F}}$: Similarly, for simplicity, let $w_i = \lambda_i^{(k)}(\boldsymbol{a}_i^{(k)})^\top \boldsymbol{X}^{(k)}\boldsymbol{b}_i^{(k)}$, $\tilde{w}_i = \tilde{\lambda}_i^{(k)}(\tilde{\boldsymbol{a}}_i^{(k)})^\top \tilde{\boldsymbol{X}}^{(k)}\tilde{\boldsymbol{b}}_i^{(k)}$, $\boldsymbol{U_i} = \boldsymbol{c}_i^{(k)} \otimes \boldsymbol{d}_i^{(k)}$, and $\tilde{\boldsymbol{U}}_i = \tilde{\boldsymbol{c}}_i^{(k)} \otimes \tilde{\boldsymbol{d}}_i^{(k)}$.

Since $\left\|\tilde{\boldsymbol{Y}}^{(k)}-\boldsymbol{Y}^{(k)}\right\|_{\mathsf{F}}=\left\|\sum_{i=1}^{r^{(k)}}w_i\boldsymbol{U_i}-\sum_{i=}^{r^{(k)}}\tilde{d}_i\tilde{\boldsymbol{U}}_i\right\|_{\mathsf{F}}$,

$$
\begin{aligned}
&\left\|\sum_{i=1}^{r^{(k)}}w_i\boldsymbol{U_i}-\sum_{i=}^{r^{(k)}}\tilde{w}_i\tilde{\boldsymbol{U}}_i\right\|_{\mathsf{F}}\\
&\quad=\left\|\sum_{i=1}^{r^{(k)}}(w_i\boldsymbol{U_i}-\tilde{w}_i\tilde{\boldsymbol{U}}_i)\right\|_{\mathsf{F}}\\
&\quad\le\sum_{i=1}^{r^{(k)}}\left\|w_i\boldsymbol{U_i}-\tilde{w}_i\tilde{\boldsymbol{U}}_i\right\|_{\mathsf{F}}\\
&\quad=\sum_{i=1}^{r^{(k)}}\left\|w_i\boldsymbol{U_i}-w_i\tilde{\boldsymbol{U}}_i+w_i\tilde{\boldsymbol{U}}_i-\tilde{w}_i\tilde{\boldsymbol{U}}_i\right\|_{\mathsf{F}}\\
&\quad\le\sum_{i=1}^{r^{(k)}}\left\|w_i\boldsymbol{U_i}-w_i\tilde{\boldsymbol{U}}_i\right\|_{\mathsf{F}}+\left\|w_i\tilde{\boldsymbol{U}}_i-\tilde{w}_i\tilde{\boldsymbol{U}}_i\right\|_{\mathsf{F}}\\
&\quad=\sum_{i=1}^{r^{(k)}}\left\|w_i(\boldsymbol{U_i}-\tilde{\boldsymbol{U}}_i)\right\|_{\mathsf{F}}+\left\|(w_i-\tilde{w}_i)\tilde{\boldsymbol{U}}_i\right\|_{\mathsf{F}}\\
&\quad=\sum_{i=1}^{r^{(k)}}|w_i|\left\|\boldsymbol{U_i}-\tilde{\boldsymbol{U}}_i\right\|_{\mathsf{F}}+|w_i-\tilde{w}_i|\left\|\tilde{\boldsymbol{U}}_i\right\|_{\mathsf{F}}\\
&\quad\le\sum_{i=1}^{r^{(k)}}|w_i|\times\sqrt{4\sqrt{s^{(k+1)}}\mu}+\left(\mu\left\|\boldsymbol{X}^{(k)}\right\|(1+2|\lambda_i|\sqrt{s^{(k)}})+|\lambda_i|\epsilon^{(k-1)}\right)\times\left\|\tilde{\boldsymbol{U}}_i\right\|_{\mathsf{F}}\\
&\quad\le\sum_{i=1}^{r^{(k)}}|\lambda_i^{(k)}|\left\|\boldsymbol{X}^{(k)}\right\|\times\sqrt{4\sqrt{s^{(k+1)}}\mu}+\left(\mu\left\|\boldsymbol{X}^{(k)}\right\|(1+2|\lambda_i|\sqrt{s^{(k)}})+|\lambda_i|\epsilon^{(k-1)}\right)\times\left\|\tilde{\boldsymbol{c}}_i^{(k)}\otimes\tilde{\boldsymbol{d}}_i^{(k)}\right\|_{\mathsf{F}}\\
&\quad=\sum_{i=1}^{r^{(k)}}2|\lambda_i^{(k)}|\left\|\boldsymbol{X}^{(k)}\right\|\times\sqrt{\sqrt{s^{(k+1)}}\mu}+\mu\left\|\boldsymbol{X}^{(k)}\right\|(1+2|\lambda_i|\sqrt{s^{(k)}})+|\lambda_i|\epsilon^{(k-1)}\\
&\quad\le\sum_{i=1}^{r^{(k)}}\mu\left\|\boldsymbol{X}^{(k)}\right\|\left(1+2|\lambda_i^{(k)}|(\sqrt{s^{(k)}}+\sqrt{s^{(k+1)}})\right)+|\lambda_i^{(k)}|\epsilon^{(k-1)},\text{ assume }\sqrt{\sqrt{s^{(k+1)}}\mu}\le\sqrt{s^{(k+1)}}\mu\\
&\quad\le r^{(k)}\times\{\mu\left\|\boldsymbol{X}^{(k)}\right\|\left(1+2|\lambda_{max}^{(k)}|(\sqrt{s^{(k)}}+\sqrt{s^{(k+1)}})\right)+|\lambda_{max}^{(k)}|\epsilon^{(k-1)}\}\\
&\quad\le\mu r^{(k)}[1+2|\lambda_{max}^{(k)}|(\sqrt{s^{(k)}}+\sqrt{s^{(k+1)}})]\left\|\boldsymbol{X}^{(k)}\right\|+r^{(k)}|\lambda_{max}^{(k)}|\epsilon^{(k-1)}
\end{aligned}
\tag{80}
$$

Let $\alpha^{(k)}:=\mu r^{(k)}[1+2|\lambda_{max}^{(k)}|(\sqrt{s^{(k)}}+\sqrt{s^{(k+1)}})]\left\|\boldsymbol{X}^{(k)}\right\|$, and $\beta^{(k)}=r^{(k)}|\lambda_{max}^{(k)}|$, then by the recurrence relationship in 80, the difference between the final output of the two networks are bounded by:

$$
\begin{aligned}
&\left\|\hat{\mathbb{M}}(\boldsymbol{X})-\tilde{\mathbb{M}}(\boldsymbol{X})\right\|_{\mathsf{F}}\\
&\quad=\left\|\mathsf{ReLU}\left(\hat{\boldsymbol{Y}}^{(n)}\right)-\mathsf{ReLU}\left(\boldsymbol{Y}^{(n)}\right)\right\|_{\mathsf{F}}\ (=\boldsymbol{X}^{(n+1)}-\boldsymbol{X}^{(n+1)})\\
&\quad\le\left\|\tilde{\boldsymbol{Y}}^{(n)}-\boldsymbol{Y}^{(n)}\right\|_{\mathsf{F}}\\
&\quad\le\sum_{k=1}^{n}\alpha^{(k)}\Pi_{i=k+1}^{n}\beta^{(i)}
\end{aligned}
$$

Since $\forall k\in[n],\left\|\boldsymbol{X}^{(k)}\right\|\le\Pi_{i=k}^{n}\frac{\rho^{(i)}}{\zeta^{(i)}\left\|\boldsymbol{A}^{(i)}\right\|_{\mathsf{F}}}\left\|\boldsymbol{X}^{(n+1)}\right\|_{\mathsf{F}}$, to obtain an ϵ-cover of the compressed network, we can first

assume $\beta^{(k)} \geq 1\ \forall k \in [n]$. Then μ need to satisfy:

$$\mu \leq \frac{\epsilon}{(r^{(*)}|\lambda^*|)^n \left\|\boldsymbol{X^{(n+1)}}\right\|_{\mathsf{F}} nr^{(*)}(1+4|\lambda^{(*)}|\sqrt{s^{(*)}})(\frac{\rho^{(*)}}{\mu^{(*)}\left\|\boldsymbol{A^{(*)}}\right\|_{\mathsf{F}}})^n}$$

where $r^{(*)} = \max_k r^{(k)}\lambda^{(*)} = \max_{i,k} \lambda_i^{(k)}$, $s^{(*)} = \max_k s^{(k)}$, and $\frac{\rho^{(*)}}{\mu^{(*)}\left\|\boldsymbol{A^{(*)}}\right\|_{\mathsf{F}}} = \max_k \frac{\rho^{(k)}}{\mu^{(k)}\left\|\boldsymbol{A^{(k)}}\right\|_{\mathsf{F}}}$

As when μ is fixed, the number of networks in our cover will at most be $(\frac{1}{\mu})^d$ where d denote the number of parameters in the original network. Hence, the covering number w.r.t to a given ϵ is $\tilde{O}(nd)$ (n is the number of layers in the given neural network). As for practical neural networks, the number of layers n is usually much less than $O(\log(d))$, thus the covering number we obtained w.r.t to a given ϵ is just $\tilde{O}(d)$ for practical neural networks. □

F Neural Networks with Skip Connections

F.1 Problem Setup

For neural nets with skip connections, the current theoretical analyses consider convolutional neural networks with one skip connection used on each layer, since our theoretical results can easily extend to general neural nets with skip connections. Therefore, we used the same the notations for neural nets with skip connections as what we defined for convolutional neural networks.

Forward pass functions Under the above assumptions, the only difference that we need to take into account between our analysis of CNN with skip connections and our analysis of standard CNN is the forward pass functions. In neural networks with skip connections, we have

$$\mathcal{X}^{(k)} = \mathsf{ReLU}\left(\mathcal{Y}^{(k-1)}\right)$$

$$\mathcal{Y}^{(k)} = \mathcal{M}^{(k)}\left(\mathcal{X}^{(k)}\right) + \mathcal{X}^{(k)}$$

and

$$\hat{\mathcal{X}}^{(k)} = \mathsf{ReLU}\left(\hat{\mathcal{Y}}^{(k-1)}\right)$$

$$\hat{\mathcal{Y}}^{(k)} = \hat{\mathcal{M}}^{(k)}\left(\hat{\mathcal{X}}^{(k)}\right) + \hat{\mathcal{X}}^{(k)}$$

where $\mathcal{M}^{(k)}\left(\mathcal{X}^{(k)}\right)$ and $\hat{\mathcal{M}}^{(k)}\left(\hat{\mathcal{X}}^{(k)}\right)$ compute the outputs of the k^{th} convolutional layer.

Similarly, we use *tensorization factor*, *tensor noise bound* and *layer cushion* as in convolutional neural network defined in 4.2, 4.3 and 4.4. But note that the input $\mathcal{X}^{(k)}$ in the definition of *layer cushion* is the input of k^{th} layer after skip connection.

F.2 Generalization Guarantee of Compressed Network Proposed

Theorem F.1. For any convolutional neural network $\mathbb{M}$ of n layers with skip connection satisfying the assumptions in section 3 and any margin $\gamma \geq 0$, Algorithm 1 generates a compressed network $\hat{\mathbb{M}}$ such that with high probability over the training set, the expected error $L_0(\hat{\mathbb{M}})$ is bounded by

$$\hat{L}_\gamma(\mathbb{M}) + \tilde{O}\Big(\sqrt{\frac{\sum_{k=1}^n \hat{R}^{(k)}(s^{(k)} + o^{(k)} + k_x^{(k)} \times k_y^{(k)} + 1)}{m}}\Big) \tag{81}$$

where

$$\hat{R}^{(k)} = \min\left\{j \in [R^{(k)}] | \xi_j^{(k)} \Pi_{i=k+1}^n (t_j^{(i)} + 1) \leq \frac{\gamma}{2n \max_{\mathcal{X}\in S} \left\|\mathbb{M}(\mathcal{X})\right\|_{\mathsf{F}}} \Pi_{i=k}^n \zeta^{(i)} \left\|\mathcal{M}^{(i)}\right\|_{\mathsf{F}}\right\} \tag{82}$$

To prove this theorem, Lemma F.2 is needed.

Lemma F.2. For any convolutional neural network $\mathbb{M}$ of n layers with skip connection satisfying the assumptions in section 3 and any error $0 \leq \epsilon \leq 1$, Algorithm 1 generates a compressed tensorial neural network $\hat{\mathbb{M}}$ such that for any $X \in S$:

$$\left\|\mathbb{M}(\mathcal{X}) - \hat{\mathbb{M}}(\mathcal{X})\right\|_{\mathsf{F}} \leq \epsilon \left\|\mathbb{M}(\mathcal{X})\right\|_{\mathsf{F}} \tag{83}$$

The compressed convolutional neural network $\hat{\mathbb{M}}$ has with $\sum_{k=1}^{n} \hat{R}^{(k)}(s^{(k)} + o^{(k)} + k_x^{(k)} \times k_y^{(k)} + 1)$ total parameters, where each $\hat{R}^{(k)}$ satisfies:

$$\hat{R}^{(k)} = \min\left\{ j \in [R^{(k)}] | \xi_j^{(k)} \Pi_{i=k+1}^{n} (t_j^{(i)} + 1) \leq \frac{\epsilon}{n} \Pi_{i=k}^{n} \zeta^{(i)} \left\|\mathcal{M}^{(i)}\right\|_{\mathsf{F}} \right\} \tag{84}$$

F.3 Complete Proofs of Neural Networks with Skip Connection

To prove Lemma F.2, the following Lemma F.3 is needed.

Lemma F.3. For any convolutional neural network $\mathbb{M}$ of n layers with skip connection satisfying the assumptions in section 3, Algorithm 5 generates a compressed tensorial neural network $\hat{\mathbb{M}}$ where for any $\mathcal{X} \in S$:

$$\left\|\mathbb{M}(\mathcal{X}) - \hat{\mathbb{M}}(\mathcal{X})\right\|_{\mathsf{F}} \leq \left(\sum_{k=1}^{n} \frac{\xi^{(k)}}{\zeta^{(k)} \left\|\mathcal{M}^{(k)}\right\|_{\mathsf{F}}} \prod_{l=k+1}^{n} \frac{t^{(l)} + 1}{\zeta^{(l)} \left\|\mathcal{M}^{(l)}\right\|_{\mathsf{F}}} \right) \left\|\mathbb{M}(\mathcal{X})\right\|_{\mathsf{F}}$$

where ξ, ζ, and t are *tensor noise bound*, *layer cushion*, *tensorization factor* defined in 4.3, 4.4 and 4.2 respectively.

Proof. (of Lemma F.3)

We know by construction, $\mathbb{M}(\mathcal{X}) = \mathcal{X}^{(n+1)}$ and $\hat{\mathbb{M}}(\mathcal{X}) = \hat{\mathcal{X}}^{(n+1)}$, we can just show $\left\|\mathcal{X}^{(n+1)} - \hat{\mathcal{X}}^{(n+1)}\right\|_{\mathsf{F}}$ satisfies the above inequality, and we will prove this by induction. Notice

Induction Hypothesis: For any layer $m > 0$,

$$\left\|\mathcal{X}^{(m)} - \hat{\mathcal{X}}^{(m)}\right\|_{\mathsf{F}} \leq \left(\sum_{k=1}^{m} \frac{\xi^{(k)}}{\zeta^{(k)} \left\|\mathcal{M}^{(k)}\right\|_{\mathsf{F}}} \prod_{l=k+1}^{m} \frac{t^{(l)} + 1}{\zeta^{(l)} \left\|\mathcal{M}^{(l)}\right\|_{\mathsf{F}}} \right) \left\|\mathcal{X}^{(m)}\right\|_{\mathsf{F}}$$

Base case: when $m = 1$, the above inequality hold trivially as $\mathcal{X}^{(1)} = \hat{\mathcal{X}}^{(1)}$ as we cannot modify the input, and the RHS is always ≥ 0.

Inductive Step: Now we assume show that the induction hypothesis is true for all m, then at layer $m + 1$ we have

$$
\begin{aligned}
&\left\|\mathcal{X}^{(m+1)}-\hat{\mathcal{X}}^{(m+1)}\right\|_{\mathsf{F}} \\
&\quad=\left\|\mathsf{ReLU}\left(\mathcal{Y}^{(m)}\right)-\mathsf{ReLU}\left(\hat{\mathcal{Y}}^{(m)}\right)\right\|_{\mathsf{F}} \\
&\quad\leq\left\|\mathcal{Y}^{(m)}-\hat{\mathcal{Y}}^{(m)}\right\|_{\mathsf{F}} \\
&\quad\leq\left\|\mathcal{M}^{(m)}\left(\mathcal{X}^{(m)}\right)+\mathcal{X}^{(m)}-\left(\hat{\mathcal{M}}^{(m)}\left(\hat{\mathcal{X}}^{(m)}\right)+\hat{\mathcal{X}}^{(m)}\right)\right\|_{\mathsf{F}} \\
&\quad\leq\left\|\mathcal{M}^{(m)}\left(\mathcal{X}^{(m)}\right)-\hat{\mathcal{M}}^{(m)}\left(\hat{\mathcal{X}}^{(m)}\right)\right\|_{\mathsf{F}}+\left\|\mathcal{X}^{(m)}-\hat{\mathcal{X}}^{(m)}\right\|_{\mathsf{F}} \\
&\quad\leq\left\|\hat{\mathcal{M}}^{(m)}\left(\mathcal{X}^{(m)}-\hat{\mathcal{X}}^{(m)}\right)+\left(\mathcal{M}^{(m)}-\hat{\mathcal{M}}^{(m)}\right)\left(\mathcal{X}^{(m)}\right)\right\|_{\mathsf{F}}+\left\|\mathcal{X}^{(m)}-\hat{\mathcal{X}}^{(m)}\right\|_{\mathsf{F}} \\
&\quad\leq\sqrt{HW}\left(t^{(m)}+1\right)\left\|\mathcal{X}^{(m)}-\hat{\mathcal{X}}^{(m)}\right\|_{\mathsf{F}}+\sqrt{HW}\xi^{(m)}\left\|\mathcal{X}^{(m)}\right\|_{\mathsf{F}} \\
&\quad\leq\sqrt{HW}\left(t^{(m)}+1\right)\left(\sum_{k=1}^{m-1}\frac{\xi^{(k)}}{\zeta^{(k)}\left\|\mathcal{M}^{(k)}\right\|_{\mathsf{F}}}\prod_{l=k+1}^{m-1}\frac{t^{(l)}+1}{\zeta^{(l)}\left\|\mathcal{M}^{(l)}\right\|_{\mathsf{F}}}\right)\left\|\mathcal{X}^{(m)}\right\|_{\mathsf{F}}+\sqrt{HW}\xi^{(m)}\left\|\mathcal{X}^{(m)}\right\|_{\mathsf{F}} \\
&\quad\leq\left(\sum_{k=1}^{m-1}\frac{\xi^{(k)}}{\zeta^{(k)}\left\|\mathcal{M}^{(k)}\right\|_{\mathsf{F}}}\prod_{l=k+1}^{m-1}\frac{t^{(l)}+1}{\zeta^{(l)}\left\|\mathcal{M}^{(l)}\right\|_{\mathsf{F}}}\right)\frac{(t^{(m)}+1)}{\zeta^{(m)}\left\|\mathcal{M}^{(m)}\right\|_{\mathsf{F}}}\left\|\mathcal{X}^{(m+1)}\right\|_{\mathsf{F}}+\frac{\xi^{(m)}}{\zeta^{(m)}\left\|\mathcal{M}^{(m)}\right\|_{\mathsf{F}}}\left\|\mathcal{X}^{(m+1)}\right\|_{\mathsf{F}} \\
&\quad\leq\left(\sum_{k=1}^{m}\frac{\xi^{(k)}}{\zeta^{(k)}\left\|\mathcal{M}^{(k)}\right\|_{\mathsf{F}}}\prod_{l=k+1}^{m}\frac{t^{(l)}+1}{\zeta^{(l)}\left\|\mathcal{M}^{(l)}\right\|_{\mathsf{F}}}\right)\left\|\mathcal{X}^{(m+1)}\right\|_{\mathsf{F}}
\end{aligned}
$$

The proof of Lemma F.3 is then completed by induction.

□

Now we can proove Lemma F.2

Proof. (of Lemma F.2)

The proof is similar with the proof of Lemma D.5. The only difference is we replace $t^{(l)}$ by $t^{(l)}+1$.

□

To prove Theorem F.1, the following lemma is needed.

Lemma F.4. For any convolutional neural network $\mathbb{M}$ of n layers with skip connection satisfying the assumptions in section 3 and any margin $\gamma \geq 0$, $\mathbb{M}$ can be compressed to a tensorial convolutional neural network $\hat{\mathbb{M}}$ with $\sum_{k=1}^{n}\hat{R}^{(k)}(s^{(k)}+t^{(k)}+k_x^{(k)}\times k_y^{(k)}+1)$ total parameters such that for any $\mathcal{X}\in S$, $\hat{L}_0(\hat{\mathbb{M}})\leq\hat{L}_\gamma(\mathbb{M})$. Here, for each layer k,

$$\hat{R}^{(k)}=\min\left\{j\in[R^{(k)}]|\xi_j^{(k)}\Pi_{i=k+1}^{n}(t_j^{(i)}+1)\leq\frac{\epsilon}{n}\Pi_{i=k}^{n}\zeta^{(i)}\left\|\mathcal{M}^{(i)}\right\|_{\mathsf{F}}\right\}$$

The proof of Lemma F.4 is the same with Lemma E.9. And by setting $\epsilon=\frac{\gamma}{2\max_{\mathcal{X}\in S}}$, we get the desired expression of $\hat{R}^{(k)}$ in the main theorem.

Proof. (of Theorem F.1) Similarly, let us bound the covering number of the compressed network $\hat{\mathbb{M}}$ by approximating each parameter with accuracy μ.

Covering Number Analysis for Convolutional Neural Network Let $\tilde{\mathcal{M}}$ denote the network after approximating each parameter in $\hat{\mathbb{M}}$ with accuracy μ. We use the same assumptions and notations with the proof of Theorem 4.5. And we still use $\mathcal{X}^{(k)},\mathcal{Y}^{(k)},\mathbb{M}^{(k)}$ to denote $\hat{\mathcal{X}}^{(k)},\hat{\mathcal{Y}}^{(k)},\hat{\mathbb{M}}^{(k)}$

Bound $\tau^{(k)}=\left\|\tilde{\mathcal{Y}}^{(k)}-\hat{\mathcal{Y}}^{(k)}\right\|_{\mathsf{F}}$**:**

$$
\begin{aligned}
&\left\|\tilde{\mathcal{Y}}^{(k)}-\mathcal{Y}^{(k)}\right\|_{\mathsf{F}} \\
&\quad=\left\|\tilde{\mathbb{M}}^{(k)}(\tilde{\mathcal{X}}^{(k)})+\tilde{\mathcal{X}}^{(k)}-\left(\mathbb{M}^{(k)}(\mathcal{X}^{(k)})+\mathcal{X}^{(k)}\right)\right\|_{\mathsf{F}} \\
&\quad\leq\left\|\tilde{\mathbb{M}}^{(k)}(\tilde{\mathcal{X}}^{(k)})-\mathbb{M}^{(k)}(\mathcal{X}^{(k)})\right\|_{\mathsf{F}}+\left\|\tilde{\mathcal{X}}^{(k)}-\mathcal{X}^{(k)}\right\|_{\mathsf{F}} \\
&\quad=\left\|\tilde{\mathbb{M}}^{(k)}(\tilde{\mathcal{X}}^{(k)})-\mathbb{M}^{(k)}(\mathcal{X}^{(k)})\right\|_{\mathsf{F}}+\left\|\mathsf{ReLU}\left(\tilde{\mathcal{Y}}^{(k)}\right)-\mathsf{ReLU}\left(\mathcal{Y}^{(k)}\right)\right\|_{\mathsf{F}} \\
&\quad\leq\left\|\tilde{\mathbb{M}}^{(k)}(\tilde{\mathcal{X}}^{(k)})-\mathbb{M}^{(k)}(\mathcal{X}^{(k)})\right\|_{\mathsf{F}}+\left\|\tilde{\mathcal{Y}}^{(k-1)}-\mathcal{Y}^{(k-1)}\right\|_{\mathsf{F}} \\
&\quad=\left\|\tilde{\mathbb{M}}^{(k)}(\tilde{\mathcal{X}}^{(k)})-\mathbb{M}^{(k)}(\mathcal{X}^{(k)})\right\|_{\mathsf{F}}+\tau^{(k-1)}
\end{aligned}
$$

Based on the proof of Theorem 4.5 (in Appendix D), we can easily get

$$
\begin{aligned}
&\left\|\tilde{\mathcal{Y}}^{(k)}-\mathcal{Y}^{(k)}\right\|_{\mathsf{F}} \\
&\qquad=\sum_{k=1}^{n}\sum_{i=1}^{k}\alpha^{(i)}\prod_{t=i+1}^{k}\beta^{(t)}
\end{aligned}
$$

where $\alpha^{(k)}=4HW\left\|\mathcal{X}^{(k)}\right\|_{\mathsf{F}}^{2}\left(\sum_{r}^{\hat{R}^{(k)}}(\lambda_r^{(k)})^2\hat{R}^{(k)}k_x^{(k)}k_y^{(k)}+4\sum_{r}^{\hat{R}^{(k)}}(\lambda_r^{(k)})^2(o^{(k)}+s^{(k)})\hat{R}^{(k)}+2(\hat{R}^{(k)})^2\right)\mu^2$,
and $\beta^{(k)}=2\left\|\tilde{\mathcal{M}}^{(k)}\right\|_{\mathsf{F}}^{2}$.

Since $\forall k\in[n],\left\|\mathcal{X}^{(k)}\right\|\leq\Pi_{i=k}^{n}\frac{\left\|\mathcal{X}^{(n+1)}\right\|_{\mathsf{F}}}{\zeta^{(i)}\left\|\mathcal{M}^{(i)}\right\|_{\mathsf{F}}}$, to obtain an ϵ-cover of the compressed network, we can first assume $\beta^{(k)}\geq 1\ \forall k\in[n]$. Then μ need to satisfy:

$$
\mu\leq\frac{\epsilon}{2\sqrt{HW}n^2\left\|\mathcal{X}^{(n+1)}\right\|_{\mathsf{F}}\hat{R}^{(*)}\sqrt{(\lambda^{(*)})^2k_x^{(*)}k_y^{(*)}+4(\lambda^{(*)})^2(o^{(*)}+s^{(*)})+2}(\frac{\sqrt{2}\left\|\tilde{\mathcal{M}}^{(*)}\right\|_{\mathsf{F}}}{\zeta^{(*)}\left\|\mathcal{M}^{(*)}\right\|_{\mathsf{F}}})^n}
$$

where $\hat{R}^{(*)}=\max_k r^{(k)}$ $\lambda^{(*)}=\max_{r,k}\lambda_r^{(k)}$, $s^{(*)}=\max_k s^{(k)}$, $o^{(*)}=\max_k o^{(k)}$, $k_x^{(*)}=\max_k k_x^{(k)}$, $k_y^{(*)}=\max_k k_y^{(k)}$ and $\frac{\left\|\tilde{\mathcal{M}}^{(*)}\right\|_{\mathsf{F}}}{\mu^{(*)}\left\|\mathcal{M}^{(*)}\right\|_{\mathsf{F}}}=\max_k\frac{\left\|\tilde{\mathcal{M}}^{(k)}\right\|_{\mathsf{F}}}{\mu^{(k)}\left\|\mathcal{M}^{(*)}\right\|_{\mathsf{F}}}$

So the skip connections don't change the limiting behavior of the covering number, which w.r.t to a given ϵ is $\tilde{O}(nd)$ (n is the number of layers in the given neural network, d is the number of parameters), and $\tilde{O}(d)$ for practical neural networks. Because skip connections don't need extra parameters, the neural network still has $\sum_{k=1}^{n}\hat{R}^{(k)}(s^{(k)}+t^{(k)}+k_x^{(k)}\times k_y^{(k)}+1)$ total parameters.

□

G Additional Algorithms and Algorithmic Details

Details of Step 3 in Algorithm 1. We use the alternating least squares (ALS) in the implementation of step 3, which is the 'parafac' method of the tensorly library (Kossaifi et al., 2019), to obtain the CP decomposition. Though CP decomposition is in general NP-hard, the ALS method usually converges for most tensors, with polynomial convergence rate w.r.t. the given precision of the allowed reconstruction error (Anandkumar et al., 2015, 2014a,b). In addition, step 3 obtains a CP parametrization of the weight tensor rather than recovers the true components of the weight tensor's CP decomposition. The rank in the CP decomposition is selected in step 2 and is an upper bound of the true rank of the tensor (Proposition 4.1). Thus, with the chosen rank, we can obtain a CP decomposition with a very low reconstruction error. In practice, for our cases, the CP decomposition method (ALS) used in step 3 always converges within a few iterations, with reasonable run time.

Algorithm 3 Find Best Rank for CNN (FBRC)

Input: A list of weight tensors $\{\mathcal{M}^{(k)}\}_{k=1}^n$ in the original network $\mathbb{M}$ where each $\mathcal{M}^{(k)} \in \mathbb{R}^{s^{(k)} \times o^{(k)} \times k_x^{(k)} \times k_y^{(k)}}$, a list of number of components $\{R^{(k)}\}_{k=1}^n$, a list of layer cushions $\{\zeta^{(k)}\}_{k=1}^n$ of the original network, and a perturbation parameter ϵ which denotes the maximum error we could tolerate regarding the difference between the output of original network and that of compressed network.

Output: Returns a list of number of components $\{\hat{R}^{(k)}\}_{k=1}^n$ for the compressed network such that $\left\|\mathbb{M}(\mathcal{X}) - \hat{\mathbb{M}}(\mathcal{X})\right\|_{\mathsf{F}} \leq \epsilon$. Notice that for each k, if the original network does not have skip connections, $\hat{R}^{(k)}$ satisfies that

$$\xi^{(k)}_{\hat{R}^{(k)}} \Pi_{i=k+1}^n t^{(i)}_{\hat{R}^{(k)}} \leq \frac{\epsilon}{n} \Pi_{i=k}^n \zeta^{(i)} \left\|\mathcal{M}^{(i)}\right\|_{\mathsf{F}} \tag{85}$$

or if skip connection is used, $\hat{R}^{(k)}$ satisfies that

$$\xi^{(k)}_{\hat{R}^{(k)}} \Pi_{i=k+1}^n (t^{(i)}_{\hat{R}^{(k)}} + 1) \leq \frac{\epsilon}{n} \Pi_{i=k}^n \zeta^{(i)} \left\|\mathcal{M}^{(i)}\right\|_{\mathsf{F}} \tag{86}$$

1: For each layer k, calculate the following properties: layer cushion $\zeta^{(k)}$, weight norm $\left\|\mathcal{M}^{(k)}\right\|_{\mathsf{F}}$, then calculate the RHS $\frac{\epsilon}{n} \Pi_{i=k}^n \zeta^{(i)} \left\|\mathcal{M}^{(i)}\right\|_{\mathsf{F}}$ for each k
2: Find the smallest $\hat{R}^{(n)}$ such that the tensor noise bound for the last layer $\xi^{(n)}$ satisfies $\xi^{(n)} \leq \frac{\epsilon}{n} \zeta^{(n)} \left\|\mathcal{M}^{(n)}\right\|_{\mathsf{F}}$
3: **for** $k = n-1$ to 1 **do**
4: **if** $\mathbb{M}$ does not have skip connections **then**
5: Calculate the multiplication of tensorization factor for layers upper than k, i.e., $\Pi_{i=k+1}^n t^{(i)}_{\hat{R}^{(i)}}$, based on the choices of $\hat{R}^{(i)}$ for $k \leq i \leq n$
6: Find the smallest $\hat{R}^{(k)}$ by calculating the largest possible $\xi^{(k)}$ such that Equation 85 holds.
7: **else**
8: Calculate the multiplication of tensorization factor for layers upper than k, i.e., $\Pi_{i=k+1}^n (t^{(i)}_{\hat{R}^{(k)}} + 1)$, based on the choices of $\hat{R}^{(i)}$ for $k \leq i \leq n$
9: Find the smallest $\hat{R}^{(k)}$ by calculating the largest possible $\xi^{(k)}$ such that Equation 86 holds.
10: Return $\{\hat{R}^{(k)}\}_{k=1}^n$

Remark. The FBRC algorithm finds a set of ranks that satisfies inequality 85 (CNNs) or 86 (NNs with skip connections) within polynomial time because of the following guarantees. The total number of possible sets of ranks (say T), which the FBRC algorithm will at most search through, is equal to the product of the ranks of all layers. The rank of each layer is upper bounded by Proposition 4.1 and thus T is polynomial w.r.t. the shape of the original weight tensors and the number of layers. Moreover, the search will definitely succeed as the inequalities 85 and 86 automatically hold when $\hat{R}^{(k)} = R^{(k)}$.

Algorithm 4 Find Best Rank (FBR)

Input: A list of tensors $\{\mathcal{M}^{(k)}\}_{k=1}^n$ where each $\mathcal{M}^{(k)} \in \mathbb{R}^{s_1^{(k)} \times s_2^{(k)} \times s_1^{(k+1)} \times s_2^{(k+1)}}$ is reshaped from a matrix $\boldsymbol{A^{(k)}}$, a list of number of components $\{R^{(k)}\}_{k=1}^n$, a list of layer cushions $\{\zeta^{(k)}\}_{k=1}^n$ of the original network, and a perturbation parameter ϵ which denotes the maximum error we could tolerate regarding the difference between the output of original network and that of compressed network.

Output: Returns a list of number of components $\{\hat{R}^{(k)}\}_{k=1}^n$ for the compressed network such that $\left\|\mathbb{M}(\boldsymbol{X}) - \hat{\mathbb{M}}(\boldsymbol{X})\right\|_{\mathsf{F}} \leq \epsilon$.

$$\rho^{(k)} \xi^{(k)}_{\hat{R}^{(k)}} \Pi_{i=k+1}^n t^{(i)}_{\hat{R}^{(k)}} \leq \frac{\epsilon}{n} \left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}} \Pi_{i=k}^n \zeta^{(i)}\} \tag{87}$$

1: For each layer k, calculate the following properties: reshaping factor $\rho^{(k)}$, layer cushion $\zeta^{(k)}$, weight norm $\left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}}$, then calculate the RHS $\frac{\epsilon}{n} \left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}} \Pi_{i=k}^n \zeta^{(i)}$ for each k
2: Find the smallest $\hat{R}^{(n)}$ such that the tensor noise bound for the last layer $\xi^{(n)}$ satisfies $\rho^{(n)} \xi^{(n)} \leq \frac{\epsilon}{n} \zeta^{(n)} \left\|\boldsymbol{A}^{(k)}\right\|_{\mathsf{F}}$
3: **for** $k = n-1$ to 1 **do**
4: Calculate the multiplication of tensorization factor for layers upper than k, i.e., $\Pi_{i=k+1}^n t^{(i)}_{\hat{R}^{(i)}}$, based on the choices of $\hat{R}^{(i)}$ for $k \leq i \leq n$
5: Find the smallest $\hat{R}^{(k)}$ by calculating the largest possible $\xi^{(k)}$ such that Equation 87 holds.
6: Return $\{\hat{R}^{(k)}\}_{k=1}^n$

Algorithm 5 CNN-Project

Input: A convolutional neural network $\mathbb{M}$ of n layers where its weight tensor $\mathcal{M}^{(k)}$ of the k^{th} layer is parametrized by $\{\lambda_r^{(k)}, a_r^{(k)}, b_r^{(k)}, c_r^{(k)}\}_{r=1}^{R^{(k)}}$, and a list of ranks $\{\hat{R}^{(k)}\}_{i=1}^n$.

Output: Returns a compressed network $\hat{\mathbb{M}}$ of $\mathbb{M}$ where for each layer k, $\left\|\hat{\mathcal{M}}^{(k)}\right\|$ is constructed by the top $\hat{R}^{(k)}$ components from CP components of $\mathcal{M}^{(k)}$.

for $k = 1$ to n **do**
  $\hat{\mathcal{M}}^{(k)} \leftarrow \sum_{r=1}^{\hat{R}^{(k)}} \lambda_r^{(k)} a_r^{(k)} \otimes b_r^{(k)} \otimes c_r^{(k)}$
  Let $\hat{\mathcal{M}}^{(k)}$ be the weight tensor of the k^{th} layer in $\hat{\mathbb{M}}$
Return $\hat{\mathbb{M}}$

Algorithm 6 TNN-Project

Input: A fully connected neural network $\mathbb{M}$ of n layers where its weight tensor $\mathcal{M}^{(k)}$ of the k^{th} layer is parametrized by $\{\lambda_r^{(k)}, a_r^{(k)}, b_r^{(k)}, c_r^{(k)}, d_r^{(k)}\}_{r=1}^{R^{(k)}}$, and a list of ranks $\{\hat{R}^{(k)}\}_{i=1}^n$.

Output: Returns a compressed network $\hat{\mathbb{M}}$ of $\mathbb{M}$ where for each layer k, $\left\|\hat{\mathcal{T}}^{(k)}\right\|$ is constructed by the top $\hat{R}^{(k)}$ components from CP components of $\mathcal{M}^{(k)}$.

for $k = 1$ to n **do**
  $\hat{\mathcal{M}}^{(k)} \leftarrow \sum_{r=1}^{\hat{R}^{(k)}} \lambda_r^{(k)} a_r^{(k)} \otimes b_r^{(k)} \otimes c_r^{(k)} \otimes d_r^{(k)}$
  Let $\hat{\mathcal{T}}^{(k)}$ be the weight tensor of the k^{th} layer in $\hat{\mathbb{M}}$
Return $\hat{\mathbb{M}}$