SpecFormer: Guarding Vision Transformer Robustness via Maximum Singular Value Penalization

Xixu Hu[1,2], Runkai Zheng[3], Jindong Wang[2,4](✉), Cheuk Hang Leung[1], Qi Wu[1](✉), Xing Xie[2]

[1] City University of Hong Kong, Kowloon, Hong Kong SAR
[2] Microsoft Research Asia
[3] The Chinese University of Hong Kong (Shenzhen)
[4] William & Mary
qi.wu@cityu.edu.hk, jindong.wang@microsoft.com

A General Optimization Objective for MSVP

Here we provide the most general form of our proposed **Maximum Singular Value Penalization (MSVP)**. In this context, we employ three hyperparameters to flexibly adjust the strength of the maximum singular value penalties for different attention branches. This aspect is crucial for achieving improved performance and better adaptability within the self-attention mechanism, playing distinct and non-interchangeable roles for $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V$. Denote the classification loss as $\mathcal{L}_{cls}$ such as cross-entropy loss, the overall general training objective with MSVP is:

$$\mathcal{J} = \mathcal{L}_{cls} + \mathcal{L}_{msvp} = \mathcal{L}_{cls} + \lambda_q \cdot \sigma^2_{\max}(\mathbf{W}^Q) + \lambda_k \cdot \sigma^2_{\max}(\mathbf{W}^K) + \lambda_v \cdot \sigma^2_{\max}(\mathbf{W}^V), \quad (1)$$

where λ_q, λ_k and λ_v are trade-off parameters. The original Transformer [8] adopts multi-head self-attention to jointly attend to information from different subspaces at different positions. In line with that spirit, MSVP can also be added in a multi-head manner. Incorporating the summation over all the heads and layers, the overall training objective with multi-head MSVP is:

$$\mathcal{J} = \mathcal{L}_{cls} + \mathcal{L}_{msvp} = \mathcal{L}_{cls} + \lambda_q \cdot \sum_{l=1}^{L}\sum_{h=1}^{H} \sigma^2_{\max}(\mathbf{W}_l^{Q,h}) + \lambda_k \cdot \sum_{l=1}^{L}\sum_{h=1}^{H} \sigma^2_{\max}(\mathbf{W}_l^{K,h}) + \lambda_v \cdot \sum_{l=1}^{L}\sum_{h=1}^{H} \sigma^2_{\max}(\mathbf{W}_l^{V,h}), \quad (2)$$

where $\mathbf{W}_l^{*,h}$ denotes the h^{th} head in the l^{th} attention layer, and $*$ could be Q, K or V. By restricting the maximum size of the perturbation's largest singular value, we can control the range of the output changes when attacked, leading to a model with better robustness.The computation of maximum singular values can be seamlessly integrated into the forward process.

B Proof of Theorem 2

In this section, we derive the local Lipschitz constant upper bound for the self-attention mechanism **(Attn)** around input $\mathbf{X}_0$.

Theorem 3.2 1 (Local Lipschitz Constant of self-attention layer) *The self-attention layer is local Lipschitz continuous in* $B_2(\mathbf{X}_0, \delta_0)$

$$\mathrm{Lip}_2(\mathrm{Attn}, \mathbf{X}_0) \leq N(N+1)\left(B+\delta_0\right)^2 \left[\left\|\mathbf{W}^V\right\|_2 \left\|\mathbf{W}^Q\right\|_2 \left\|\mathbf{W}^{K,\top}\right\|_2 + \left\|\mathbf{W}^V\right\|_2\right]. \tag{3}$$

Proof. To begin, we introduce some fundamental notations and re-formulate the self-attention mechanism as follows:

$$\begin{aligned} \mathrm{Attn}\left(\mathbf{X}, \mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V\right) &= \mathrm{softmax}\left(\frac{\mathbf{X}\mathbf{W}^Q\left(\mathbf{X}\mathbf{W}^K\right)^\top}{\sqrt{D}}\right)\mathbf{X}\mathbf{W}^V, \\ &= \mathbf{P}\mathbf{X}\mathbf{W}^V, \end{aligned} \tag{4}$$

where we use $\mathbf{P}$ to denote the softmax matrix for brevity. We can further express the attention mechanism as a collection of row vector functions $f_i(\mathbf{X})$:

$$f(\mathbf{X}) = \mathrm{Attn}(\mathbf{X}) = \mathbf{P}\mathbf{X}\mathbf{W}^V = \begin{bmatrix} f_1(\mathbf{X}) \\ \vdots \\ f_N(\mathbf{X}) \end{bmatrix} \in \mathbb{R}^{N\times D}, \text{ where } f_i(\mathbf{X}) \in \mathbb{R}^{1\times D}, \tag{5}$$

We denote the input data matrix $\mathbf{X}$ as a collection of row vectors and the (i,j) element of $\mathbf{P}$ as P_{ij}:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_N^\top \end{bmatrix} \in \mathbb{R}^{N\times d}, \text{ where } \mathbf{x}_i^\top \in \mathbb{R}^{1\times d},$$

Then we have the representation for the row vector function $f_i(\mathbf{X})$:

$$f_i(\mathbf{X}) = \sum_{j=1}^{N} P_{ij}\mathbf{x}_j^\top \mathbf{W}^V, \tag{6}$$

Denote the transpose of the i^{th} row of the softmax matrix $\mathbf{P}$ as $\mathbf{P}_{i:}^\top = \mathrm{softmax}(\mathbf{X}\mathbf{A}\mathbf{x}_i) \in \mathbb{R}^{N\times 1}$, where $\mathbf{A}$ stands for $\frac{\mathbf{W}^Q\mathbf{W}^{K,\top}}{\sqrt{D}}$, we can further write it as a matrix multiplication form:

$$f_i(\mathbf{X}) = \mathbf{P}_{i:}\mathbf{X}\mathbf{W}^V \in \mathbb{R}^{1\times D}, \tag{7}$$

To calculate the Jacobian of the attention mechanism, we need to introduce some preliminaries. Since f is a map from $\mathbb{R}^{N\times D}$ to $\mathbb{R}^{N\times d}$, its Jacobian is

$$\mathbf{J}_f(\mathbf{X}) = \begin{bmatrix} \mathbf{J}_{11}(\mathbf{X}) & \dots & \mathbf{J}_{1N}(\mathbf{X}) \\ \vdots & \ddots & \vdots \\ \mathbf{J}_{N1}(\mathbf{X}) & \dots & \mathbf{J}_{NN}(\mathbf{X}) \end{bmatrix} \in \mathbb{R}^{ND\times Nd}, \tag{8}$$

where $\mathbf{J}_{ij}(\mathbf{X}) = \frac{\partial f_i(\mathbf{X})}{\partial \mathbf{x}_j} \in \mathbb{R}^{D\times d}$.

Recall that

$$f_i(\mathbf{X}) = \sum_{j=1}^{N} P_{ij}\mathbf{x}_j^\top \mathbf{W}^V = \mathbf{P}_{i:}\mathbf{X}\mathbf{W}^V \in \mathbb{R}^{1\times D}, \tag{9}$$

where P_{ij} is a function that depends on $\mathbf{X}$: $\mathbf{P}_{i:}^\top = \text{softmax}(\mathbf{X}\mathbf{A}\mathbf{x}_i) \in \mathbb{R}^{N\times 1}$. Denote $\mathbf{X}\mathbf{A}\mathbf{x}_i = \mathbf{q}$ for brevity, by applying the chain rule and product rule, we obtain a commonly used result that can be applied as follows:

$$\frac{\partial \mathbf{P}_{i:}}{\partial \mathbf{q}} = \frac{\partial\, \text{softmax}(\mathbf{q})}{\partial \mathbf{q}} = \text{diag}(\mathbf{P}_{i:}) - \mathbf{P}_{i:}\mathbf{P}_{i:}^\top := \mathbf{P}^{(i)}, \tag{10}$$

By utilizing this result, we can continue to calculate the Jacobian,

$$\begin{aligned}
\frac{\partial f_i(\mathbf{X})}{\partial \mathbf{x}_j} &= \sum_{k=1}^{N} \mathbf{W}^V \left(\frac{\partial P_{ik}}{\partial \mathbf{x}_j}\mathbf{x}_k + P_{ik}\frac{\partial \mathbf{x}_k}{\partial \mathbf{x}_j} \right), \\
&= \mathbf{W}^V \sum_{k=1,k\neq j}^{N} \frac{\partial P_{ik}}{\partial \mathbf{x}_j}\mathbf{x}_k + \mathbf{W}^V \frac{\partial P_{ij}}{\partial \mathbf{x}_j}\mathbf{x}_j + \mathbf{W}^V P_{ij} I, \\
&= \mathbf{W}^V \sum_{k=1}^{N} \frac{\partial P_{ik}}{\partial \mathbf{x}_j}\mathbf{x}_k + \mathbf{W}^V P_{ij} I,
\end{aligned} \tag{11}$$

We can further write it into a matrix form for the following derivation:

$$\begin{aligned}
\frac{\partial f_i(\mathbf{X})}{\partial \mathbf{x}_j} &= \mathbf{W}^V \sum_{k=1}^{N} \frac{\partial P_{ik}}{\partial \mathbf{x}_j}\mathbf{x}_k + \mathbf{W}^V P_{ij} I = \mathbf{W}^V \left[\mathbf{x}_1 \dots \mathbf{x}_N\right] \begin{bmatrix} \frac{\partial P_{i1}}{\partial \mathbf{x}_j} \\ \vdots \\ \frac{\partial P_{iN}}{\partial \mathbf{x}_j} \end{bmatrix} + \mathbf{W}^V P_{ij} I, \\
&= \mathbf{W}^V \mathbf{X}^\top \begin{bmatrix} \frac{\partial P_{i1}}{\partial \mathbf{x}_j} \\ \vdots \\ \frac{\partial P_{iN}}{\partial \mathbf{x}_j} \end{bmatrix} + \mathbf{W}^V P_{ij} I = \mathbf{W}^V \left(\underbrace{\mathbf{X}^\top \frac{\partial \mathbf{P}_{i:}^\top}{\partial \mathbf{x}_j}}_{\text{(I)}} + \underbrace{P_{ij} I}_{\text{(II)}} \right),
\end{aligned} \tag{12}$$

By utilizing the chain rule again and with the result from equation (10), we can further decompose the first term (I) as

$$\mathbf{X}^\top \frac{\partial \mathbf{P}_{i:}^\top}{\partial \mathbf{x}_j} = \mathbf{X}^\top \frac{\partial \mathbf{P}_{i:}^\top}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial \mathbf{x}_j} = \mathbf{X}^\top \mathbf{P}^{(i)} \frac{\partial \mathbf{q}}{\partial \mathbf{x}_j}, \tag{13}$$

Recall that $\mathbf{q} = \mathbf{X}\mathbf{A}\mathbf{x}_i$, we can analyze $\frac{\partial \mathbf{q}}{\partial \mathbf{x}_j} = \frac{\partial \mathbf{X}\mathbf{A}\mathbf{x}_i}{\partial \mathbf{x}_j}$ in two cases:

– Case 1: if $i \neq j$,

$$\frac{\partial \mathbf{X}\mathbf{A}\mathbf{x}_i}{\partial \mathbf{x}_j} = \begin{bmatrix} \left(\frac{\partial \mathbf{x}_1^\top \mathbf{A}\mathbf{x}_i}{\partial \mathbf{x}_j}\right)^\top \\ \vdots \\ \left(\frac{\partial \mathbf{x}_N^\top \mathbf{A}\mathbf{x}_i}{\partial \mathbf{x}_j}\right)^\top \end{bmatrix} = \begin{bmatrix} \mathbf{0}^\top \\ .. \\ (\mathbf{A}\mathbf{x}_i)^\top \\ .. \\ \mathbf{0}^\top \end{bmatrix} = \begin{bmatrix} \mathbf{0}^\top \\ .. \\ \mathbf{x}_i^\top \mathbf{A}^\top \\ .. \\ \mathbf{0}^\top \end{bmatrix} = \begin{bmatrix} \mathbf{0}^\top \\ .. \\ \mathbf{x}_i^\top \\ .. \\ \mathbf{0}^\top \end{bmatrix} \mathbf{A}^\top = \mathbf{E}_{ji}\mathbf{X}\mathbf{A}^\top, \tag{14}$$

where $\mathbf{E}_{ji}$ represents the all-zero matrix except for the entry at (j, i), which is equal to 1.

– Case 2: if $i = j$,

$$\frac{\partial \mathbf{X}\mathbf{A}\mathbf{x}_i}{\partial \mathbf{x}_i} = \begin{bmatrix} \left(\frac{\partial \mathbf{x}_1^\top \mathbf{A}\mathbf{x}_i}{\partial \mathbf{x}_i}\right)^\top \\ \vdots \\ \left(\frac{\partial \mathbf{x}_N^\top \mathbf{A}\mathbf{x}_i}{\partial \mathbf{x}_i}\right)^\top \end{bmatrix}, \tag{15}$$

Note that for $k \neq i$, we have

$$\frac{\partial \mathbf{x}_k^\top \mathbf{A}\mathbf{x}_i}{\partial \mathbf{x}_i} = \left(\mathbf{x}_k^\top \mathbf{A}\right)^\top = \mathbf{A}^\top \mathbf{x}_k, \tag{16}$$

when $k = i$, we have

$$\frac{\partial \mathbf{x}_i^\top \mathbf{A}\mathbf{x}_i}{\partial \mathbf{x}_i} = \mathbf{A}\mathbf{x}_i + \left(\mathbf{x}_i^\top \mathbf{A}\right)^\top = \mathbf{A}\mathbf{x}_i + \mathbf{A}^\top \mathbf{x}_i, \tag{17}$$

Therefore,

$$\begin{aligned} \frac{\partial \mathbf{X}\mathbf{A}\mathbf{x}_i}{\partial \mathbf{x}_i} &= \begin{bmatrix} \left(\frac{\partial \mathbf{x}_1^\top \mathbf{A}\mathbf{x}_i}{\partial \mathbf{x}_i}\right)^\top \\ \vdots \\ \left(\frac{\partial \mathbf{x}_N^\top \mathbf{A}\mathbf{x}_i}{\partial \mathbf{x}_i}\right)^\top \end{bmatrix} = \begin{bmatrix} (\mathbf{A}^\top \mathbf{x}_1)^\top \\ .. \\ (\mathbf{A}\mathbf{x}_i + \mathbf{A}^\top \mathbf{x}_i)^\top \\ .. \\ (\mathbf{A}^\top \mathbf{x}_N)^\top \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^\top \mathbf{A}^\top \\ .. \\ \mathbf{x}_i^\top \mathbf{A} + \mathbf{x}_i^\top \mathbf{A}^\top \\ .. \\ \mathbf{x}_N^\top \mathbf{A}^\top \end{bmatrix}, \\ &= \begin{bmatrix} \mathbf{0}^\top \\ .. \\ \mathbf{x}_i^\top \mathbf{A}^\top \\ .. \\ \mathbf{0}^\top \end{bmatrix} + \mathbf{X}\mathbf{A} = \mathbf{E}_{ii}\mathbf{X}\mathbf{A}^\top + \mathbf{X}\mathbf{A}, \end{aligned} \tag{18}$$

By combining the two cases above, we can express $\frac{\partial \mathbf{q}}{\partial \mathbf{x}_j}$ using a unified formula:

$$\frac{\partial \mathbf{q}}{\partial \mathbf{x}_j} = \mathbf{E}_{ji}\mathbf{X}\mathbf{A}^\top + \mathbf{X}\mathbf{A}\delta_{ij}, \tag{19}$$

where δ_{ij} is the Kronecker delta function, which takes the value 1 when $i = j$ and 0 otherwise.

Combining equation 13, 19, we have

$$\mathbf{X}^\top \frac{\partial \mathbf{P}_{i:}^\top}{\partial \mathbf{x}_j} = \mathbf{X}^\top \frac{\partial \mathbf{P}_{i:}^\top}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial \mathbf{x}_j} = \mathbf{X}^\top \mathbf{P}^{(i)} \frac{\partial \mathbf{q}}{\partial \mathbf{x}_j} = \mathbf{X}^\top \mathbf{P}^{(i)} (\mathbf{E}_{ji}\mathbf{X}\mathbf{A}^\top + \mathbf{X}\mathbf{A}\delta_{ij}), \tag{20}$$

Substituting this result back into equation 12, we can get the (i, j) block of the Jacobian:

$$\begin{aligned}\mathbf{J}_{ij}(\mathbf{X}) = \frac{\partial f_i(\mathbf{X})}{\partial \mathbf{x}_j} &= \mathbf{W}^V \left(\mathbf{X}^\top \frac{\partial \mathbf{P}_{i:}^\top}{\partial \mathbf{x}_j} + P_{ij} I \right), \\ &= \mathbf{W}^V \left(\mathbf{X}^\top \mathbf{P}^{(i)} (\mathbf{E}_{ji} \mathbf{X} \mathbf{A}^\top + \mathbf{X} \mathbf{A} \delta_{ij}) + P_{ij} I \right), \quad (\forall 1 \leq i, j \leq N)\end{aligned} \tag{21}$$

Specifically, we can write the diagnol (i, i) block of the Jacobian as

$$\mathbf{J}_{ii}(\mathbf{X}) = \mathbf{W}^V \left(\mathbf{X}^\top \mathbf{P}^{(i)} (\mathbf{E}_{ii} \mathbf{X} \mathbf{A}^\top + \mathbf{X} \mathbf{A}) + P_{ii} I \right), \tag{22}$$

while the non-diagnoal $(i, j), j \neq i$ block can be written as

$$\mathbf{J}_{ij}(\mathbf{X}) = \mathbf{W}^V \left(\mathbf{X}^\top \mathbf{P}^{(i)} \mathbf{E}_{ij} \mathbf{X} \mathbf{A}^\top + P_{ij} I \right), \tag{23}$$

Let us denote the i^{th} block row of the Jacobian $\mathbf{J}$ as $[\mathbf{J}_{i1}, \cdots, \mathbf{J}_{iN}]$. We state the following lemma, which establishes a connection between the spectral norm of a block matrix and its block rows:

Lemma 1 (Relationship between the spectral norm of a block row and the block matrix, [4]). *Let* $\mathbf{A}$ *be a block matrix with block columns* $\mathbf{A}_1, \ldots, \mathbf{A}_N$. *Then* $\|A\|_2 \leq \sqrt{\sum_i \|\mathbf{A}_i\|_2^2}$.

Utilizing this lemma, and consider the input around $\mathbf{X}_0$:

$$B_2(\mathbf{X}_0, \delta_0) := \{\mathbf{X} : \|\mathbf{X} - \mathbf{X}_0\|_F \leq \delta_0\},$$

we can focus on derive the spectral norm of the i^{th} block row $[\mathbf{J}_{i1}, \cdots, \mathbf{J}_{iN}]$ in the neighbourhood. Considering that all inputs received by ViT are bounded (for instance, all entries of image data fall within the range of 0 to 255), and the inputs entering the Attention layer have been normalized by LayerNorm to a specific range $[0, 1]$, we can further replace the norm $\|\mathbf{X}\|$ with a constant $B = \max_{\mathbf{X} \in \mathcal{X}} \|\mathbf{X}\|_2$, where $\mathcal{X}$ represents a bounded open set in the Euclidean space $\mathbb{R}^{N \times D}$.

$$\begin{aligned}&\|[\mathbf{J}_{i1}(\mathbf{X}), \ldots, \mathbf{J}_{iN}(\mathbf{X})]\|_2 \Big|_{\mathbf{X} \in B_2(\mathbf{X}_0, \delta_0)} \\ &\leq \|\mathbf{J}_{ii}(\mathbf{X})\|_2 + \textstyle\sum_{j \neq i} \|\mathbf{J}_{ij}(\mathbf{X})\|_2 \Big|_{\mathbf{X} \in B_2(\mathbf{X}_0, \delta_0)}, \\ &= \|\mathbf{W}^V \left(\mathbf{X}^\top \mathbf{P}^{(i)} \left(\mathbf{E}_{ii} \mathbf{X} \mathbf{A}^\top + \mathbf{X} \mathbf{A} \right) + P_{ii} I \right)\|_2 + \textstyle\sum_{j \neq i} \|\mathbf{W}^V \left(\mathbf{X}^\top \mathbf{P}^{(i)} \mathbf{E}_{ij} \mathbf{X} \mathbf{A}^\top + P_{ij} I \right)\|_2 \\ &\leq \left\|\mathbf{W}^V\right\|_2 \left\|\mathbf{P}^{(i)}\right\|_2 \left(\|\mathbf{E}_{ii}\|_2 \left\|\mathbf{W}^Q\right\|_2 \left\|\mathbf{W}^{K,\top}\right\|_2 + \left\|\mathbf{W}^Q\right\|_2 \left\|\mathbf{W}^{K,\top}\right\| \right) (B + \delta_0)^2 + \left\|\mathbf{W}^V\right\|_2, \\ &\quad \textstyle\sum_{j \neq i} \left[\left\|\mathbf{W}^V\right\|_2 \left\|\mathbf{P}^{(i)}\right\|_2 \|\mathbf{E}_{ij}\|_2 \left\|\mathbf{W}^Q\right\|_2 \left\|\mathbf{W}^{K,\top}\right\|_2 + \left\|\mathbf{W}^V\right\|_2 \right] (B + \delta_0)^2,\end{aligned} \tag{24}$$

The first equality is in accordance with equation 22,23, while the second inequality arises from the Cauchy-Schwarz inequality and the boundedness of the input space $\mathcal{X}$ and the perturbation δ_0. Furthermore, by utilizing $\left\|\mathbf{P}^{(i)}\right\|_2 \leq 1$ and $\|\mathbf{E}_{ij}\|_2 \leq 1$, we can omit them in the inequality, leave with pure weight matrices,

$$\begin{aligned}
&\|[\mathbf{J}_{i1}(\mathbf{X}), \ldots, \mathbf{J}_{iN}(\mathbf{X})]\|_2 \Big|_{\mathbf{X} \in B_2(\mathbf{X}_0, \delta_0)} \\
&\leq 2\left(B+\delta_0\right)^2 \left\|\mathbf{W}^V\right\|_2 \left(\left\|\mathbf{W}^Q\right\|_2 \left\|\mathbf{W}^{K,\top}\right\|_2 + 1\right) + \\
&\quad (N-1)\left(B+\delta_0\right)^2 \left[\left\|\mathbf{W}^V\right\|_2 \left\|\mathbf{W}^Q\right\|_2 \left\|\mathbf{W}^{K,\top}\right\|_2 + \left\|\mathbf{W}^V\right\|_2\right], \\
&= (N+1)\left(B+\delta_0\right)^2 \left[\left\|\mathbf{W}^V\right\|_2 \left\|\mathbf{W}^Q\right\|_2 \left\|\mathbf{W}^{K,\top}\right\|_2 + \left\|\mathbf{W}^V\right\|_2\right],
\end{aligned} \tag{25}$$

By directly summing across the row index or utilizing the lemma 1 can yield our main theorem.

$$\|\mathbf{J}\|_2 \leq N(N+1)\left(B+\delta_0\right)^2 \left[\left\|\mathbf{W}^V\right\|_2 \left\|\mathbf{W}^Q\right\|_2 \left\|\mathbf{W}^{K,\top}\right\|_2 + \left\|\mathbf{W}^V\right\|_2\right]. \tag{26}$$

This ends the proof.

C Convergence Guarantee of Power Iteration Methods

Theorem 1 (Convergence guarantee of the power iteration method [5]). *Assuming the dominant singular value $\sigma_{max}(\mathbf{A})$ is strictly greater than the subsequent singular values and that $\boldsymbol{u}_0$ is initially selected at random, then there is a probability of 1 that $\boldsymbol{u}_0$ will have a non-zero component in the direction of the eigenvector linked with the dominant singular value. Consequently, the convergence will be geometric with a ratio of $\left|\frac{\sigma_2(\mathbf{A})}{\sigma_{max}(\mathbf{A})}\right|$.*

Proof. Denote $\sigma_1, \sigma_2, \cdots, \sigma_m$ as the m eigenvalues of matrix $\mathbf{A}^\top\mathbf{A}$, and $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m$ as the corresponding eigenvectors. Suppose σ_1 is the dominant eigenvalue, denote as $\sigma_{max} = \sigma_1$, with $|\sigma_1| > |\sigma_j|$ for $\forall j > 1$. The initial vector $\boldsymbol{u}_0$ can be written as the linear combination of the eigenvectors: $\boldsymbol{u}_0 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + \cdots + c_m\mathbf{v}_m$. If $\boldsymbol{u}_0$ is chosen randomly with uniform probability, then the eigenvector corresponding to the largest singular value has a nonzero coefficient, namely $c_1 \neq 0$. After multiplying the initial vector $\boldsymbol{u}_0$ with the matrix $\mathbf{A}$ k times, we have:

$$\begin{aligned}
\mathbf{A}^k\boldsymbol{u}_0 &= c_1\mathbf{A}^k\mathbf{v}_1 + c_2\mathbf{A}^k\mathbf{v}_2 + \cdots + c_m\mathbf{A}^k\mathbf{v}_m, \\
&= c_1\sigma_1^k\mathbf{v}_1 + c_2\sigma_2^k\mathbf{v}_2 + \cdots + c_m\sigma_m^k\mathbf{v}_m, \\
&= c_1\sigma_1^k\left(\mathbf{v}_1 + \frac{c_2}{c_1}\left(\frac{\sigma_2}{\sigma_1}\right)^k\mathbf{v}_2 + \cdots + \frac{c_m}{c_1}\left(\frac{\sigma_m}{\sigma_1}\right)^k\mathbf{v}_m\right), \\
&\to c_1\sigma_1^k\mathbf{v}_1 \quad (k \to \infty).
\end{aligned} \tag{27}$$

The second equality holds for the eigenvectors with $\mathbf{A}^k\mathbf{v}_i = \sigma_i^k\mathbf{v}_i, \forall i = 1, \cdots, m$, while the last equality is valid when $\left|\frac{\sigma_i}{\sigma_1}\right| < 1$ for all $i > 1$.

On the other hand, the vector for iterative step k can be written as $\boldsymbol{u}_k = \frac{A^k\boldsymbol{u}_0}{\|A^k\boldsymbol{u}_0\|}$. Combining these two equations above, we obtain $\boldsymbol{u}_k \to C\mathbf{v}_1$ as $k \to \infty$, where C is a constant. Therefore, we can use the power iteration method to approximate the largest singular value. The convergence is geometric, with a ratio of $\left|\frac{\sigma_2}{\sigma_1}\right|$.

D Comparison with existing bounds

In [4], it was demonstrated that the self-attention mechanism, due to the potential unboundedness of its inputs, is not globally Lipschitz continuous. Instead, they introduced an L2 self-attention mechanism based on the L2 distance $\left\|\mathbf{x}_i^\top W^Q - \mathbf{x}_j^\top W^K\right\|_2^2$. However, their proposed L2 attention mechanism also lacks global Lipschitz continuity in cases where $\mathbf{W}^K \neq \mathbf{W}^Q$, possessing global Lipschitz continuity only when $\mathbf{W}^K = \mathbf{W}^Q$. This constraint imposes significant limitations on various expressive capabilities of the model itself, as demonstrated by the experimental results presented in the original paper's Appendix L. Confining the model to $\mathbf{W}^K = \mathbf{W}^Q$ results in a substantially higher loss during convergence compared to the unconstrained model, rendering the model suboptimal.

In [1], the authors proposed a normalization approach, denoted as $g(\mathbf{X}) = \frac{\tilde{g}(\mathbf{X})}{c(\mathbf{X})}$, where $g(X)$ represents the score function in the softmax operation, to address the issue of potentially unbounded inputs, as mentioned earlier. However, their analysis is based on a simplified version of the attention mechanism, which significantly differs from the practical application of the self-attention mechanism. Additionally, through their choice of the function $c(\mathbf{X})$, it appears that the authors have implicitly assumed that the input is bounded. They defined $c(\mathbf{X})$ as follows: $c(\mathbf{X}) = \max\{\|\mathbf{Q}\|_F \left\|\mathbf{K}^\top\right\|_{(\infty,2)}, \|\mathbf{Q}\|_F \left\|\mathbf{V}^\top\right\|_{(\infty,2)}, \left\|\mathbf{K}^\top\right\|_{(\infty,2)} \left\|\mathbf{V}^\top\right\|_{(\infty,2)}\}$. Here, the input matrix $\mathbf{X}$ is represented as $\mathbf{X} = (\mathbf{Q}\|\mathbf{K}\|\mathbf{V})$. It's worth noting that this choice of $c(\mathbf{X})$ seems to assume boundedness in the input data, which may not fully address the issue of unbounded inputs in practical self-attention mechanisms.

In [6], the authors introduced a series of modules to replace components in the vanilla Vision Transformer that they deemed to introduce instability during training. These modifications include using CenterNorm to replace LayerNorm, employing scaled cosine similarity attention (SCSA) as an alternative to vanilla self-attention, and utilizing weighted residual shortcuts controlled by additional learnable parameters, along with spectral initialization for convolutions and feed-forward connections, all aimed at ensuring that the Lipschitz constant for each component remains below 1 for training stability.

However, it is noteworthy that these improvements, as presented, appear unnecessary and excessively restrictive in terms of the model's expressiveness. For instance, the authors created CenterNorm and SCSA by merely shifting the operation of dividing by the standard deviation from LayerNorm to the

attention layer, resulting in no substantial improvement. Furthermore, the ℓ_2 row normalization applied to the $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ matrices strongly impacts the overall expressiveness of the model, imposing overly stringent constraints.

Additionally, the introduction of the extra parameter alpha in the weighted residual shortcut to control the influence of the residual path on the entire pathway, while aiming for contractive Lipschitz properties, restricts α to be learned within a predefined, highly limited range or even kept fixed. These operations are cumbersome and introduce additional computational overhead.

Lastly, the proposed spectral initialization, though effective in ensuring a Lipschitz constant of 1 for convolutions and feed-forward parts, is extremely time-consuming during the initialization phase. Despite these extensive operations, the SCSA module in the authors' paper still requires **bounded norm** values for $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V$ to satisfy Lipschitz properties. Our observation posits that, in practical scenarios, due to the Lipschitz continuity of LayerNorm, each entry of the input image (e.g., pixel values ranging between 0 and 255) is inherently bounded. Moreover, the LayerNorm applied to the input before entering the attention layer constrains the input values to the $[0, 1]$ range. Therefore, when contemplating the Lipschitz continuity of the attention layer in practical terms, it suffices to consider bounded conditions. To delve further, in the context of adversarial robustness, we only need to consider pointwise Lipschitz continuity and, more specifically, the local Lipschitz continuity around a fixed point. This Lipschitz continuity can be reliably guaranteed.

E Experiments

E.1 Hyperparameter Analysis

In this section, we present the results of a hyperparameter ablation study. From Table 1, it is evident that the performance of our proposed SpecFormer remains robust even with varying hyperparameters. We recommend a tuning range of $[1e-6, 1e-3]$ for optimal results. When penalties that are too large, such as $1e-2$, can result in overly constrained representations, whereas penalties that are too small, such as $1e-7$, may have little to no effect.

E.2 Implementation Details

The experiments were conducted on a system equipped with an Intel(R) Xeon(R) CPU E5-2687W v4 @ 3.00GHz and NVIDIA TITAN RTX. The abbreviation ViT-S stands for the Vision Transformer [2] Small backbone, which comprises of 8 attention blocks with 8 heads and an embedding dimension of 768. The DeiT-Ti backbone, on the other hand, refers to the DeiT [7] Tiny model, which comprises of 12 attention blocks with 3 heads and an embedding dimension of 192. Finally, ConViT-Ti refers to the ConViT [3] Tiny model, comprising of 10 layers with 4 heads and an embedding size of 48.

Table 1: Performance (%) of SpecFormer on CIFAR10 under different hyperparameter choices for both standard training and adversarial training. The best results are highlighted in **bold**.

$(\lambda_q, \lambda_k, \lambda_v)$	CIFAR-10-Standard Training			CIFAR-10-Adversarial Training		
	Standard	PGD-2	FGSM	Standard	CW-20	PGD-20
(1e-4,0,0)	87.36	35.32	**53.91**	72.54	31.47	31.59
(0,1e-4,0)	88.58	29.02	49.81	**72.56**	31.37	31.19
(0,0,1e-4)	87.66	**36.19**	52.40	72.51	31.41	31.65
(5e-4, 7e-5 ,2e-4)	88.32	31.42	48.82	72.53	31.59	31.92
(1e-4,1e-4,1e-4)	88.14	36.07	53.71	72.46	**31.94**	32.23
(2e-4,5e-5,7e-5)	88.61	30.01	48.31	72.42	31.56	31.82
(1e-3,9e-5,3e-4)	88.15	31.75	50.46	72.43	31.91	32.23
(1e-3,1e-3,1e-3)	88.07	32.59	47.85	71.70	31.31	31.79
(5e-4,3e-4,3e-4)	**88.81**	31.28	48.32	72.19	31.69	31.89
(5e-5,3e-5,3e-5)	88.52	29.53	50.58	72.55	31.68	31.72
(2e-3,3e-4,4e-4)	88.29	33.25	48.25	71.75	31.75	**32.29**
(4e-3,8e-4,1e-3)	88.29	33.94	46.72	71.37	31.84	32.19

References

1. Dasoulas, G., Scaman, K., Virmaux, A.: Lipschitz normalization for self-attention layers with application to graph neural networks. In: International Conference on Machine Learning. pp. 2456–2466. PMLR (2021)
2. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International conference on learning representations (ICLR) (2020)
3. d'Ascoli, S., Touvron, H., Leavitt, M.L., Morcos, A.S., Biroli, G., Sagun, L.: Convit: Improving vision transformers with soft convolutional inductive biases. In: International Conference on Machine Learning. pp. 2286–2296. PMLR (2021)
4. Kim, H., Papamakarios, G., Mnih, A.: The lipschitz constant of self-attention. In: International Conference on Machine Learning. pp. 5562–5571. PMLR (2021)
5. Mises, R., Pollaczek-Geiringer, H.: Praktische verfahren der gleichungsauflösung. ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik **9**(1), 58–77 (1929)
6. Qi, X., Wang, J., Chen, Y., Shi, Y., Zhang, L.: Lipsformer: Introducing lipschitz continuity to vision transformers. In: International conference on Learning Representations (ICLR) (2023)
7. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International conference on machine learning. pp. 10347–10357. PMLR (2021)
8. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)