A. Proof of Proposition 3.3

Before we give the proof for Proposition 3.3, we first illustrate how we obtain Lemma 3.2.

Proof of Lemma 3.2 For the first module f_1 in DS-Net, the Lipschitz smoothness is represented as

$$|f_1(x) - f_1(x+\delta)| \le \sum_{j=1}^{N} w_1^j \cdot L_1^j \cdot \|\delta\|_p, \tag{16}$$

which is obtained by the definition of Lipschitz smoothness for each atomic block. By composing different layers behind together, we get

$$|f_K(x) - f_K(x+\delta)| \le \prod_{i=1}^{K} \sum_{j=1}^{N} w_i^j \cdot L_i^j \cdot \|\delta\|_p, \tag{17}$$

Therefore, the Lipschitz constant of DS-Net is decomposed as $L_f = \prod_{i=1}^{K} \sum_{j=1}^{N} w_i^j \cdot L_i^j$. Following the same decomposition process, the Lipschitz constant of the common network architecture is decomposed as $L_{\hat{f}} = \prod_{j=1}^{NK} L_j$ with the same number of network parameters.

Proof of Proposition 3.3 For the Lipschitz constant of the parameterized convolutional layers, we focus on the L_2 bounded perturbations. According to the definition of spectral norm, the Lipschitz constant of these layers is the spectral norm of its weight matrix. Mathematically, we get $L_i^j = \prod_{k=1}^{M} \|W_k\|_2$ where M is the number of convolutional layers in the current block. The spectral norm is also the maximum singular value of W_k. According to Pascanu et al. (2013), we need $\|W_k\|_2 \ge 1$ in order to prevent vanishing gradient problem during adversarial training. Therefore, we get $L_i^j \ge 1$ for all the blocks.

Comparing L_f and $L_{\hat{f}}$ then degenerates to comparing $\sum_{j=1}^{N} w_i^j \cdot L_i^j$ for DS-Net and $\prod_{j=1}^{N} L_j$ for the common network architecture since we can reorder the blocks and do not change the Lipschitz constant. And we have the following comparison results

$$\sum_{j=1}^{N} w_i^j \cdot L_i^j \le max_j L_i^j \le \prod_{j=1}^{N} L_j, \tag{18}$$

which obtained by using the fact that $L_i^j \ge 1$. Note that although the perturbation is L_2 bounded, the robustness against L_∞ can be also achieved, as stated by (Qian & Wegman, 2019).

Next, we prove DS-Net with a learnable attention weight is more robust than DS-Net with an arbitrary fixed set of attention weight. According to Cissé et al. (2017); Ma et al. (2020), the robust training objective $\ell(f(x+\delta, y, w))$ can be approximated by

$$\ell(f(x+\delta, y, w)) \approx \ell(f(x, y, w)) + L\varepsilon, \tag{19}$$

which is the standard classification loss on the natural images plus a term that is linearly correlated with the global Lipschitz constant.

Given a fixed number of network parameters, the standard classification loss function for the DS-Net with a learnable or fixed set of attention weights does not vary much. Therefore, adversarial training minimizes the global Lipschitz constant L of DS-Net implicitly. Recall that the global Lipschitz constant L is a function of the attention weight w_i^j and the Lipschitz constant L_i^j for each block, if we assume w_i^j is learnable and adversarial training leads to a global minimization of L, then changing the optimized w_i^j in DS-Net will cause the global Lipschitz constant L to increase, which validates our claim $L_f \le L'_f$.

B. Proof of Theorem 3.7

The proof of Theorem 3.7 is inspired by Shu et al. (2020).

Proof of Theorem 3.7 Following Fig. 3 in the main paper, we first explain how we obtain the bound of the block-wise Lipschitz constant for the common network architecture given a bounded block-wise Lipschitz constant of DS-Net.

To begin with, the derivative w.r.t. the parameter W_i in DS-Net is calculated as:

$$\nabla_{W_i} f = \nabla_{x_i} f x^T. \tag{20}$$

For the common network architecture, we get $\hat{x}_i = \prod_{k=1}^{i} W_k x$. Similarly, the gradient w.r.t. the parameter matrix W_i is calculated by the chain rule as

$$\begin{aligned} \nabla_{W_i}\hat{f} &= \sum_{k=i}^{N}\left(\prod_{j=i+1}^{k} W_j\right)^T \nabla_{\hat{x}_k}\hat{f}\left(\prod_{j=1}^{i-1} W_j x\right)^T \\ &= \sum_{k=i}^{N}\left(\prod_{j=i+1}^{k} W_j\right)^T \nabla_{\hat{x}_k}\hat{f} x^T\left(\prod_{j=1}^{i-1} W_j\right)^T . \end{aligned} \tag{21}$$

Using the fact that $\nabla_{\hat{x}_k}\hat{f} = \nabla_{x_k} f$, we replace $\nabla_{\hat{x}_k}\hat{f}x^T$ with $\nabla_{W_k} f$ according to Eqn. (20) and we get

$$\nabla_{W_i}\hat{f} = \sum_{k=i}^{N}\left(\prod_{j=i+1}^{k} W_j\right)^T \nabla_{W_k} f\left(\prod_{j=1}^{i-1} W_j\right)^T . \tag{22}$$

To avoid the complexity of using the standard Lipschitz constant of the smoothness for analysis, we explore and compare the block-wise Lipschitz constant (Beck & Tetruashvili, 2013) for DS-Net and the common network architecture. Specifically, we analyze for each parameter matrix W_i while fixing others. Currently, we have the block-wise Lipschitz constant bound for DS-Net, which is $\|\nabla_{W_i^1} f - \nabla_{W_i^2} f\| \leq L_i \left\|W_i^1 - W_i^2\right\|$. W_i^1, W_i^2 are any two possible assignments of W_i.

Denote λ_i as the largest eigenvalue of the parameter matrix W_i, assume we use a 2-norm for the parameter matrix W_i, then we get $\lambda_i = \|W_i\|$ where λ_i is the largest eigenvalue of $\|W_i\|$. The local smoothness w.r.t. the network parameters of the common network architectures is shown as

$$\begin{aligned} \left\|\nabla_{W_i^1}\hat{f} - \nabla_{W_i^2}\hat{f}\right\| &= \left\|\sum_{k=i}^{N}\left(\prod_{j=i+1}^{k} W_j\right)^T \left(\nabla_{W_k^1} f - \nabla_{W_k^2} f\right)\left(\prod_{j=1}^{i-1} W_j\right)^T\right\| \\ &\leq \sum_{k=i}^{N}\left\|\left(\prod_{j=i+1}^{k} W_j\right)^T \left(\nabla_{W_k^1} f - \nabla_{W_k^2} f\right)\left(\prod_{j=1}^{i-1} W_j\right)^T\right\| \\ &\leq \sum_{k=i}^{N}\left(\frac{1}{\lambda_i}\prod_{j=1}^{k}\lambda_j\right) L_k \left\|W_k^1 - W_k^2\right\| \\ &\leq \left(\prod_{j=1}^{i-1}\lambda_j\right) L_i \left\|W_i^1 - W_i^2\right\| . \end{aligned} \tag{23}$$

The first line of Eqn. (23) is obtained based on Eqn. (22) and the fact W_j is the same when we focus on the investigating the block-wise Lipschitz smoothness of W_i. The second line of Eqn. (23) is based on triangle inequality of norm. The third line is obtained by the inequality $\|WV\| \leq \|W\|\|V\|$ and the given block-wise smoothness of our DS-Net. The last line is obtained by using the fact that $W_k^1 = W_k^2$ if $k \neq i$.

Similarly, for the gradient variance bound of the common network architecture, we start from the gradient variance bound for DS-Net as follows

$$\mathbb{E}\left\|\nabla_{W_i} f - \mathbb{E}\nabla_{W_i} f\right\|^2 \leq \sigma_i^2 . \tag{24}$$

Given such a bound for DS-Net, the bound for the common network architectures is shown as follows:

$$\begin{aligned} \mathbb{E}\left\|\nabla_{W_i}\hat{f} - \mathbb{E}\nabla_{W_i}\hat{f}\right\|^2 &= \mathbb{E}\left\|\sum_{k=i}^{N}\left(\prod_{j=i+1}^{k} W_j\right)^T \left(\nabla_{W_k} f - \mathbb{E}\nabla_{W_k} f\right)\left(\prod_{j=1}^{i-1} W_j\right)^T\right\|^2 \\ &\leq N\mathbb{E}\sum_{k=i}^{N}\left\|\left(\prod_{j=i+1}^{k} W_j\right)^T \left(\nabla_{W_k} f - \mathbb{E}\nabla_{W_k} f\right)\left(\prod_{j=1}^{i-1} W_j\right)^T\right\|^2 \\ &\leq N\sum_{k=i}^{N}\left(\frac{\sigma_k}{\lambda_i}\prod_{j=1}^{k}\lambda_j\right)^2 . \end{aligned} \tag{25}$$

The first line of Eqn. (25) is obtained by using Eqn. (22). The second line of Eqn. (23) is obtained by Cauchy-Schwarz inequality. The last line is obtained based on the inequality $\|WV\| \leq \|W\|\|V\|$ and the bounded gradient variance of DS-Net.

C. Additional Results on TRADES

To further illustrate the effectiveness of DS-Net on different adversarial training styles, we test its robustness and standard accuracy on TRADES with both $\beta = 1$ and $\beta = 6$ in Tab. 5. The detailed experimental setting is described in Section 4.1 of the main paper. Tab. 5 shows a similar trend as the DS-Net trained under standard AT and MART as stated in the main paper.

Table 5. Evaluations (test accuracy) of deep models on CIFAR-10 and SVHN dataset using TRADES (Zhang et al., 2019b). ^{1} means $\beta = 1.0$ and ^{2} means $\beta = 6.0$. † means the results by our implementation. The perturbation bound ϵ is set to 0.031 for each architecture.

		CIFAR-10				
Defense Architecture	Param (M)	Natural	FGSM	PGD-20	C&W$_\infty$	AA
RobNet-large-v2^1 (Guo et al., 2020)†	33.42	87.90±0.132	57.01±0.258	49.27±0.315	49.00±0.124	46.84±0.130
WRN-34-10^1 (Zagoruyko & Komodakis, 2016)†	46.16	88.07±0.231	56.03±0.120	49.27±0.200	48.98±0.119	46.62±0.288
IE-WRN-34-10^1 (Li et al., 2020a)†	48.24	88.31±0.303	54.32±0.129	50.22±0.100	**50.37**±0.331	48.92±0.138
DS-Net-4-softmax^1(ours)	20.78	87.89±0.176	64.38±0.218	50.70±0.322	46.78±0.303	48.10±0.200
DS-Net-6-softmax^1(ours)	46.35	**88.44**±0.301	**65.00**±0.120	**52.50**±0.286	49.75±0.174	**50.00**±0.166
Improv.(%)	-	0.15%	14.02%	4.54%	-	2.21%
RobNet-large-v2^2 (Guo et al., 2020)†	33.42	81.95±0.119	60.31±0.320	53.21±0.166	50.09±0.300	50.13±0.263
WRN-34-10^2 (Zagoruyko & Komodakis, 2016)†	46.16	83.88±0.110	62.28±0.206	55.49±0.231	53.94±0.158	52.21±0.145
IE-WRN-34-10^2 (Li et al., 2020a)†	48.24	83.23±0.134	61.28±0.209	56.03±0.099	**61.72**±0.201	52.73±0.273
DS-Net-4-softmax^2(ours)	20.78	82.81±0.375	64.69±0.154	54.61±0.272	50.63±0.219	52.02±0.116
DS-Net-6-softmax^2(ours)	46.35	**83.98**±0.177	**66.56**±0.208	**56.87**±0.311	54.12±0.272	**53.33**±0.256
Improv.(%)	-	0.12%	6.87%	1.50%	-	1.14%
		SVHN				
WRN-34-10^1 (Zhang et al., 2019b)†	46.16	94.23±0.117	72.76±0.287	52.42±0.300	48.65±0.216	48.86±0.183
DS-Net-4-softmax^1(ours)	20.78	94.77±0.213	72.85±0.340	**55.69**±0.272	48.90±0.136	**51.37**±0.401
DS-Net-6-softmax^1(ours)	46.35	**95.73**±0.197	**76.61**±0.362	54.92±0.351	**49.12**±0.272	51.26±0.228
Improv.(%)	-	1.59%	5.29%	6.24%	0.97%	5.14%
WRN-34-10^2 (Zhang et al., 2019b)†	46.16	91.92±0.223	73.65±0.128	57.46±0.125	50.34±0.231	54.11±0.231
DS-Net-4-softmax^2(ours)	20.78	91.74±0.370	**73.83**±0.414	59.84±0.351	53.92±0.184	56.54±0.306
DS-Net-6-softmax^2(ours)	46.35	**92.54**±0.217	73.04±0.361	**60.54**±0.212	**54.58**±0.153	**56.75**±0.065
Improv.(%)	-	0.67%	0.24%	5.36%	8.42%	4.88%

D. Effect of Weight Decay

To demonstrate the effect of weight decay in adversarial training, we change the weight cay to 3e-4, 4e-4, 6e-4 and 7e-4 and report the average performance in terms of robustness and generalization ability for DS-Net. We conduct experiments using standard AT on CIFAR-10 with factor $k = 4$, which are shown in Tab. 6. The results demonstrate the importance of weight decay in adversarial training (align well with the empirical findings in Pang et al. (2021)), which should be carefully selected.

Table 6. Model robustness and generalization ability with different weight decay. The perturbation bound for evaluation is set to 0.031.

Weight decay	3e-4	4e-4	5e-4	6e-4	7e-4
PGD-20 Acc. (%)	48.28	49.69	**54.14**	52.67	49.69
Standard Acc. (%)	82.50	85.00	**85.39**	84.06	83.75

E. Comparison with using different optimizers for DS-Net

SGD is commonly used in AT literature. We tried other optimizers such as Adam, RMSprop, Adadelta and Adagrad. The PGD-20 accuracy is listed in Tab. 7 for DS-Net-4-softmax on CIFAR-10 (vs. 54.14% by SGD).

Table 7. PGD-20 accuracy of DS-Net trained with different optimizers for block parameters.

Optimizer	Adam	Adadelta	Adagrad	RMSprop
PGD-20 Acc.	40.76%	41.12%	39.28%	37.46%