Harnessing Edge Information for Improved Robustness in Vision Transformers

Yanxi Li, Chengbin Du, Chang Xu

School of Computer Science, Faculty of Engineering, The University of Sydney, Australia

Motivation

Previous studies hypothesize that the vulnerability of DNNs might stem from the fact that high-accuracy DNNs heavily rely on irrelevant and non-robust features, such as textures and the background. In this work, we reveal that edge information extracted from images can provide relevant and robust features related to shapes and the foreground. These features assist pretrained DNNs in achieving improved adversarial robustness without compromising their accuracy on clean images. A lightweight and plug-and-play EdgeNet is proposed, which can be seamlessly integrated into existing pretrained DNNs.

Preliminary

The common supervised training objective of vision transformers can be written as:

$$\mathcal{L}_{\text{ACC}}(f;\mathcal{D}) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell_{\text{CE}}(f(\boldsymbol{x}),y)].$$

Adversarial training is a common method to improve adversarial robustness, which can be formulated as a min-max problem:

$$\mathcal{L}_{\text{ROB}}(f;\mathcal{D}) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\max_{\boldsymbol{x}'\in\mathcal{B}_p(x,\varepsilon)} \ell_{\text{CE}}(f(\boldsymbol{x}'),y)\right],$$

where $\mathcal{B}_p(\boldsymbol{x},\varepsilon) = \{\boldsymbol{x}': \|\boldsymbol{x}-\boldsymbol{x}'\|_p \le \varepsilon\}$ is a l_p ball.

EdgeNet Building Blocks

We implement a "sandwich" architecture for each building block in our EdgeNet framework. We add zero convolutions Z(·) (Zhang and Agrawala 2023) to both the input and output of each block. Nestled between the two zero convolutions, we place a ViT block T(·) with randomized initialization, maintaining the same architecture to those found in the backbone:

$$\boldsymbol{e}_l = \mathcal{Z}^{(l)}_{\textbf{out}}\left(\mathcal{T}^{(l)}\left(\mathcal{Z}^{(l)}_{\textbf{in}}\left(\boldsymbol{e}_{l-1}\right)\right)\right). \qquad (6)$$

The zero convolution at input functions as a filter for extracting information related to the optimization objective. The zero convolution at output functions as a filter for determining information to be integrated into the backbone.

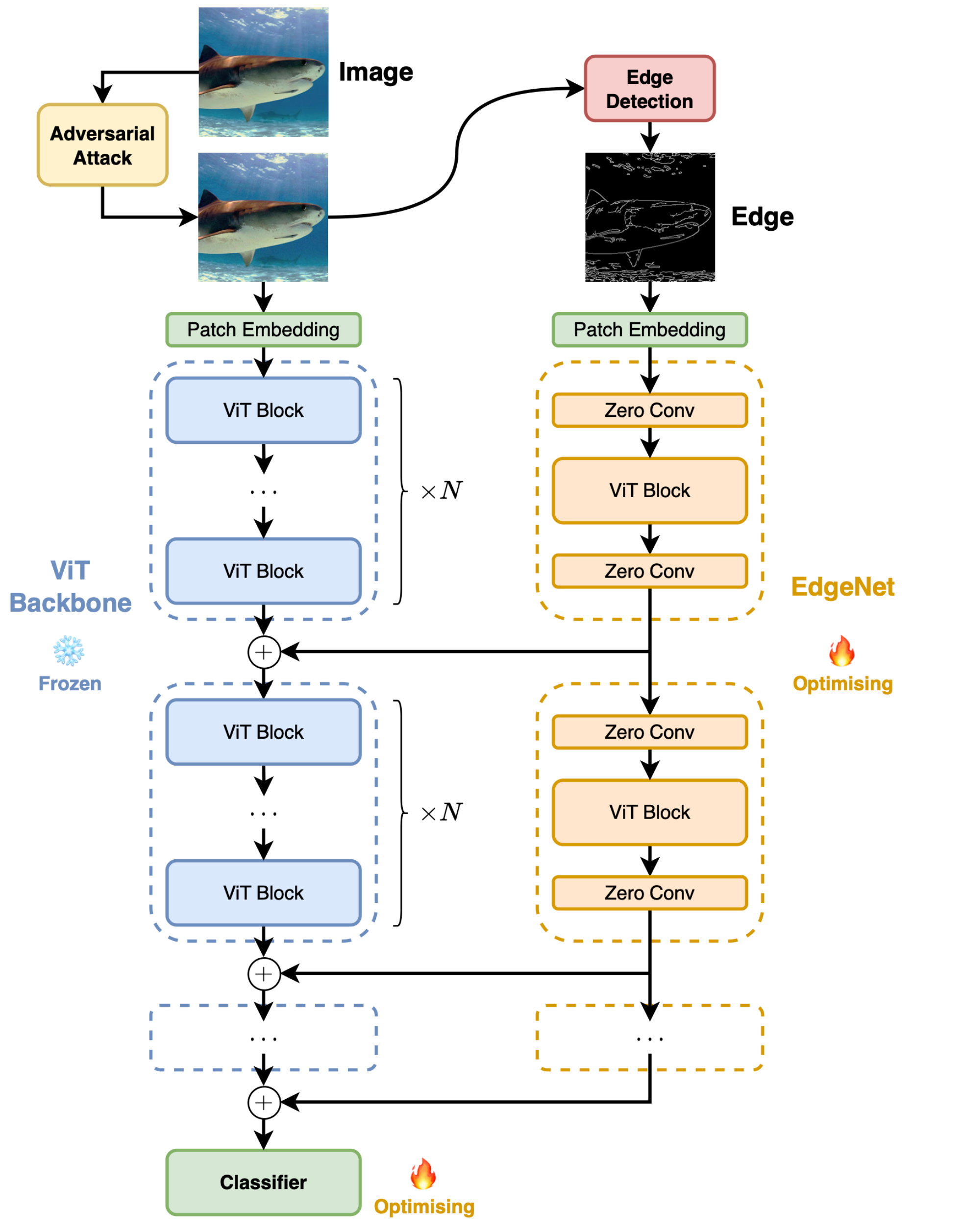
Joint Optimization

We adopt a joint optimization objective:

$$\min_f \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\Big[\alpha\cdot\ell\left(f(\boldsymbol{x},\text{Edge}(\boldsymbol{x})),y\right) + \beta\cdot\max_{\boldsymbol{x}'\in B_p(\boldsymbol{x},\varepsilon)}\ell\left(f(\boldsymbol{x}',\text{Edge}(\boldsymbol{x}')),y\right)\Big], \qquad (9)$$

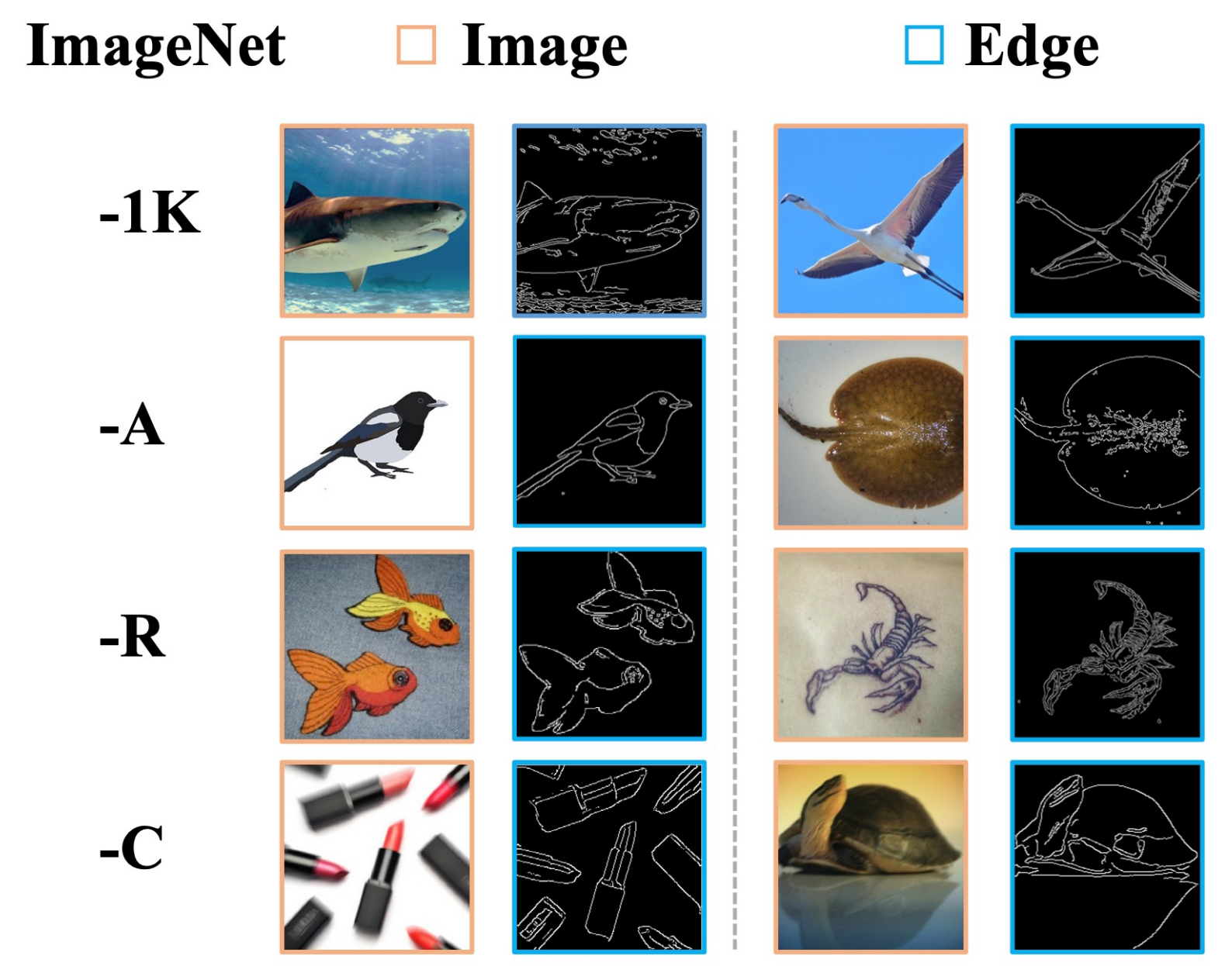
where α is the weight for accuracy and β is the weight for robustness. The cross-entropy loss is used for l(·,·). Through the adjusting of α and β, we can fine-tune our EdgeNet in a manner that enhances its robustness, meanwhile ensuring that the accuracy won't drop significantly.

EdgeNet Architecture



The architecture of EdgeNet with ViT as the backbone. An interval of N is employed, signifying the addition of one EdgeNet block for every N×ViT blocks. The output of each EdgeNet block is integrated into the intermediate layer of the ViT backbone through element-wise addition. Throughout the optimization process, the backbone remains frozen, while the EdgeNet and classification head undergoes training.

Visualization



Instances selected from ImageNet-1K, -A, -R, and -C, accompanied by their respective edges extracted by the Canny edge detector.

Scales of EdgeNet

# Intervals	# New Blocks	FLOPs (G)	Params (M)	Throughput (Images/Sec)	Clean	Attacks		ImageNet Variants		
						FGSM	PGD	A	R	C (↓)
1	12	37.88	186.14	375.16	83.4	69.0	48.0	39.5	56.8	**34.3**
3	4	24.37	119.99	543.40	**83.7**	**69.8**	**48.8**	**39.6**	56.9	34.4
6	2	21.00	103.45	601.64	83.3	66.8	46.3	37.6	**57.2**	35.0
-	0	17.60	88.1	635.81	80.2	41.1	15.5	22.1	42.0	56.9

Table 1: The performance of EdgeNet across varying scales. The "# Intervals" determines the frequency of adding a new block in relation to existing ones, while "# New Blocks" denotes the total number of added blocks. We also include results achieved by fine-tuning the classification head of the backbone for comparison (the last row).

White-box Attacks

Categories	Models	Clean	Attacks		ImageNet Variants		
			FGSM	PGD	A	R	C (↓)
CNNs	ResNet-50 (He et al. 2016)	76.1	12.2	0.9	0.0	36.1	76.7
	ResNeXt50-32x4d (Xie et al. 2017)	79.8	34.7	13.5	10.7	41.5	64.7
	EfficientNet-B4 (Tan and Le 2019)	83.0	44.6	18.5	26.3	47.1	71.1
	ConvNeXt-B (Liu et al. 2022)	83.8	-	-	36.7	51.3	46.8
Robust CNNs	ANT (Rusak et al. 2020)	76.1	17.8	3.1	1.1	39.0	63.0
	AugMix (Hendrycks et al. 2019)	77.5	20.2	3.8	3.8	41.0	65.3
	Debiased CNN (Li et al. 2020)	76.9	20.4	5.5	3.5	40.8	67.5
	DeepAugment (Hendrycks et al. 2021a)	75.8	27.1	9.5	3.9	46.7	53.6
	Anti-Aliased CNN (Zhang 2019)	79.3	32.9	13.5	8.2	41.1	68.1
ViTs	ViT-B/16 (Dosovitskiy et al. 2020)	72.8	-	-	8.0	27.1	74.8
	ViT-B/16 + CutMix (Dosovitskiy et al. 2020)	75.5	-	-	14.8	28.5	64.1
	ViT-B/16 + MixUp (Dosovitskiy et al. 2020)	77.8	-	-	12.2	34.9	61.8
	ViT-B/16 + AugReg (Steiner et al. 2021)	79.9	-	-	17.5	38.2	52.5
	ViT-B/16-384 + AugReg (Steiner et al. 2021)	81.4	-	-	26.2	38.2	58.2
	PVT-Large (Wang et al. 2021)	81.7	33.1	7.3	26.6	42.7	59.8
	ConViT-B (d'Ascoli et al. 2021)	82.4	45.4	20.8	29.0	48.4	46.9
	DeiT-B/16 (Touvron et al. 2021)	82.0	46.4	21.3	27.4	44.9	48.5
	T2T-ViT_t-24 (Yuan et al. 2021)	82.6	46.7	17.5	28.9	47.9	48.0
	Swin-B (Liu et al. 2021)	83.4	49.2	21.3	35.8	46.6	54.4
	PiT-B (Heo et al. 2021)	82.4	49.3	23.7	33.9	43.7	48.2
Robust ViTs	PyramidAT (Herrmann et al. 2022)	81.7	-	-	23.0	47.7	45.0
	PyramidAT-384 (Herrmann et al. 2022)	83.3	-	-	36.4	46.7	47.8
	RVT-B (Mao et al. 2022)	82.5	52.3	27.4	27.7	48.2	47.3
	RVT-B* (Mao et al. 2022)	82.7	53.0	29.9	28.5	48.7	46.8
	MAE-ViT-B (He et al. 2022)	83.6	-	-	35.9	48.3	51.7
	FAN-L-ViT (Zhou et al. 2022)	83.9	-	-	34.2	53.1	43.3
Robust Fine-tuning	TORA-ViT-B/16 ($\lambda = 0.1$) (Li and Xu 2023)	84.1	48.4	23.3	46.5	57.6	31.7
	TORA-ViT-B/16 ($\lambda = 0.5$) (Li and Xu 2023)	83.7	54.7	38.0	39.2	56.3	34.4
	TORA-ViT-B/16 ($\lambda = 0.9$) (Li and Xu 2023)	80.3	74.2	57.5	22.2	53.7	41.6
	EdgeNet-ViT-B/16 **(Ours)**	83.7	69.8	48.8	39.6	56.9	34.4

Table 2: Evaluation of SOTA methods on ImageNet-1K and its variants (A, R and C). The top-1 accuracy is used to assess performance on clean ImageNet-1K, under adversarial attacks (FGSM and PGD), on ImageNet-A, and -R. In the case of ImageNet-C, the focus is on the mean Corruption Error (mCE), where lower values indicate better performance (marked by ↓). "ViT-B/16-384 + AugReg" and "PyramidAT-384" employ input dimensions of 384 × 384 inputs, while the remaining models utilize input dimensions of 224 × 224.

Black-box Attacks

Source Model	Defense Model	Valid Acc. (%)	
		FGSM	PGD
ViT-B/16	ViT-B/16	35.03	14.26
ViT-B/16	EdgeNet-ViT-B/16	74.41	70.32
ViT-S/16	ViT-B/16	74.09	75.59
ViT-S/16	EdgeNet-ViT-B/16	79.34	80.09
ViT-L/16	ViT-B/16	78.31	77.29
ViT-L/16	EdgeNet-ViT-B/16	80.62	80.18
Swin-B	ViT-B/16	82.94	82.40
Swin-B	EdgeNet-ViT-B/16	83.24	82.96

Table 3: The validation accuracy under black-box attacks on ImageNet-1K. Using ViT-B/16 as both source model and defense model is equivalent to a white-box attack, included here solely for the purpose of comparison.

Image or Edge

Input	Clean	Attacks		ImageNet Variants		
		FGSM	PGD	A	R	C (↓)
Image	82.7	64.4	47.0	32.2	56.1	37.2
Edge	83.7	69.8	48.8	39.6	56.9	34.4

Table 4: The performance of integrating image or edge information into the backbone.