KCon

归源 · 智变

Returning to the source · Intelligent transformation

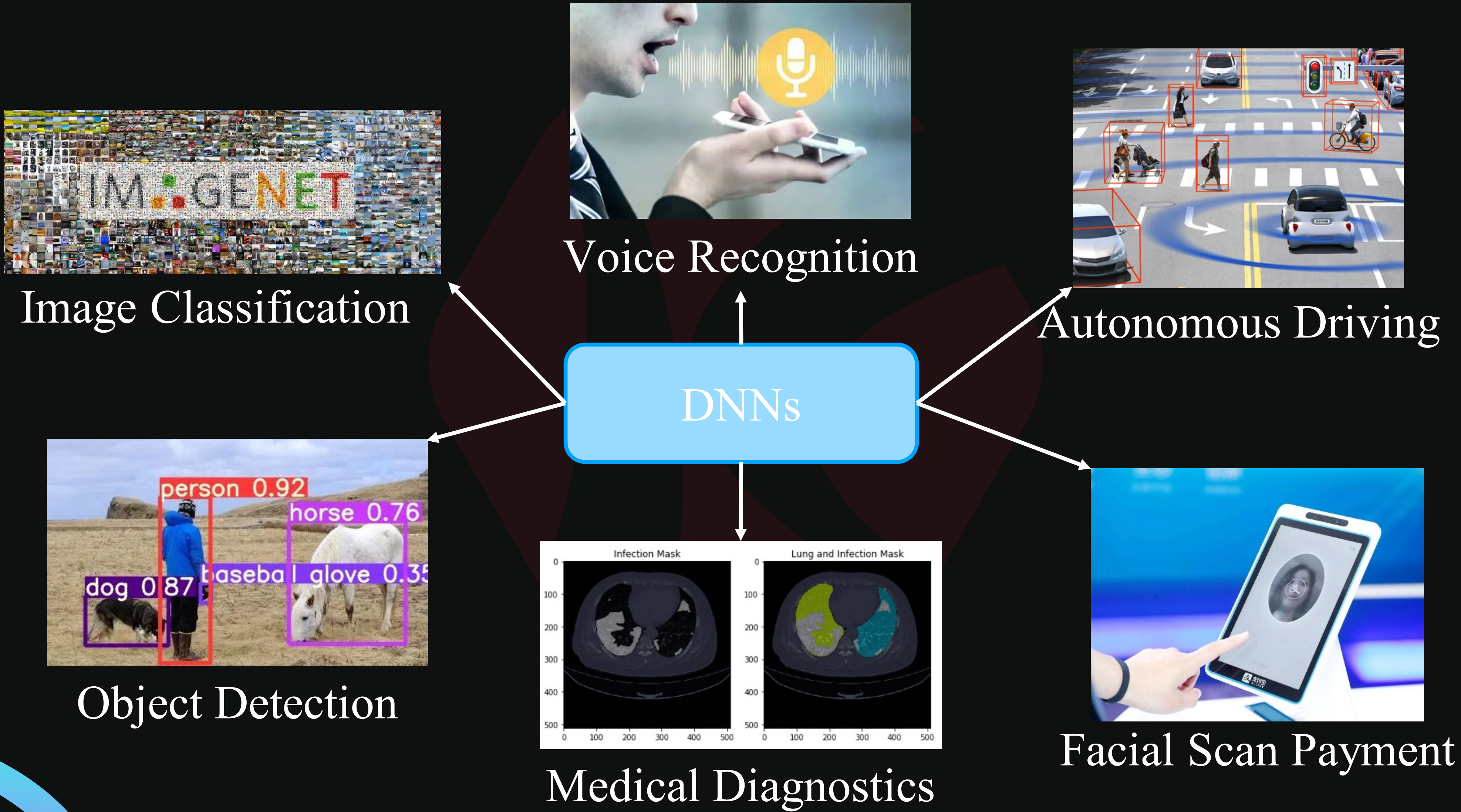
Recent Progresses in Transfer-based Attack for Image Recognition

Xiaosen Wang
Huawei Singularity Security Lab

CONTENTS

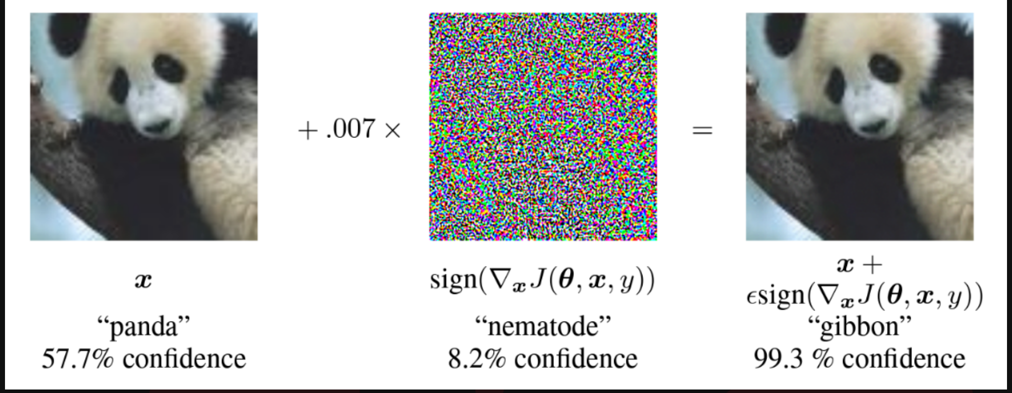
Preliminaries

- DNNs are everywhere in our life!

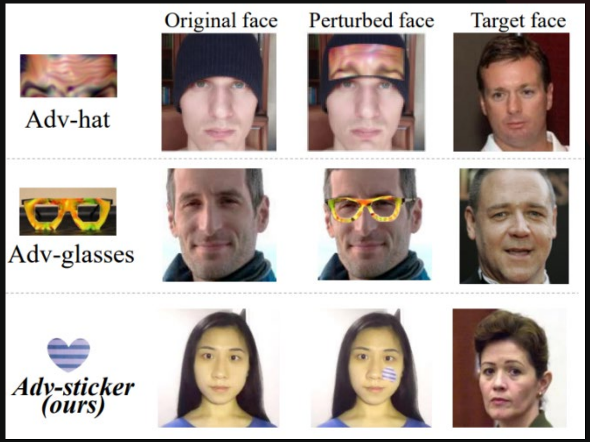


Preliminaries

- **Adversarial examples** are **indistinguishable** from **legitimate ones** by adding small perturbations, but lead to **incorrect model prediction**.



- Adversarial examples bring a huge threats to AI applications.



Goodfellow et al. Explaining and Harnessing Adversarial Examples. ICLR 2015.
Wei et al. Adversarial Sticker: A Stealthy Attack Method in the Physical World. TPAMI 2022.
Eykholt et al. Robust Physical-World Attacks on Deep Learning Visual Classification . CVPR 2018.

Preliminaries

- **How to generate Adversarial examples?**

Training a Network:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} J(x,y;\theta).$$

Generating Adversarial Example:

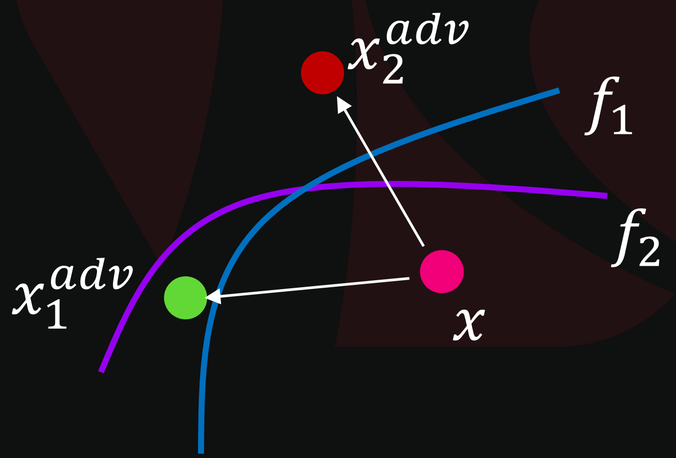
$$\max_{||x-x^{adv}||<\epsilon} J\left(x^{adv},y;\theta\right).$$

$\mathcal{D}$:	Training dataset
$J(\cdot)$:	Loss function
x:	Clean input
y:	Ground-truth label
x^{adv}:	Adversarial example

- **Untargeted attack:** The victim model predicts the generated adversarial example into *any incorrect categories*.
- **Targeted attack:** The victim model predicts the generated adversarial example into *a specific category*.

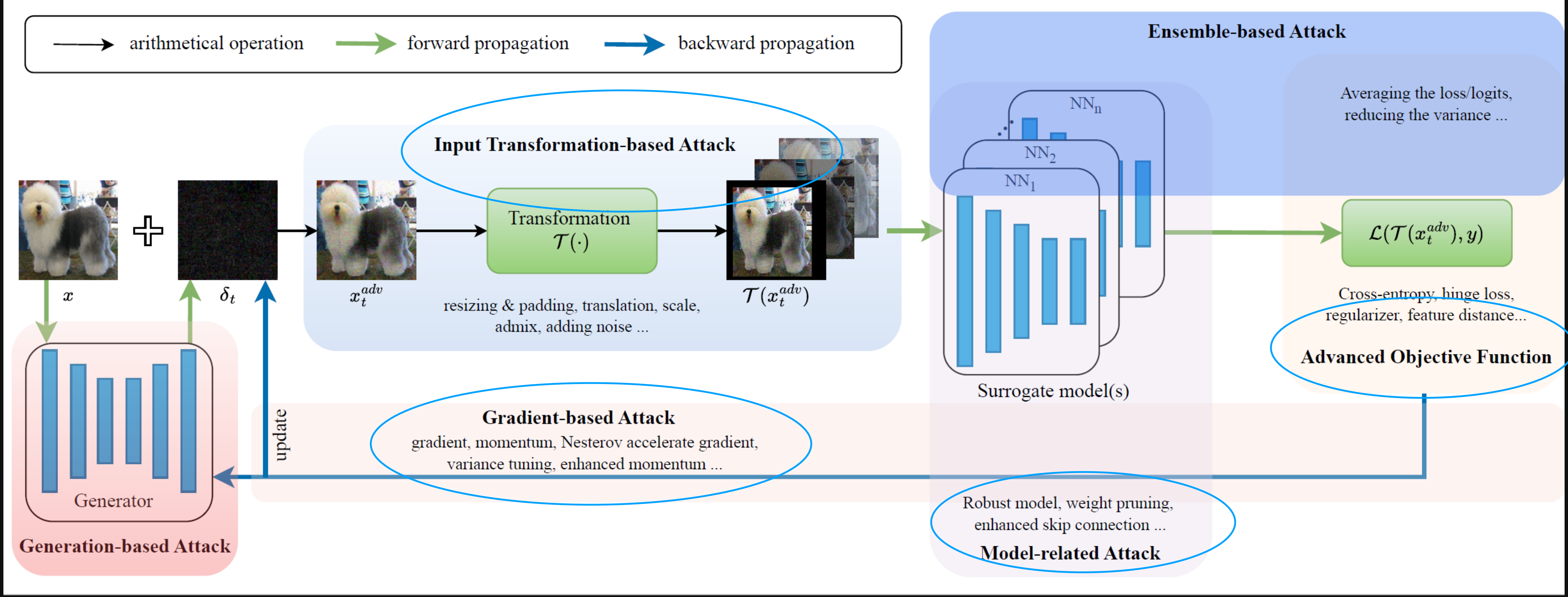
Preliminaries

- **White-box Attack**: The attacker could access any information of victim model, *e.g.*, architecture, weights, gradients, *etc*.
- **Black-box Attack**: The attacker could access limited information of victim model.
  - **Score-based Attack**: The attacker could obtain the prediction probability.
  - **Decision-based Attack**: The attacker could obtain the prediction label.
  - **Transfer-based Attack**: The adversarial examples generated on one model could mislead other victim models.



Preliminaries

- **Transfer-based Attacks**



Wang et al. Towards Boosting Adversarial Transferability on Image Classification: A Survey. To be released.

CONTENTS

Gradient-based Attacks

- **Gradient-based adversarial attacks are widely investigated:**
  - FGSM [Goodfellow et al., 2015]:

$$x^{adv} = x + \epsilon \cdot \mathrm{sign}\big(\nabla_x J(x, y; \theta)\big)$$

  - I-FGSM [Kurakin et al., 2018]:

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \mathrm{sign}\left(\nabla_x J\left(x_t^{adv}, y; \theta\right)\right)$$

  - MI-FGSM [Dong et al., 2018]:

$$g_{t+1} = \mu g_t + \frac{\nabla_x J\left(x_t^{adv}, y; \theta\right)}{||\nabla_x J\left(x_t^{adv}, y; \theta\right)||_1},\ x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \mathrm{sign}(g_{t+1})$$

  - NI-FGSM [Lin et al., 2020]: $\bar{x}_t^{adv} = x_t^{adv} + \alpha \cdot \mu \cdot g_t$

$$g_{t+1} = \mu g_t + \frac{\nabla_x J\left(\bar{x}_t^{adv}, y; \theta\right)}{||\nabla_x J\left(\bar{x}_t^{adv}, y; \theta\right)||_1},\ x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \mathrm{sign}(g_{t+1})$$

Goodfellow et al. Explaining and Harnessing Adversarial Examples. ICLR 2015.
Kurakin et al. Adversarial Examples in the Physical World. ICLR Workshop 2018.
Dong et al. Boosting Adversarial Attacks with Momentum. CVPR 2018.
Lin et al. Nesterov Accelerated Gradient and Scale Invariance for Adversarial Attacks. ICLR 2020.

Gradient-based Attacks

- **Variance Tuning (VT)**

Adversarial Example Generation		Standard Model Training
Input image x	$\Leftrightarrow$	Parameters
Transferability	$\Leftrightarrow$	Generalization

NI-FGSM finds that Nestorve Accelerated Gradient (NAG) that **accelerates the convergence** of optimization process, also **enhances the transferability**.

We treat the iterative gradient-based adversarial attack as **SGD optimization process,** in which at each iteration, the attacker always chooses the target model for update.

SGD introduces variance due to randomness.

Wang et al. Enhancing the Transferability of Adversarial Attacks through Variance Tuning. CVPR 2021.

Gradient-based Attacks

- **Variance Tuning (VT)**

Gradient Variance. Given a classifier f with parameters θ and loss function $J(x, y; \theta)$, an arbitrary image x and upper bound ϵ' for the neighborhood, the gradient variance can be defined as:

$$V_{\epsilon'}^{g}(\mathrm{x}) = \mathbb{E}_{|x'-x|_p<\epsilon'}\left[\nabla_{x'} J(x', y; \theta)\right] - \nabla_x J(x, y; \theta).$$

In practice, we approximate the gradient variance by **sampling N examples in the neighborhood of $\boldsymbol{x}$**:

$$V(x) = \frac{1}{N}\sum_{i=1}^{N} \nabla_{x^i} J\left(x^i, y; \theta\right) - \nabla_x J(x, y; \theta),$$

where $x^i = x + U[-(\beta\cdot\epsilon)^d, (\beta\cdot\epsilon)^d]$.

At t-th iteration, we **tune the gradient of $\boldsymbol{x_t^{adv}}$ with the gradient variance at (t-1)-th iteration** to stabilize the update direction.

Wang et al. Enhancing the Transferability of Adversarial Attacks through Variance Tuning. CVPR 2021.

Gradient-based Attacks

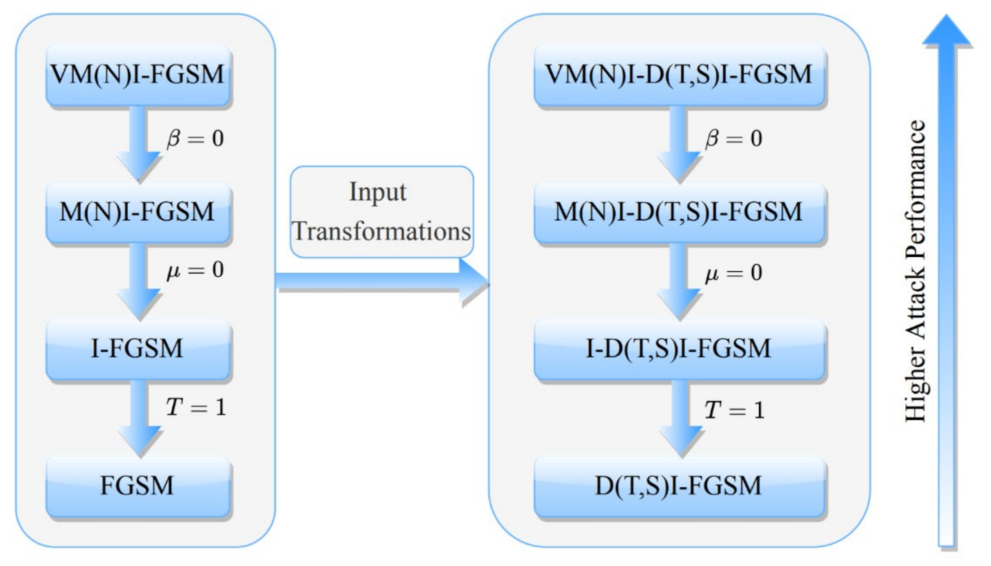
- **Variance Tuning (VT)**

The variance tuning is generally applicable to all iterative gradient based attacks.

VMI-FGSM:

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x_t^{adv}} J(x_t^{adv}, y; \theta) + \boldsymbol{V(x_{t-1}^{adv})}}{||\nabla_{x_t^{adv}} J(x_t^{adv}, y; \theta) + \boldsymbol{V(x_{t-1}^{adv})}||_1},$$

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1})$$



Wang et al. Enhancing the Transferability of Adversarial Attacks through Variance Tuning. CVPR 2021.

Gradient-based Attacks

- **Variance Tuning (VT)**

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
Inc-v3	MI-FGSM	**100.0***	43.6	42.4	35.7	13.1	12.8	6.2
	VMI-FGSM	**100.0***	**71.7**	**68.1**	**60.2**	**32.8**	**31.2**	**17.5**
	NI-FGSM	**100.0***	51.7	50.3	41.3	13.5	13.2	6.0
	VNI-FGSM	**100.0***	**76.5**	**74.9**	**66.0**	**35.0**	**32.8**	**18.8**
Inc-v4	MI-FGSM	56.3	99.7*	46.6	41.0	16.3	14.8	7.5
	VMI-FGSM	**77.9**	**99.8***	**71.2**	**62.2**	**38.2**	**38.7**	**23.2**
	NI-FGSM	63.1	**100.0***	51.8	45.8	15.4	13.6	6.7
	VNI-FGSM	**83.4**	99.9*	**76.1**	**66.9**	**40.0**	**37.7**	**24.5**
IncRes-v2	MI-FGSM	60.7	51.1	**97.9***	46.8	21.2	16.0	11.9
	VMI-FGSM	**77.9**	**72.1**	**97.9***	**67.7**	**46.4**	**40.8**	**34.4**
	NI-FGSM	62.8	54.7	**99.1***	46.0	20.0	15.1	9.6
	VNI-FGSM	**80.8**	**76.9**	98.5*	**69.8**	**47.9**	**40.3**	**34.2**
Res-101	MI-FGSM	58.1	51.6	50.5	**99.3***	23.9	21.5	12.7
	VMI-FGSM	**75.1**	**68.9**	**70.5**	99.2*	**45.2**	**41.4**	**30.1**
	NI-FGSM	65.6	58.3	57.0	99.4*	24.5	21.4	11.7
	VNI-FGSM	**79.8**	**74.6**	**73.2**	**99.7***	**46.1**	**42.5**	**32.1**

Table 1: The success rates (%) on seven models in the single model setting by various gradient-based iterative attacks. The adversarial examples are crafted on Inc-v3, Inc-v4, IncRes-v2, and Res-101 respectively. * indicates the white-box model.

Wang et al. Enhancing the Transferability of Adversarial Attacks through Variance Tuning. CVPR 2021.

CONTENTS

Input Transformation-based Attacks

- **Similar to data augmentation in training, input transformation can enhance the diversity of image, thus boosting adversarial transferability.**
  - **DIM** [Xie et al., 2019]: Randomly resize the image and add padding for gradient calculation.
  - **TIM** [Dong et al., 2019]: Accumulate the gradient on a set of translated images. To approximate this process, TIM convolves the gradient of original image with a predefined kernel.
  - **SIM** [Lin et al., 2020]: Accumulate the gradient on a set of scaled images.
  - **Admix** [Wang et al., 2021]: Mixup the image with the images from other categories for gradient calculation.
  - **SSA** [Long et al., 2022]: Add noise and randomly mask the elements in the frequency domain to generate several images for gradient calculation.

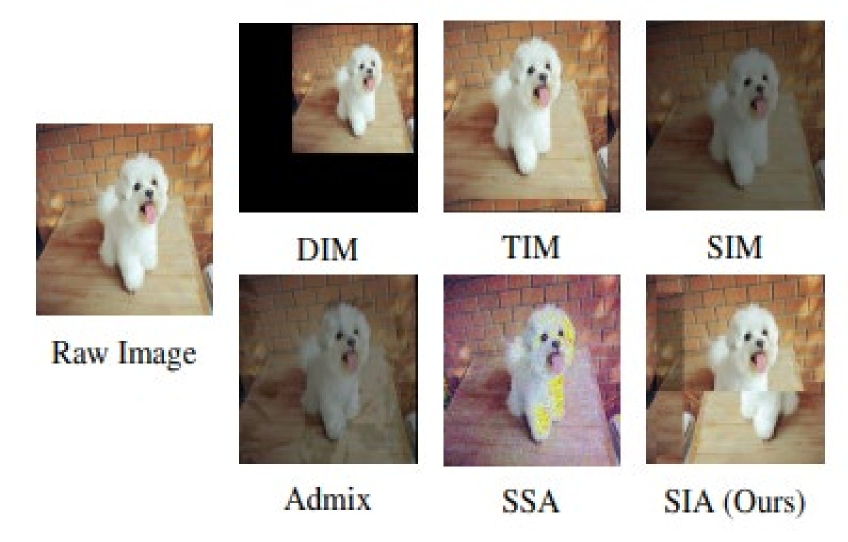
Xie et al. Improving Transferability of Adversarial Examples with Input Diversity. CVPR 2019.
Dong et al. Evading Defenses to Transferable Adversarial Examples by Translation-Invariant Attacks. CVPR 2019.
Lin et al. Nesterov Accelerated Gradient and Scale Invariance for Adversarial Attacks. ICLR 2020.
Wang et al. Admix: Enhancing the Transferability of Adversarial Attacks. ICCV 2021.
Long et al. Frequency Domain Model Augmentation for Adversarial Attack. ECCV 2022.

Input Transformation-based Attacks

- **Structure Invariant Attack (SIA)**

Assumption: The **more diverse** the transformed images are, the **better transferability** the adversarial examples have.

$$\mathrm{LPIPS}(x, \hat{x}) = \frac{1}{H \times W} \sum_{l} \sum_{h,w} ||z^{l}_{h,w} - \hat{z}^{l}_{h,w}||_2$$



	TIM	DIM	SIM	SSA	*Admix*
Transferability	57.4	77.6	79.3	80.6	83.6
LPIPS	0.25	0.43	0.48	0.54	0.73

Table 1: The transferability of TIM, DIM, SIM, *Admix*, SSA, and similarity between 1,000 images and the transformed images evaluated by LPIPS. The transferability is evaluated by the attack success rate of Inception-v3 on the adversarial examples generated on ResNet-18 .

Wang et al. Structure Invariant Transformation for better Adversarial Transferability. ICCV 2023.

Input Transformation-based Attacks

- **Structure Invariant Attack (SIA)**

Structure of Image: Given an image x, which is randomly split into $s \times s$ blocks, the relative relation between each anchor point is the structure of image, where the anchor point is the center of the image block.

Figure 2: The randomly sampled raw image and its transformed images by scaling on the full image and 3×3 blocks.

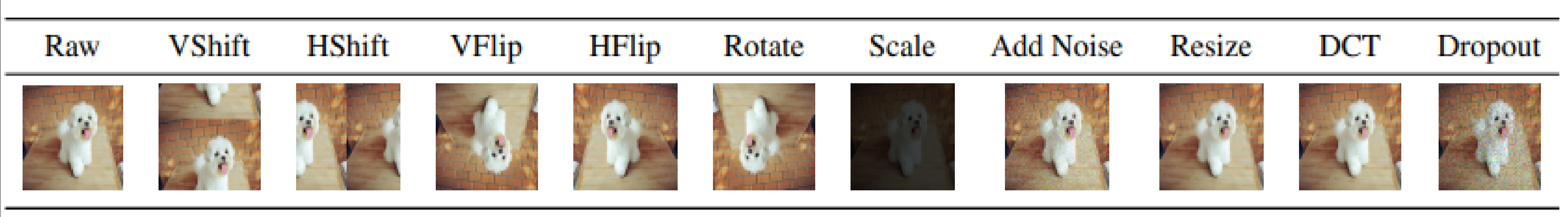
The structure of image depicts important semantic information for human recognition. Scaling the image blocks with various factors does not change the structure of image so that the generated image can be correctly recognized by humans as well as deep models.

Wang et al. Structure Invariant Transformation for better Adversarial Transferability. ICCV 2023.

Input Transformation-based Attacks

- **Structure Invariant Attack (SIA)**

To improve the diversity and maintain the semantic information, we apply **various image transformations** to **different image blocks**, denoted as Structure Invariant Transformation (SIT).



- The proposed transformation significantly improves the diversity but maintains the structure invariance.
- The proposed transformation can be integrated into existing gradient-based methods.
- The gradient is computed on several transformed images.

Wang et al. Structure Invariant Transformation for better Adversarial Transferability. ICCV 2023.

Input Transformation-based Attacks

- **Structure Invariant Attack (SIA)**

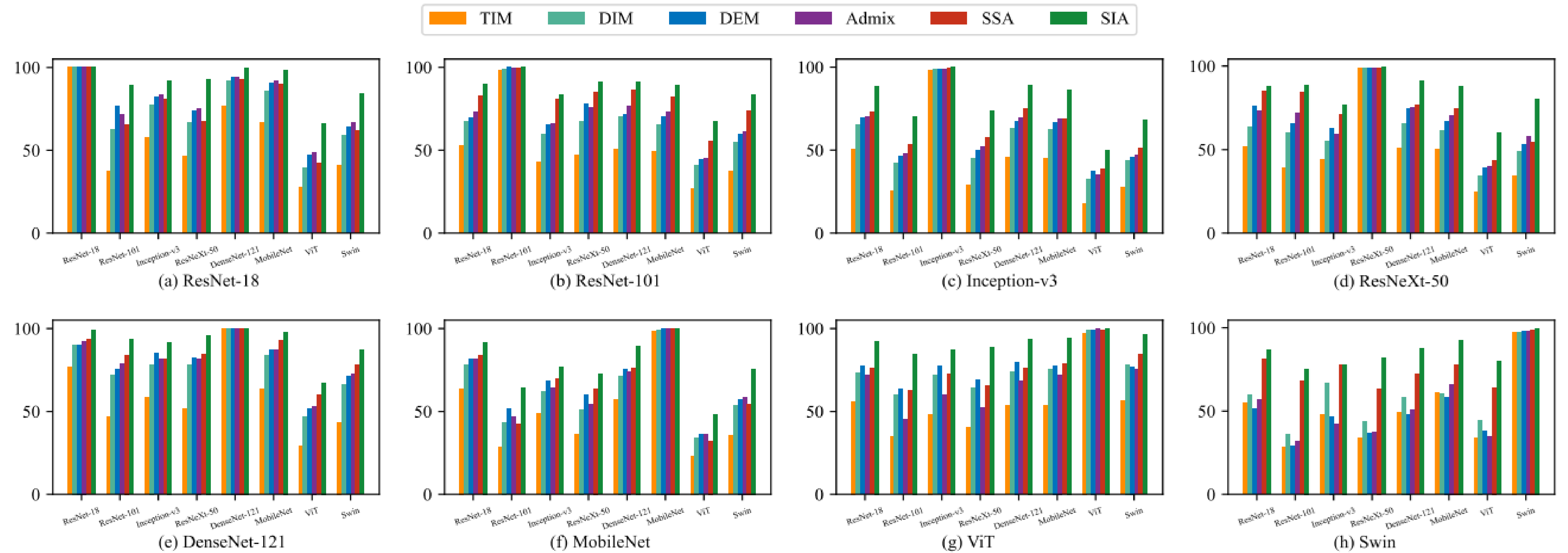


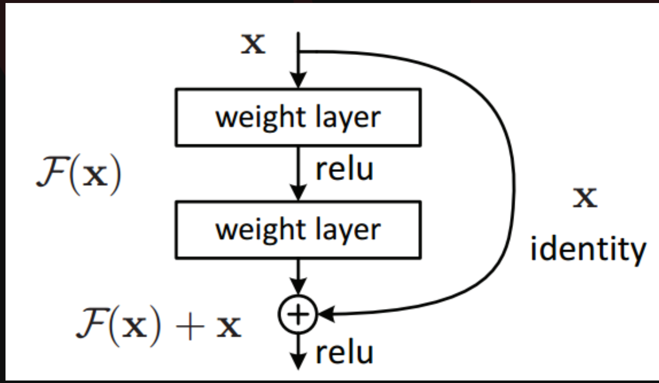
Figure 3: Attack success rates (%) of eight deep models on the adversarial examples crafted on each model by TIM, DIM, DEM, *Admix*, SSA, and SIA.

Wang et al. Structure Invariant Transformation for better Adversarial Transferability. ICCV 2023.

CONTENTS

Model-related Attacks

- **Modifying the surrogate model to boost adversarial transferability.**
  - **Ghost Network** [Li et al., 2020]: Densely add dropout layer and randomly scale the feature passing the skip connection of ResNets.
  - **SGM** [Wu et al., 2020]: Adopt more gradient from the skip connections instead of the residual modules using a decay factor for backpropagation.
  - **LinBP** [Guo et al., 2020]: Adopt constant value as the gradient of ReLU activation and modify the gradient of residual modules to makes backpropagation more linear.



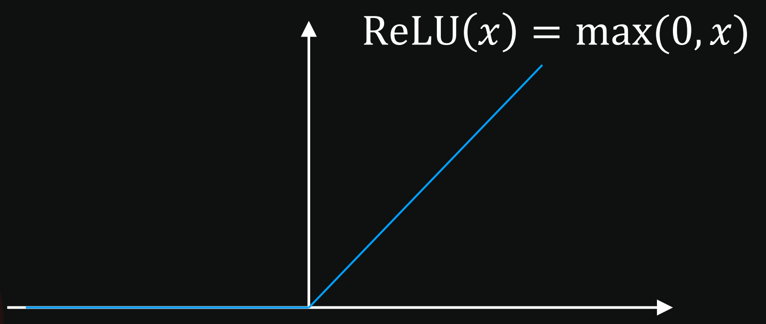
Li et al. Learning Transferable Adversarial Examples via Ghost Networks. AAAI 2020.
Wu et al. Skip Connections Matter: On the Transferability of Adversarial Examples Generated with ResNets. ICLR 2020.
Guo et al. Backpropagating Linearly Improves Transferability of Adversarial Examples. NeurIPS 2020.

Model-related Attacks

- **Backward Propagation Attack (BPA)**

Backpropagation follows the chain rule:

$$\frac{\partial J(x,y;\theta)}{\partial x} = \frac{\partial J(x,y;\theta)}{\partial f_{l+1}(z_l)} \left(\prod_{i=k+1}^{l} \frac{\partial f_{i+1}(z_i)}{\partial z_i} \right) \frac{\partial z_{k+1}}{\partial z_k} \frac{\partial z_k}{\partial_x}$$



Non-linear layers result in the truncation of gradients *w.r.t.* images.

- ReLU activation function

$$\frac{\partial z_{i+1}}{\partial z_i} = \begin{cases} 1 & \text{if } z_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

- Maxpooling layer

$$\frac{\partial z_{i+1}}{\partial z_i} = \begin{cases} 1 & \text{if } z_i \text{ is the maximum value in the window} \\ 0 & \text{otherwise} \end{cases}$$

0.1	-0.2	1.9	1.4
0.0	-0.5	2.3	0.7
-0.4	0.9	1.0	-2.0
0.7	0.6	0.5	1.7

Wang et al. Rethinking the Backward Propagation for Adversarial Transferability. Under review.

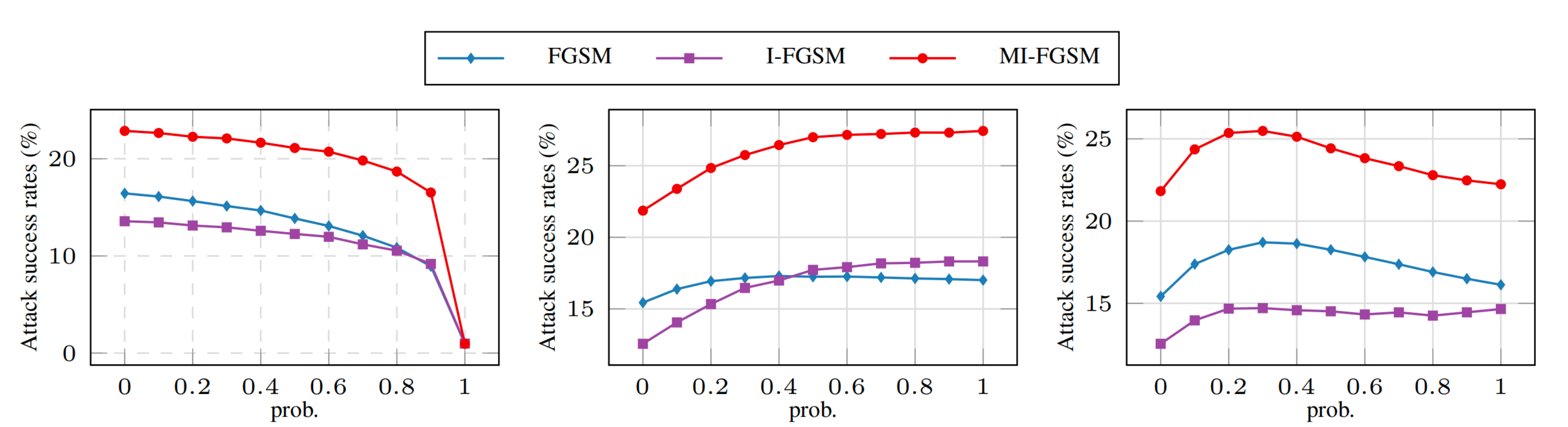
Model-related Attacks

- **Backward Propagation Attack (BPA)**

Assumption: The truncation of gradient introduced by non-linear layers in the backward propagation process decays the adversarial transferability.

➢ Randomly mask the gradient to introduce more truncation.

➢ Randomly replace the zeros in the gradient of ReLU or maxpooling layers with ones



(a) Randomly mask the gradient (b) Recover the gradient of ReLU (c) Recover the gradient of max-pooling

Gradient Truncation decays the transferability!

Wang et al. Rethinking the Backward Propagation for Adversarial Transferability. Under review.

Model-related Attacks

- **Backward Propagation Attack (BPA)**

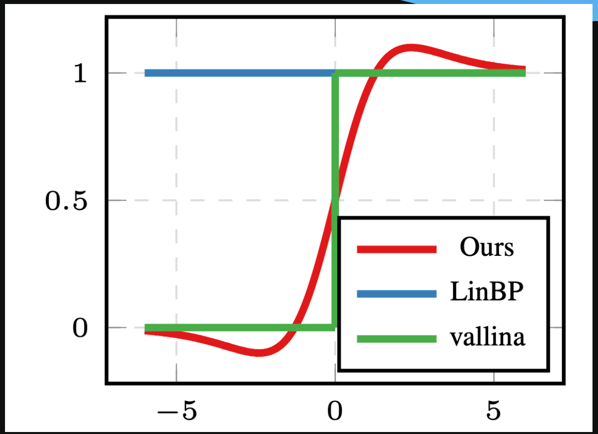
Recover the truncated gradient for better transferability:

➤ Replace the gradient of ReLU with that of SiLU

$$\frac{\partial z_{i+1}}{\partial z_i} = \sigma(z_i)\left(1 + z_i \cdot \left(1 - \sigma(z_i)\right)\right)$$

➤ Adopting the Softmax function to calculate the gradient within each window w of the max-pooling:

$$\left[\frac{\partial z_{k+1}}{\partial z_k}\right]_{i,j,w} = \frac{e^{t \cdot z_{k,i,j}}}{\sum_{v \in w} e^{t \cdot v}}$$



0.1	-0.2	1.9	1.4
0.0	-0.5	2.3	0.7
-0.4	0.9	1.0	-2.0
0.7	0.6	0.5	1.7

Wang et al. Rethinking the Backward Propagation for Adversarial Transferability. Under review.

Model-related Attacks

- **Backward Propagation Attack (BPA)**

Attacker	Method	Inc-v3	IncRes-v2	DenseNet	MobileNet	PNASNet	SENet	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
PGD	N/A	16.34	13.38	36.86	36.12	13.46	17.14	10.24	9.46	5.52
	SGM	23.68	19.82	51.66	55.44	22.12	30.34	13.78	12.38	7.90
	LinBP	27.22	23.04	59.34	59.74	22.68	33.72	16.24	13.58	7.88
	Ghost	17.74	13.68	42.36	41.06	13.92	19.10	11.60	10.34	6.04
	BPA	**35.36**	**30.12**	**70.70**	**68.90**	**32.52**	**42.02**	**22.72**	**19.28**	**12.40**
MI-FGSM	N/A	26.20	21.50	51.50	49.68	22.92	30.12	16.22	14.58	9.00
	SGM	33.78	28.84	63.06	65.84	31.90	41.54	19.56	17.48	10.98
	LinBP	35.92	29.82	68.66	69.72	30.24	41.68	19.98	16.58	9.94
	Ghost	29.76	23.68	57.28	56.10	25.00	34.76	17.10	14.76	9.50
	BPA	**47.58**	**41.22**	**80.54**	**79.40**	**44.70**	**54.28**	**32.06**	**25.98**	**17.46**
VMI-FGSM	N/A	42.68	36.86	68.82	66.68	40.78	46.34	27.36	24.20	17.18
	SGM	50.04	44.28	77.56	79.34	48.58	56.86	32.22	27.72	19.66
	LinBP	47.70	40.40	77.44	78.76	41.48	52.10	28.58	24.06	16.60
	Ghost	47.82	41.42	75.98	73.40	44.84	52.78	30.84	27.18	19.08
	BPA	**55.00**	**48.72**	**85.44**	**83.64**	**52.02**	**60.88**	**38.76**	**33.70**	**23.78**
ILA	N/A	29.10	26.08	58.02	59.10	27.60	39.16	15.12	12.30	7.86
	SGM	35.64	32.34	65.20	71.22	34.20	46.72	17.10	13.86	9.08
	LinBP	37.36	34.24	71.98	72.84	35.12	48.80	19.38	14.10	9.28
	Ghost	30.06	26.50	60.52	61.74	28.68	40.46	14.84	12.54	7.90
	BPA	**47.62**	**43.50**	**81.74**	**80.88**	**47.88**	**60.64**	**27.94**	**20.64**	**14.76**
SSA	N/A	35.78	29.58	60.46	64.70	25.66	34.18	20.64	17.30	11.44
	SGM	45.22	38.98	70.22	78.44	35.30	46.06	26.28	21.64	14.50
	LinBP	48.48	41.90	75.02	78.30	36.66	49.58	28.76	23.64	15.46
	Ghost	36.44	28.62	61.12	66.80	24.90	33.98	20.58	16.84	10.82
	BPA	**51.36**	**44.70**	**76.24**	**79.66**	**39.38**	**50.00**	**32.10**	**26.44**	**18.20**

Wang et al. Rethinking the Backward Propagation for Adversarial Transferability. Under review.

KCon

归源 · 智变
Returning to the source · Intelligent transformation

CONTENTS

Advanced Objective Functions

- **Several attacks disrupt the high-level features:**
  - ➢ **FIA** [Wang et al., 2021]: Adopt aggregate gradient to highlight important features:

$$\bar{\boldsymbol{\Delta}}_k^x = \frac{1}{C}\sum_{n=1}^{N}\frac{\partial J\left(x \odot M_p^n, y; \theta\right)}{\partial f_k\left(x \odot M_p^n\right)}, M_p \sim \text{Bernoulli}(1-p), L(x) = \sum\left(\bar{\Delta}_k^x \odot f_k(x)\right)$$

  - ➢ **RPA** [Zhang et al., 2022]: Instead of randomly masking the pixels, RPA randomly split the image into patches, which will be randomly masked for calculating the weight matrix.
  - ➢ **NAA** [Zhang et al., 2022]: Adopt integrated gradients for neuron attribution:

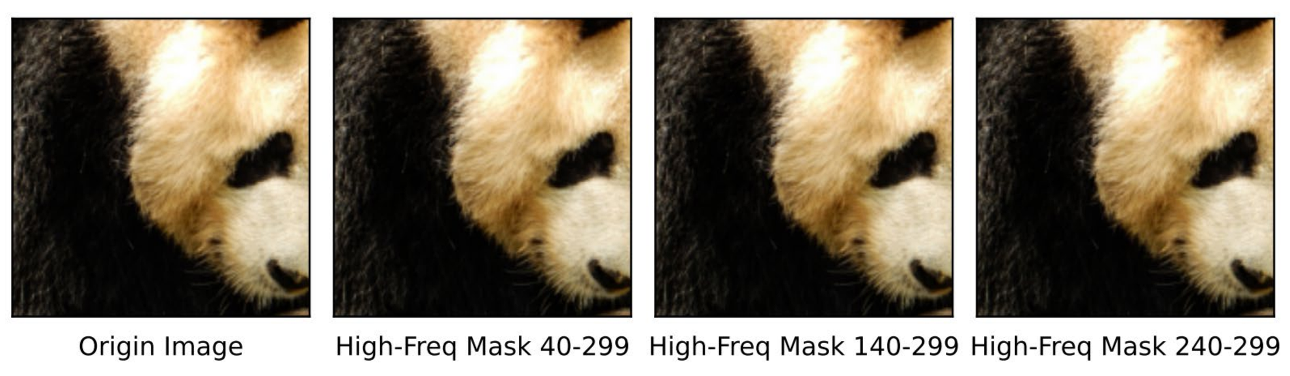
$$\bar{\Delta}_k^x = \frac{1}{N}\sum_{n=1}^{N}\frac{\partial J\left(x' + \frac{n}{N}(x - x'), y; \theta\right)}{\partial f_k\left(x' + \frac{n}{N}(x - x')\right)}, L(x) = \sum\left|\bar{\Delta}_k^x \odot \left(f_k(x) - f_k(x')\right)\right|$$

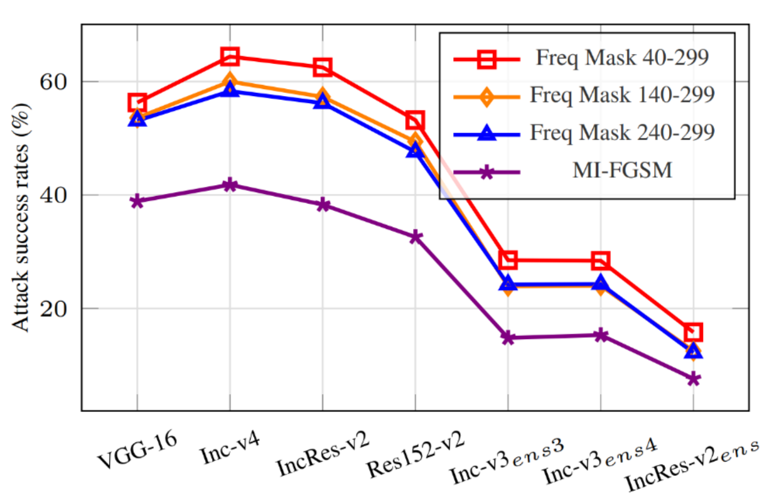
Wang et al. Feature Importance-aware Transferable Adversarial Attacks. ICCV 2021.
Zhang et al. Enhancing the Transferability of Adversarial Examples with Random Patch. IJCAI 2022.
Zhang et al. Improving Adversarial Transferability via Neuron Attribution-based Attacks. CVPR 2022.

Advanced Objective Functions

- **Semantic and Abstract FEatures disRuption (SAFER)**

DNNs usually focus more on high-frequency components (*e.g.*, texture, edge)



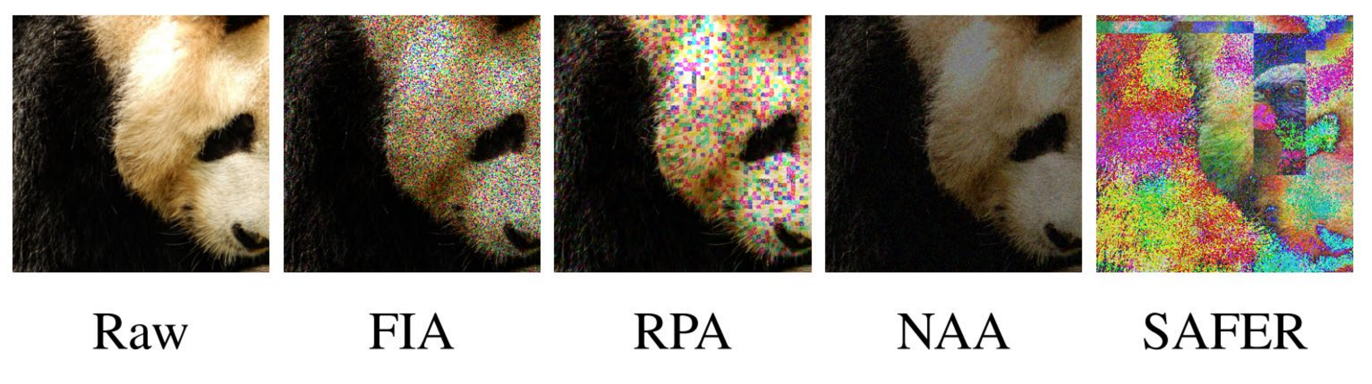


High frequency components are beneficial for boosting adversarial transferability!

Wang et al. Disrupting Semantic and Abstract Features for better Adversarial Transferability. Under review.

Advanced Objective Functions

- **Semantic and Abstract FEatures disRuption (SAFER)**



Randomly perturbing the semantic and abstract features:

Blockmix: $B(x,x') = \begin{cases} x_{i,j} & \text{with the probability } p \\ x'_{i,j} & \text{with the probability } 1-p \end{cases}$

Frequency Perturbation: $FP(x) = \mathcal{D}_I(\mathcal{D}(x+\xi) \odot \mathcal{M})$

$$x_{SAFER} = FP\big(B(x,x')\big), \bar{\boldsymbol{\Delta}}_k^x = \frac{1}{C}\sum_{n=1}^{N} \frac{\partial J(\boldsymbol{x_{SAFER}}, y; \theta)}{\partial f_k(\boldsymbol{x_{SAFER}})}, L(x) = \sum\big(\bar{\Delta}_k^x \odot f_k(x)\big)$$

Wang et al. Disrupting Semantic and Abstract Features for better Adversarial Transferability. Under review.

Advanced Objective Functions

- **Semantic and Abstract FEatures disRuption (SAFER)**

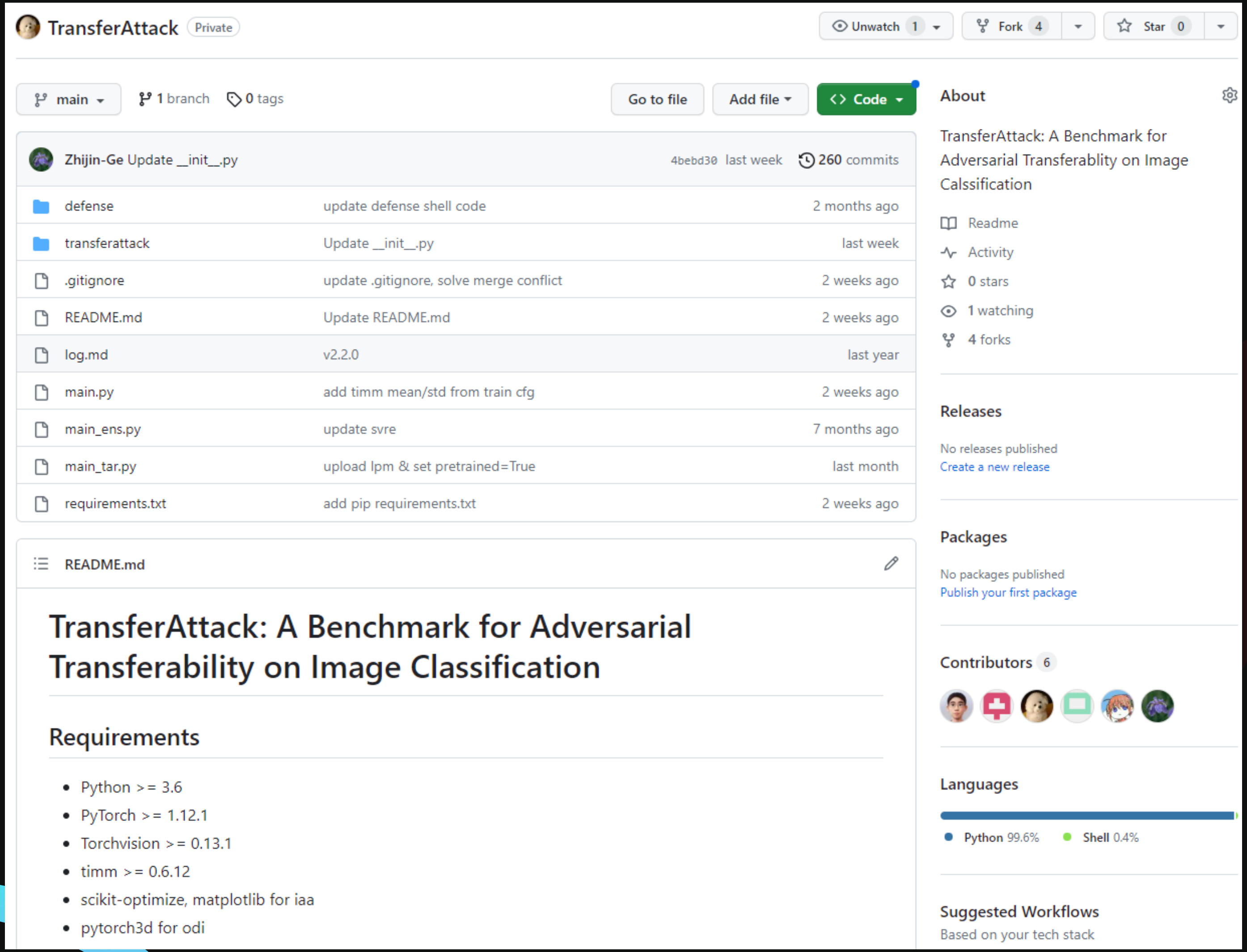
Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-152	VGG-16	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
Inc-v3	MIM	**100.0***	42.4	39.8	33.0	39.6	15.4	15.9	7.7
	FIA	98.3*	83.3	80.1	72.4	71.4	43.3	43.6	23.5
	RPA	97.9*	84.1	82.4	77.7	75.7	44.8	45.0	25.7
	NAA	97.0*	82.9	81.3	74.7	70.1	49.9	50.2	30.2
	SAFER	98.7*	**87.7**	**86.7**	**80.4**	**80.0**	**52.1**	**52.6**	**32.2**
Inc-v4	MIM	59.7	**100.0***	45.3	38.8	47.7	18.5	18.3	9.2
	FIA	75.0	90.2*	70.4	65.2	65.5	39.4	39.2	23.8
	RPA	79.1	92.8*	75.2	69.0	70.2	44.2	43.5	25.7
	NAA	81.8	96.1*	76.1	71.4	70.2	47.2	45.7	31.2
	SAFER	**86.9**	97.6*	**83.5**	**79.4**	**80.0**	**51.9**	**50.5**	**32.0**
Res-152	MIM	52.6	47.8	44.9	**99.5***	50.3	24.5	24.3	12.0
	FIA	80.6	78.6	77.6	98.2*	75.9	52.9	48.6	34.0
	RPA	81.4	80.1	80.2	98.0*	76.4	56.4	50.8	37.6
	NAA	83.9	82.2	80.4	97.5*	78.7	59.5	56.3	43.5
	SAFER	**87.6**	**86.2**	**86.2**	99.1*	**83.9**	**61.9**	**58.2**	**44.7**
VGG-16	MIM	83.0	81.6	76.4	79.5	100.0*	76.6	73.2	62.2
	FIA	95.7	96.7	94.3	94.2	100.0*	91.8	92.3	86.6
	RPA	96.2	96.3	93.4	94.1	100.0*	92.5	93.2	88.3
	NAA	94.5	93.4	91.1	92.3	98.3*	91.1	90.3	82.6
	SAFER	**98.0**	**97.3**	**95.8**	**95.6**	**100.0***	**93.9**	**93.7**	**90.4**

Wang et al. Disrupting Semantic and Abstract Features for better Adversarial Transferability. Under review.

CONTENTS

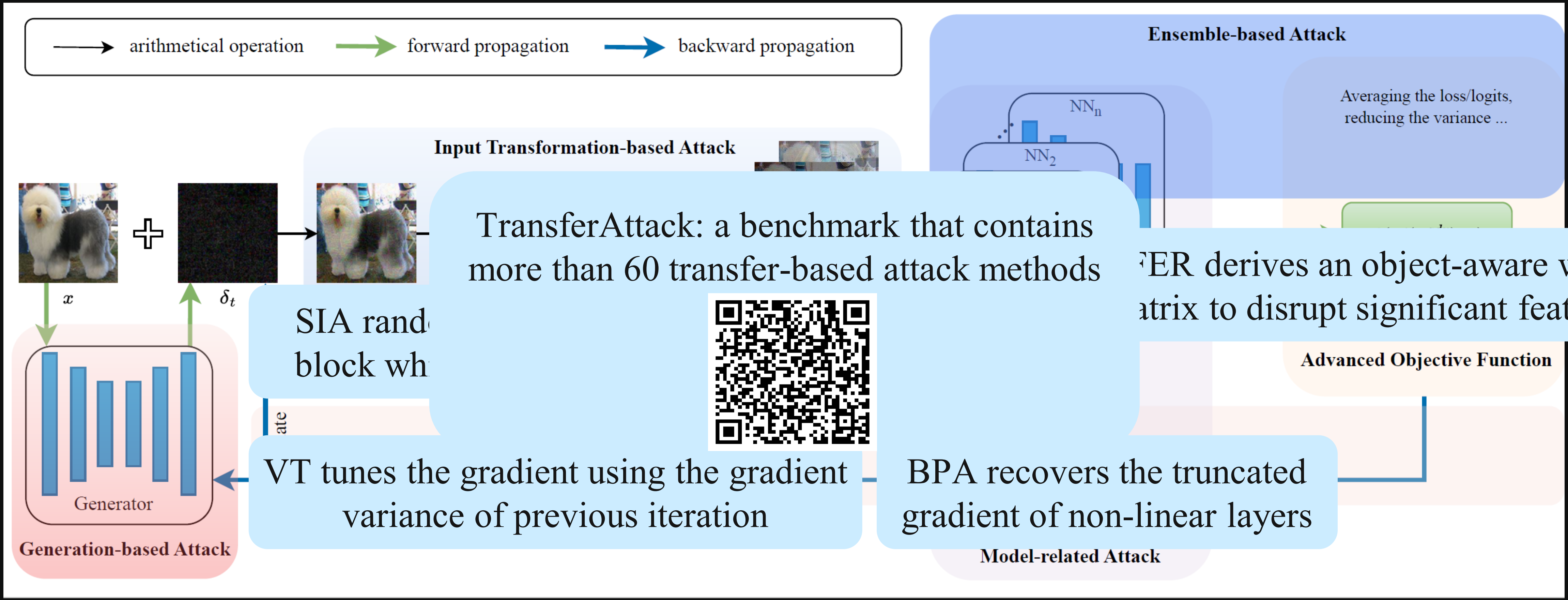
Further Discussion & Conclusion

- TransferAttack: a benchmark containing more than 60 transfer-based attack methods



The framework will be released soon!

Further Discussion & Conclusion



TransferAttack: a benchmark that contains more than 60 transfer-based attack methods

SIA rand
block wh

ER derives an object-aware weight
atrix to disrupt significant features

VT tunes the gradient using the gradient variance of previous iteration

BPA recovers the truncated gradient of non-linear layers

Thanks for your Attention!
Q&A