

Enabling Heterogeneous Adversarial Transferability via Feature Permutation Attacks

Tao Wu¹, Thomas Tie Luo²

¹Missouri University of Science and Technology (now with ByteDance, CA, USA)

²Department of Electrical and Computer Engineering & Department of Computer Science, University of Kentucky



Presenter: Thomas Tie Luo

Background

Adversarial examples (AEs):

• Inputs to a deep learning model that have been intentionally modified in small, often imperceptible ways to cause the model to make wrong predictions.

"Making a pig fly" isn't that hard:



Background

4

Transfer-based black-box attacks

- The most **realistic** attacks requires little knowledge about target models
- The key is to generate "transferable" (generalizable) AEs



The gap

- Many such transferrable attacks have been proposed and shown to be successful (among CNNs)
- However, transferring across heterogeneous architectures (e.g., CNNs, ViTs, MLPs) has been rather ineffective

Our empirical finding:

Attack Success Rate Across Models and Methods

80 Target Models 75.43% DenseNet MLP-Mixer 70 ViT I-FGSM Trend Attack Success Rate (%) 58.62% **MI-FGSM** Trend **FPA-N** Trend 49.35% 30.16% 25.38% 11.58% 8.76% 10 6.82% 4.38% MI-FGSM I-FGSM FPA-N **Attack Methods**

June 12, 2025

Hypothesis

- Inspired by the observation of receptive fields of CNNs as compared to ViTs, we hypothesize that:
- The poor adversarial transferability is due to CNNs' inadequacy in attending to long-range dependencies and large contexts.





Inductive bias

Raghu, Maithra, et al. "Do vision transformers see like convolutional neural networks?" NeurIPS (2021).



Method

- Introduce long-range dependencies into CNNs
 - by proposing a Feature Permutation Attack (FPA)
- Permute feature maps inside the surrogate model during the process of generating AEs:
 - FPA-R: random
 - FPA-N: neighborhood



	1	2	2	2
	0	6	7	8
_	4	4	1	5
	1	1	3	3

Rearrange pixels within a feature map randomly

Random

Permutation

<u> </u>		2	4	
5	6	7	8	
3	2	1	0	٢
1	2	3	4	
_				•







Exchange each pixel with one of its four neighboring pixels (randomly chosen)

Difference?

- **FPA-R:** directly introduces global (long-range) dependency
- **FPA-N:** much more indirect, preserves local spatial relationship more

- Since there are many feature maps in a CNN, which particular feature maps to permute? By how much?
 - *l*:Location (layer/block)
 - γ : ratio of channels
 - *p* : permutation probability per iteration



7

Experiments

- Target models under attack: 7 CNNs, 4 ViTs, 3 MLPs
 - CNNs: VGG-19 [22], ResNet-152 [10], Inception v3 [23], DenseNet121 [11], MobileNet v2 [21], WRN [37], PNASNet [15].
 - ViTs: ViT-B [7], DeiT-B [27], Swin-B [17], BEiT-B [1].
 - MLPs: Mixer-B [25], Res-MLP [26], gMLP [16].
- Surrogate model: ResNet-50
- 5,000 correctly classified test images from the ImageNet validation set (to generate AEs)
- **FPA-R:** *l* = 5, *γ* = 0.3, *p*= 0.2 (equiv: 6% of channels permuted)
- **FPA-N:** *l* = 2, *γ* = 0.6, *p*= 0.5 (equiv: 30% of channels permuted)

Results

• ASR: attack success rate

- FPA-N achieves the highest ASR in all 14 cases
 - +14.57 points on Swin-B (compared to the best non-FPA method)
 - +14.48 points on Res-MLP (compared to the best non-FPA method)

• FPA-R: the overall runner-up

[1-4	<u>``</u>				
	• ASI	R: atta	ack sud	ccess ra	ate						SM Serv	<i>′</i> 0			
	• FPA	A-Nac	chieve	s the hi	ghest A	SR i	n all '	14 ca	ses		·	and as the			
		+14.57	7 points	on Swin	n-B (com	pare	d to the	e best	non-Fl	PA met	hod)	***@£)ase		
		+14.48	3 points	on Res-	MLP (coi	mpar	red to t	he be	st non-	FPA m	ethod)		attac	k.	
	• FP/	A-R: th	ne ovei	rall runr	ner-up									FOR FP	1
Method	VGG-19]	ResNet-152	Inception-V3	DenseNet121	MobileNet-V2	WRN	PNASNet	ViT-B	DeiT-B	Swin-B	BEiT-B	Mixer-B	Res-MLP	gMLP	Average
I-FGSM	43.26%	23.65%	21.54%	49.35%	38.21%	45.32%	18.91%	4.38%	4.03%	4.96%	3.78%	8.76%	7.94%	7.12%	18.99%
MI-FGSM	52.89%	31.56%	32.16%	58.62%	50.35%	54.69%	29.32%	6.82%	5.86%	7.88%	6.76%	11.58%	10.92%	11.26%	27.83%
DIM	67.85%	41.25%	38.95%	70.26%	65.26%	68.42%	35.46%	10.49%	10.35%	11.06%	12.10%	15.68%	15.34%	14.82%	36.94%
TIM	46.78%	29.14%	27.83%	51.35%	48.31%	49.63%	25.34%	5.23%	5.65%	6.04%	4.97%	9.68%	10.03%	8.95%	26.08%
SIM	52.82%	35.68%	33.68%	58.96%	54.16%	58.47%	29.65%	9.35%	10.23%	10.56%	11.05%	11.65%	12.14%	10.98%	31.79%
Admix	66.95%	43.62%	39.46%	68.47%	59.21%	65.61%	30.49%	8.79%	9.62%	10.26%	11.67%	13.60%	13.43%	13.09%	34.63%
SGM	63.46%	46.52%	39.26%	71.26%	57.26%	64.18%	31.25%	11.24%	10.42%	10.96%	11.53%	14.82%	15.48%	15.67%	36.66%
LinBP	66.31%	50.18%	37.89%	69.43%	63.48%	68.14%	32.06%	12.06%	10.36%	11.23%	10.85%	14.62%	14.85%	15.21%	37.53%
FPA-R (ours)) 56.83%	43.04%	35.62%	66.59%	58.72%	60.84%	28.89%	16.39%	14.85%	15.68%	17.32%	18.46%	19.15%	19.52%	37.70%
FPA-N (ours)) 70.25%	$\mathbf{52.38\%}$	42.85%	75.43%	69.48%	72.34%	39.74%	25.38%	$\mathbf{24.64\%}$	$\mathbf{25.80\%}$	$\boldsymbol{26.19\%}$	$\mathbf{30.16\%}$	$\mathbf{31.43\%}$	$\mathbf{30.82\%}$	45.59%

FPA is very flexible

- Can be seamlessly integrated with probably any attack
 - Any attack could serve as the base and gain significant attack strength

Method	VGG-19]	ResNet-152	Inception-V3	DenseNet121	MobileNet-V2	WRN	PNASNet	ViT-B	DeiT-B	Swin-B	BEiT-B	Mixer-B	Res-MLP	gMLP	Average
MI-FGSM	52.89%	31.56%	32.16%	58.62%	50.35%	54.69%	29.32%	6.82%	5.86%	7.88%	6.76%	11.58%	10.92%	11.26%	27.83%
MI-FGSM + FPA-R	66.32%	49.13%	45.12%	71.56%	65.14%	69.10%	42.95%	18.26%	18.03%	17.95%	17.52%	21.06%	22.16%	22.53%	39.06%
MI-FGSM + FPA-N	75.46%	$\mathbf{57.64\%}$	$\mathbf{38.95\%}$	80.05%	$\mathbf{73.94\%}$	78.86%	$\boldsymbol{49.14\%}$	$\mathbf{27.95\%}$	$\mathbf{28.49\%}$	$\mathbf{28.65\%}$	$\mathbf{29.33\%}$	$\mathbf{34.02\%}$	$\mathbf{34.57\%}$	33.13%	$\mathbf{47.87\%}$
DIM	67.85%	41.25%	38.95%	70.26%	65.26%	68.42%	35.46%	10.49%	10.35%	11.06%	12.10%	15.68%	15.34%	14.82%	36.94%
$\mathrm{DIM}+\mathrm{FPA-R}$	75.61%	49.12%	46.35%	76.12%	74.31%	77.03%	45.61%	21.30%	19.16%	18.94%	23.15%	24.96%	23.84%	25.61%	42.94%
DIM + FPA-N	80.05%	$\mathbf{54.10\%}$	$\mathbf{50.23\%}$	79.96%	77.56%	82.04%	49.34%	$\mathbf{29.65\%}$	$\mathbf{31.49\%}$	$\mathbf{33.16\%}$	$\mathbf{32.09\%}$	$\mathbf{36.16\%}$	$\mathbf{36.98\%}$	$\mathbf{35.88\%}$	$\mathbf{50.62\%}$
Admix	66.95%	43.62%	39.46%	68.47%	59.21%	65.61%	30.49%	8.79%	9.62%	10.26%	11.67%	13.60%	13.43%	13.09%	34.63%
Admix + FPA-R	74.35%	48.13%	45.19%	75.49%	68.95%	76.01%	38.49%	17.53%	19.23%	20.15%	22.36%	25.16%	25.01%	24.69%	41.48%
Admix + FPA-N	79.64%	$\mathbf{50.09\%}$	$\mathbf{51.29\%}$	80.13%	$\mathbf{76.95\%}$	81.32%	$\mathbf{44.68\%}$	27.32%	$\mathbf{28.96\%}$	$\mathbf{30.40\%}$	$\mathbf{33.46\%}$	$\mathbf{32.68\%}$	$\mathbf{32.92\%}$	$\mathbf{34.05\%}$	$\mathbf{53.24\%}$

- Performance increases ~20, 14, and 19 points (see last column) by FPA-N
 - Even FPA-R achieves quite notable gains too

Ablation study

- FPA-N (triangular marker) is not sensitive to hyperparameter variation
- Dash-lines (horizontal) are vanilla attacks without FPA
- FPA-N consistently outperforms FPA-R, as FPA-N better preserves local contextual information.

• Our proposed permutation operation is executed solely through memory operations without requiring matrix computations, additional parameters, or FLOPs.

Methods	I-FGSM	MI-FGSM	DIM	TIM	SIM	Admix	SGM	I FPA-R	, FPA-N
Time (mins)	4.2	4.9	5.9	6.7	21.6	15.3	4.5	4.2	4.3
e 3: Comparing	g wall c	lock runtir	ne fo	r FP	A an	d base	eline	attacks	on Ima

Conclusion

- We **hypothesize** that the failure of heterogeneous adversarial transfer is due to CNN's inadequacy of modeling **long-range dependencies**
- We propose Feature Permutation Attack to address this limitation
- Flexible plug-in: probably any attack can serve as the base
- FPA improves attack success rates significantly (by 8-26 percentage points) even in the heterogeneous setting (from CNN to ViT and MLP)
- FPA is simple and efficient: it introduces zero FLOP and zero model parameters.

