Linear Modeling of the Adversarial Noise Space

Jordan Patracone[1], Lucas Anquetil[2], Yuan Liu[2], Gilles Gasso[2], Stéphane Canu[2]

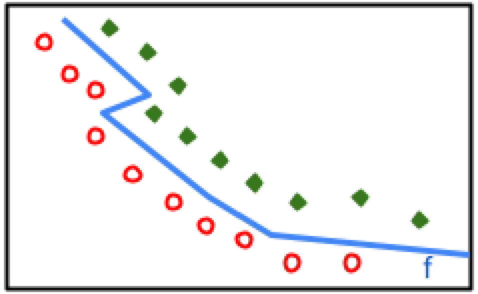
[1] Inria MALICE, Lab. Hubert Curien, France

[2] LITIS, France

Adversarial Attacks

Among the various adversarial attacks, we restrict to perburbation-based attacks

Problem: Given a classifier C_f

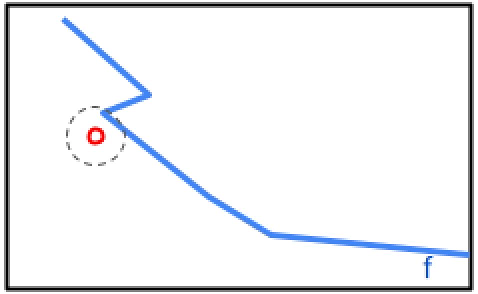


f typically is a neural network with associated classifier C_f

Adversarial Attacks

Among the various adversarial attacks, we restrict to perburbation-based attacks

Problem: Given a classifier C_f, find a small perturbation (*adversarial noise*) to a well classified example



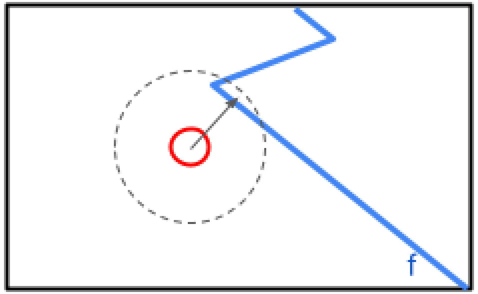
f typically is a neural network with associated classifier C_f

small $\Leftrightarrow$ inside a ℓ_p-ball with given small radius: ℓ_p-attack

Adversarial Attacks

Among the various adversarial attacks, we restrict to perburbation-based attacks

Problem: Given a classifier C_f, find a small perturbation (*adversarial noise*) to a well classified example such that the perturbed example (*adversarial example*) becomes misclassified.



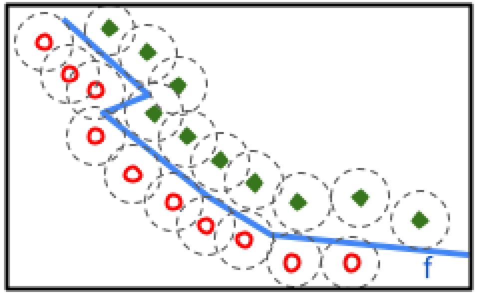
f typically is a neural network with associated classifier C_f

small $\Leftrightarrow$ inside a ℓ_p-ball with given small radius: ℓ_p-attack

Two Paradigms: Specific vs. Universal

Specific Attacks

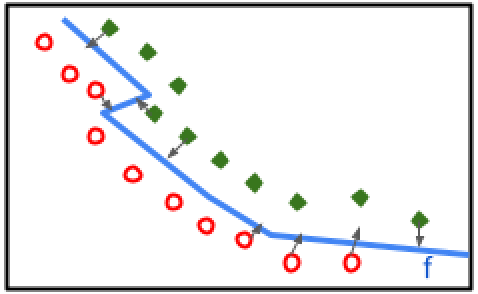
For each $\mathbf{x}^{(i)}$, learn $\epsilon^{(i)}$ such that
$\mathbf{x}^{(i)'} = \mathbf{x}^{(i)} + \epsilon^{(i)}$ is an adversarial example



Two Paradigms: Specific vs. Universal

Specific Attacks

For each $\mathbf{x}^{(i)}$, learn $\epsilon^{(i)}$ such that
$\mathbf{x}^{(i)'} = \mathbf{x}^{(i)} + \epsilon^{(i)}$ is an adversarial example

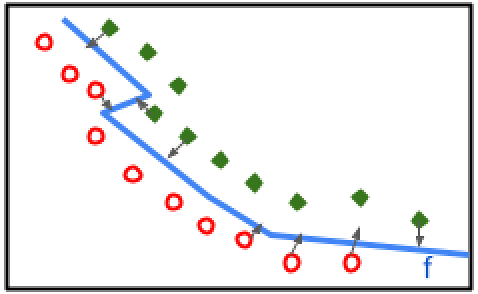


High fooling rate

Two Paradigms: Specific vs. Universal

Specific Attacks

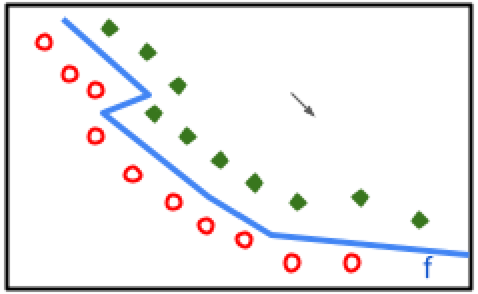
For each $\mathbf{x}^{(i)}$, learn $\epsilon^{(i)}$ such that
$\mathbf{x}^{(i)'} = \mathbf{x}^{(i)} + \epsilon^{(i)}$ is an adversarial example



High fooling rate

Universal Attack

Learn ϵ such that, for each $\mathbf{x}^{(i)}$,
$\mathbf{x}^{(i)'} = \mathbf{x}^{(i)} + \epsilon$ is an adversarial example

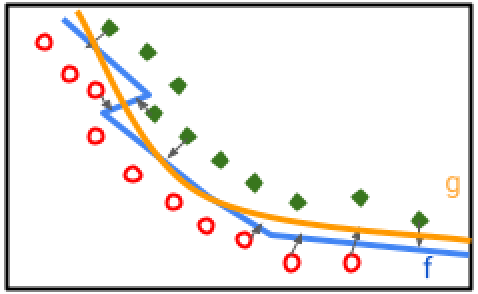


Poor fooling rate

Two Paradigms: Specific vs. Universal

Specific Attacks

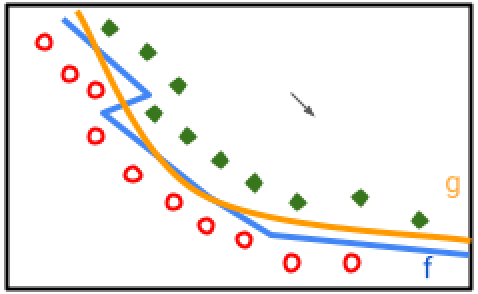
For each $\mathbf{x}^{(i)}$, learn $\epsilon^{(i)}$ such that
$\mathbf{x}^{(i)'} = \mathbf{x}^{(i)} + \epsilon^{(i)}$ is an adversarial example



High fooling rate
Poor transferability

Universal Attack

Learn ϵ such that, for each $\mathbf{x}^{(i)}$,
$\mathbf{x}^{(i)'} = \mathbf{x}^{(i)} + \epsilon$ is an adversarial example



Poor fooling rate
High transferability

Proposed Attack

Principle

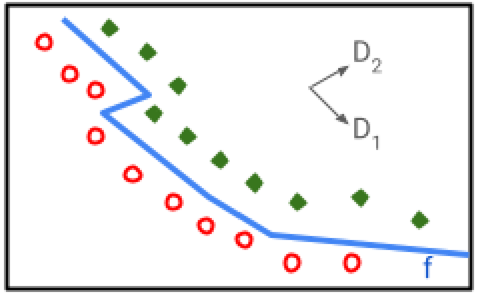
LIMANS

Linear Modeling of the Adversarial Noise Space

$$\mathbf{x}^{(i)\prime} = \mathbf{x}^{(i)} + D\mathbf{v}^{(i)}$$

$D = [D_1, \ldots, D_M]$ are universal directions (*size of* $\mathbf{x}^{(i)}$)

$\mathbf{v}^{(i)} = [\mathbf{v}_1^{(i)}, \ldots, \mathbf{v}_M^{(i)}]$ are specific coding vectors (*scalars*)



Principle

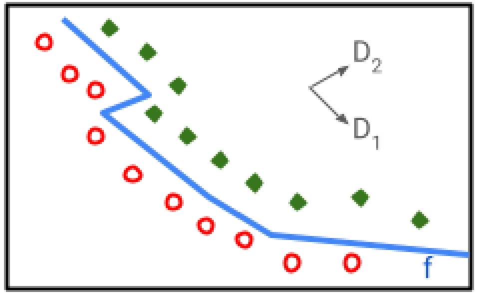
LIMANS

Linear Modeling of the Adversarial Noise Space

$$\mathbf{x}^{(i)'} = \mathbf{x}^{(i)} + D\mathbf{v}^{(i)}$$

$D = [D_1, \ldots, D_M]$ are universal directions (*size of* $\mathbf{x}^{(i)}$)

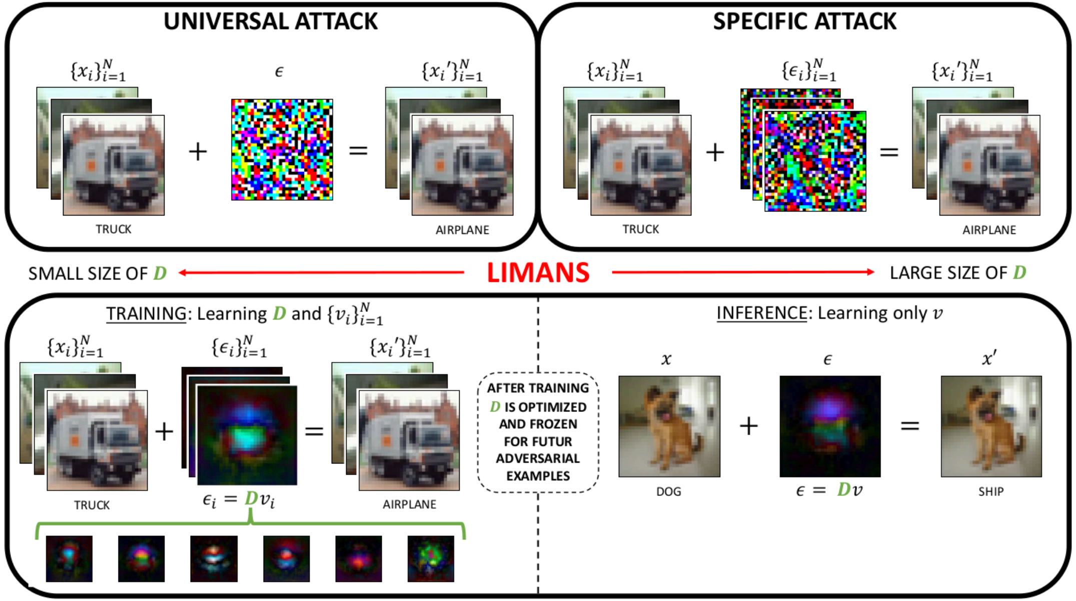
$\mathbf{v}^{(i)} = [\mathbf{v}_1^{(i)}, \ldots, \mathbf{v}_M^{(i)}]$ are specific coding vectors (*scalars*)



High fooling rate

High transferability

Principle



By tuning the size of D, LIMANS bridges the gap between universal and specific attacks

Optimization Problem

$$\underset{\substack{D=[D_1,\ldots,D_M]\in\mathbb{R}^{P\times M}\\ V=[\mathbf{v}^{(1)},\ldots\mathbf{v}^{(N)}]\in\mathbb{R}^{M\times N}}}{\text{maximize}} \frac{1}{N}\sum_{i=1}^{N} 1_{\{C_f(\mathbf{x}^{(i)\prime})\neq C_f(\mathbf{x}^{(i)})\}}$$

$$\text{s.t.} \quad \begin{cases} \mathbf{x}^{(i)\prime} = \mathbf{x}^{(i)} + D\mathbf{v}^{(i)} \in \mathcal{X} & ,(\forall i \in \{1,\ldots,N\}) \quad \textit{Valid examples} \\ \|D\mathbf{v}^{(i)}\|_p \leq \delta_p & ,(\forall i \in \{1,\ldots,N\}) \quad \textit{Small perturbations} \\ \|D_j\|_p = 1 & ,(\forall j \in \{1,\ldots,M\}) \quad \textit{Normalized directions} \end{cases}$$

Optimization Problem

$$\underset{\substack{D=[D_1,\ldots,D_M]\in\mathbb{R}^{P\times M}\\ V=[\mathbf{v}^{(1)},\ldots\mathbf{v}^{(N)}]\in\mathbb{R}^{M\times N}}}{\text{maximize}} \frac{1}{N}\sum_{i=1}^{N} 1_{\{C_f(\mathbf{x}^{(i)\prime})\neq C_f(\mathbf{x}^{(i)})\}}$$

$$\text{s.t.} \quad \begin{cases} \mathbf{x}^{(i)\prime} = \mathbf{x}^{(i)} + D\mathbf{v}^{(i)} \in \mathcal{X} & ,(\forall i \in \{1,\ldots,N\}) \quad \textit{Valid examples} \\ \|D\mathbf{v}^{(i)}\|_p \leq \delta_p & ,(\forall i \in \{1,\ldots,N\}) \quad \textit{Small perturbations} \\ \|D_j\|_p = 1 & ,(\forall j \in \{1,\ldots,M\}) \quad \textit{Normalized directions} \end{cases}$$

Solving this problem is a challenge for three main reasons:

-
-
-

Optimization Problem

$$\underset{\substack{D=[D_1,\ldots,D_M]\in\mathbb{R}^{P\times M} \\ V=[\mathbf{v}^{(1)},\ldots\mathbf{v}^{(N)}]\in\mathbb{R}^{M\times N}}}{\text{maximize}} \frac{1}{N}\sum_{i=1}^{N} 1_{\{C_f(\mathbf{x}^{(i)\prime})\neq C_f(\mathbf{x}^{(i)})\}}$$

$$\text{s.t.} \quad \begin{cases} \mathbf{x}^{(i)\prime} = \mathbf{x}^{(i)} + D\mathbf{v}^{(i)} \in \mathcal{X} & ,(\forall i \in \{1,\ldots,N\}) \quad \textit{Valid examples} \\ \|D\mathbf{v}^{(i)}\|_p \leq \delta_p & ,(\forall i \in \{1,\ldots,N\}) \quad \textit{Small perturbations} \\ \|D_j\|_p = 1 & ,(\forall j \in \{1,\ldots,M\}) \quad \textit{Normalized directions} \end{cases}$$

Solving this problem is a challenge for three main reasons:

- The indicator function $1_{\mathcal{S}}$ which is non-convex
-
-

Optimization Problem

$$\underset{\substack{D=[D_1,\ldots,D_M]\in\mathbb{R}^{P\times M}\\ V=[\mathbf{v}^{(1)},\ldots\mathbf{v}^{(N)}]\in\mathbb{R}^{M\times N}}}{\text{maximize}} \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(C_f(\mathbf{x}^{(i)'}), C_f(\mathbf{x}^{(i)}))$$

$$\text{s.t.} \quad \begin{cases} \mathbf{x}^{(i)'} = \mathbf{x}^{(i)} + D\mathbf{v}^{(i)} \in \mathcal{X} & ,(\forall i \in \{1,\ldots,N\}) & \textit{Valid examples} \\ \|D\mathbf{v}^{(i)}\|_p \leq \delta_p & ,(\forall i \in \{1,\ldots,N\}) & \textit{Small perturbations} \\ \|D_j\|_p = 1 & ,(\forall j \in \{1,\ldots,M\}) & \textit{Normalized directions} \end{cases}$$

Solving this problem is a challenge for three main reasons:

- The indicator function $1_{\mathcal{S}}$ which is non-convex → replace by surrogate loss function
-
-

Optimization Problem

$$\underset{\substack{D=[D_1,\ldots,D_M]\in\mathbb{R}^{P\times M}\\ V=[\mathbf{v}^{(1)},\ldots\mathbf{v}^{(N)}]\in\mathbb{R}^{M\times N}}}{\text{maximize}} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(C_f(\mathbf{x}^{(i)'}), C_f(\mathbf{x}^{(i)}))$$

$$\text{s.t.} \quad \begin{cases} \mathbf{x}^{(i)'} = \mathbf{x}^{(i)} + D\mathbf{v}^{(i)} \in \mathcal{X} & ,(\forall i \in \{1,\ldots,N\}) \quad \textit{Valid examples} \\ \|D\mathbf{v}^{(i)}\|_p \leq \delta_p & ,(\forall i \in \{1,\ldots,N\}) \quad \textit{Small perturbations} \\ \|D_j\|_p = 1 & ,(\forall j \in \{1,\ldots,M\}) \quad \textit{Normalized directions} \end{cases}$$

Solving this problem is a challenge for three main reasons:

- The indicator function $1_{\mathcal{S}}$ which is non-convex → replace by surrogate loss function
- The presence of the DNN f that is non-linear
-

Optimization Problem

$$\underset{\substack{D=[D_1,\ldots,D_M]\in\mathbb{R}^{P\times M}\\ V=[\mathbf{v}^{(1)},\ldots\mathbf{v}^{(N)}]\in\mathbb{R}^{M\times N}}}{\text{approx maximize}} \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(C_f(\mathbf{x}^{(i)'}), C_f(\mathbf{x}^{(i)}))$$

$$\text{s.t.} \quad \begin{cases} \mathbf{x}^{(i)'} = \mathbf{x}^{(i)} + D\mathbf{v}^{(i)} \in \mathcal{X} & ,(\forall i \in \{1,\ldots,N\}) \quad \textit{Valid examples} \\ \|D\mathbf{v}^{(i)}\|_p \leq \delta_p & ,(\forall i \in \{1,\ldots,N\}) \quad \textit{Small perturbations} \\ \|D_j\|_p = 1 & ,(\forall j \in \{1,\ldots,M\}) \quad \textit{Normalized directions} \end{cases}$$

Solving this problem is a challenge for three main reasons:

- The indicator function $1_{\mathcal{S}}$ which is non-convex → replace by surrogate loss function
- The presence of the DNN f that is non-linear → approximate solution is enough
-

Optimization Problem

$$\underset{\substack{D=[D_1,...,D_M]\in\mathbb{R}^{P\times M}\\ V=[\mathbf{v}^{(1)},...\mathbf{v}^{(N)}]\in\mathbb{R}^{M\times N}}}{\text{approx maximize}} \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(C_f(\mathbf{x}^{(i)'}), C_f(\mathbf{x}^{(i)}))$$

$$\text{s.t.} \quad \begin{cases} \mathbf{x}^{(i)'} = \mathbf{x}^{(i)} + D\mathbf{v}^{(i)} \in \mathcal{X} & ,(\forall i \in \{1,\dots,N\}) \quad \textit{Valid examples} \\ \|D\mathbf{v}^{(i)}\|_p \leq \delta_p & ,(\forall i \in \{1,\dots,N\}) \quad \textit{Small perturbations} \\ \|D_j\|_p = 1 & ,(\forall j \in \{1,\dots,M\}) \quad \textit{Normalized directions} \end{cases}$$

Solving this problem is a challenge for three main reasons:

- The indicator function $1_{\mathcal{S}}$ which is non-convex → replace by surrogate loss function
- The presence of the DNN f that is non-linear → approximate solution is enough
- The 3 constraints

Optimization Problem

$$\underset{\substack{D=[D_1,\dots,D_M]\in\mathbb{R}^{P\times M}\\ V=[\mathbf{v}^{(1)},\dots\mathbf{v}^{(N)}]\in\mathbb{R}^{M\times N}}}{\text{approx maximize}} \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(C_f(\mathbf{x}^{(i)'}),C_f(\mathbf{x}^{(i)}))$$

$$\text{s.t.} \quad \begin{cases} \mathbf{x}^{(i)'} = \mathbf{x}^{(i)} + D\mathbf{v}^{(i)} \in \mathcal{X} & ,(\forall i \in \{1,\dots,N\}) \quad \textit{Valid examples} \\ \|D\mathbf{v}^{(i)}\|_p \le \delta_p & ,(\forall i \in \{1,\dots,N\}) \quad \textit{Small perturbations} \\ \|D_j\|_p = 1 & ,(\forall j \in \{1,\dots,M\}) \quad \textit{Normalized directions} \end{cases}$$

Solving this problem is a challenge for three main reasons:

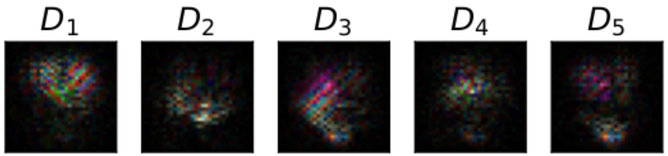
- The indicator function $1_{\mathcal{S}}$ which is non-convex → replace by surrogate loss function
- The presence of the DNN f that is non-linear → approximate solution is enough
- The 3 constraints → we propose 2 different relaxations

Numerical Experiments

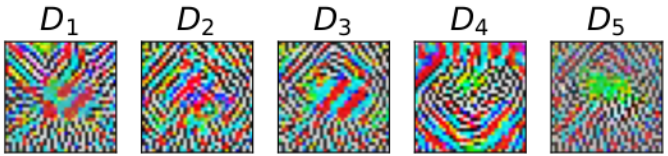
Visualisation of Adversarial Directions

Setting: Attack a VGG11 (top) or robust ResNet50 (bottom) on CIFAR10. Learn $M = 5$ directions.

ℓ_2-attack

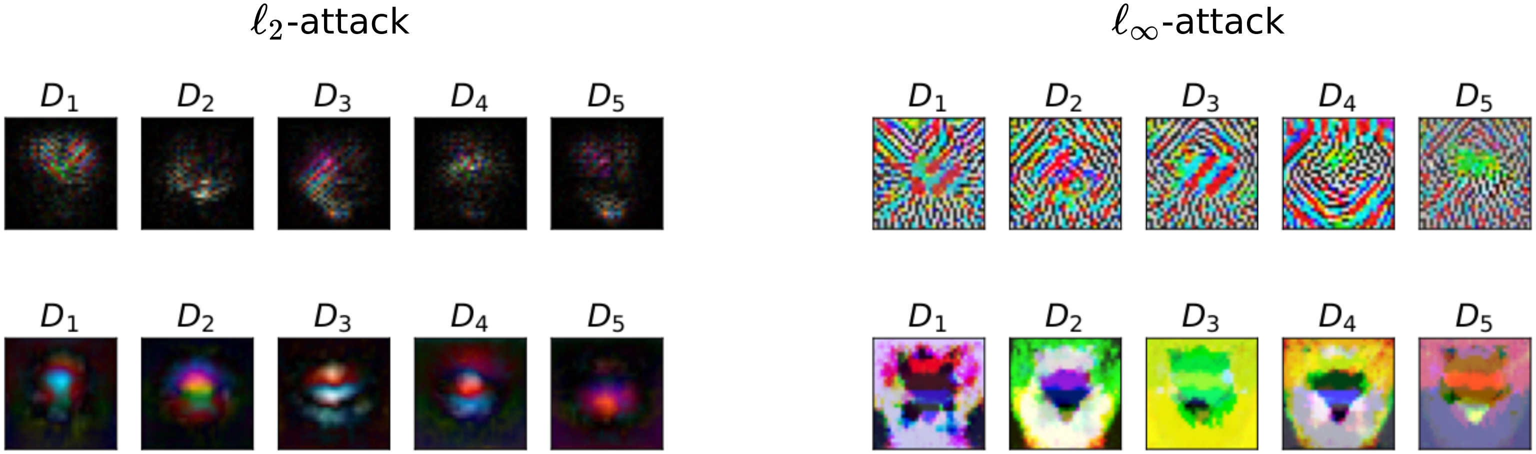


ℓ_∞-attack



Visualisation of Adversarial Directions

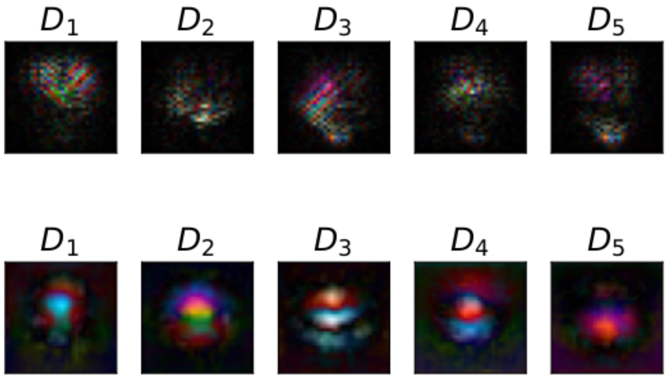
Setting: Attack a VGG11 (top) or robust ResNet50 (bottom) on CIFAR10. Learn $M = 5$ directions.



Visualisation of Adversarial Directions

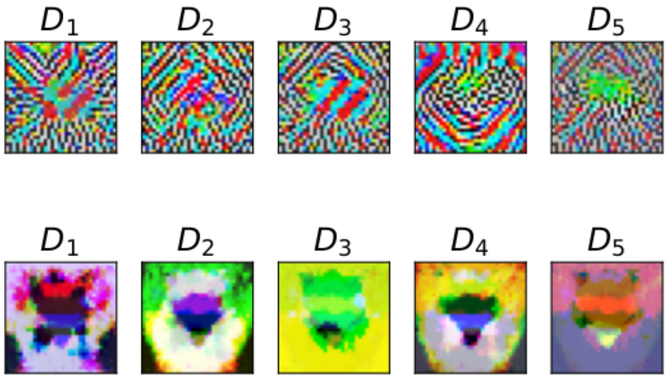
Setting: Attack a VGG11 (top) or robust ResNet50 (bottom) on CIFAR10. Learn $M = 5$ directions.

ℓ_2-attack



Mostly local spots

ℓ_∞-attack

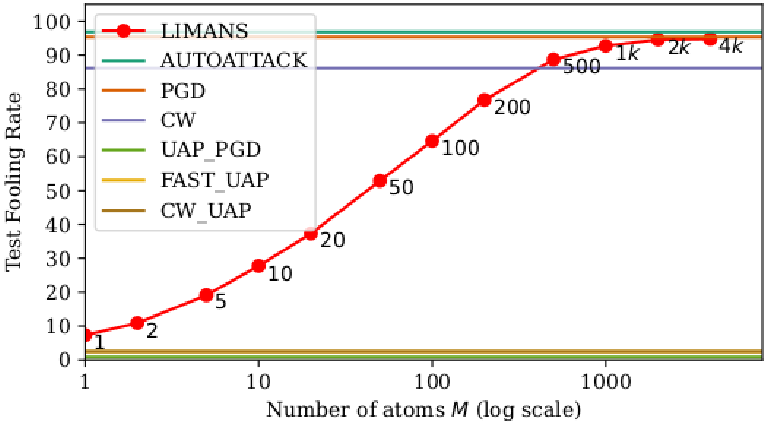


Mostly edges and corners

Having a linear model of the adversarial noise space allows for visual inspection of the adversarial directions, which is advantageous for understanding the attack behavior.

Impact of the Number of Directions

Setting: Attack a VGG11 on CIFAR10 with ℓ_2-attacks.



Specific: AutoAttack, PGD, CW

Universal: UAP PGD, FAST UAP, CW UAP

As M increases, LIMANS progressively narrows the performance gap with specific attacks

Transferability

Setting: Attack a VGG1 on CIFAR10. Evaluate fooling performance on target classifiers (columns).

	MobileNet	ResNet50	DenseNet	VGG	R-r18	R-wrn-34-10
AutoAttack	62.5	43.0	44.0	100	2.7	2.7
VNI-FGSM	69.3	62.6	61.4	96.5	3.0	2.6
NAA	42.3	14.5	1.8	71.6	1.6	1.2
RAP	46.5	39.5	40.9	73.8	3.3	3.4
Ours	97.4	87.5	81.5	91.0	11.5	12.6

Transferability

Setting: Attack a VGG1 on CIFAR10. Evaluate fooling performance on target classifiers (columns).

	MobileNet	ResNet50	DenseNet	VGG	R-r18	R-wrn-34-10
AutoAttack	62.5	43.0	44.0	100	2.7	2.7
VNI-FGSM	69.3	62.6	61.4	96.5	3.0	2.6
NAA	42.3	14.5	1.8	71.6	1.6	1.2
RAP	46.5	39.5	40.9	73.8	3.3	3.4
Ours	97.4	87.5	81.5	91.0	11.5	12.6

AutoAttack performs best when **source classifier = target classifier** (e.g. VGG)

Transferability

Setting: Attack a VGG1 on CIFAR10. Evaluate fooling performance on target classifiers (columns).

	MobileNet	ResNet50	DenseNet	VGG	R-r18	R-wrn-34-10
AutoAttack	62.5	43.0	44.0	100	2.7	2.7
VNI-FGSM	69.3	62.6	61.4	96.5	3.0	2.6
NAA	42.3	14.5	1.8	71.6	1.6	1.2
RAP	46.5	39.5	40.9	73.8	3.3	3.4
Ours	97.4	87.5	81.5	91.0	11.5	12.6

AutoAttack performs best when **source classifier = target classifier** (e.g. VGG)

Our model yields better transferability performance, i.e. **source classifier $\neq$ target classifier**

Bypassing Attack Detectors

Setting: Attack a VGG11 on CIFAR10. Train systems to detect adversarial attacks (columns)

Classifiers / Detectors	detect FGSM	detect PGD	detect AutoAttack	detect LIMANS 10
FGSM	91.1	91.1	91.1	83.4
LIMANS$_{10}$	75.7	81.0	81.6	88.9
LIMANS$_{500}$	17.5	25.6	31.8	26.6
LIMANS$_{1000}$	15.9	26.1	32.1	**21.7**
LIMANS$_{4000}$	**15.6**	**23.7**	**28.2**	31.1

RAUD (*Robust Accuracy Under Defense*): quantifies the percentage of successful attacks detected

(the lower, the better)

Bypassing Attack Detectors

Setting: Attack a VGG11 on CIFAR10. Train systems to detect adversarial attacks (columns)

Classifiers / Detectors	detect FGSM	detect PGD	detect AutoAttack	detect LIMANS 10
FGSM	91.1	91.1	91.1	83.4
LIMANS_{10}	75.7	81.0	81.6	88.9
LIMANS_{500}	17.5	25.6	31.8	26.6
LIMANS_{1000}	15.9	26.1	32.1	**21.7**
LIMANS_{4000}	**15.6**	**23.7**	**28.2**	31.1

RAUD (*Robust Accuracy Under Defense*): quantifies the percentage of successful attacks detected

(the lower, the better)

LIMANS attacks consistently evade detection

outperforming specific attacks even at $M = 10$ and exhibiting robustness from $M \geq 500$

Conclusion

Conclusion

LIMANS

Linear Modeling of the Adversarial Noise Space

$$\mathbf{x}^{(i)'} = \mathbf{x}^{(i)} + D\mathbf{v}^{(i)}$$

Experimental findings:

- Bridge the gap between specific and universal attacks
- Allows visual inspection of the learned directions
- Show great transferability
- Bypass adversarial detectors

Thank you for your attention!

Questions?

Download the paper

Take-home message: Attacks are framed as specific linear combinations of universal adversarial directions