What Matters in Model Training to Transfer Adversarial Examples

Martin Gubri

Dissertation Defence Committee

Prof. Dr. Yves Le Traon	Supervisor	University of Luxembourg
Prof. Dr. Michail Papadakis	Chairman	University of Luxembourg
Dr. Maxime Cordy	Vice-Chairman	University of Luxembourg
Prof. Dr. Seong Joon Oh	Member & Reviewer	University of Tübingen
Prof. Dr. Florian Tramèr	Member & Reviewer	ETH Zürich

uni.lu | SnT

PhD Defence
21/06/2023

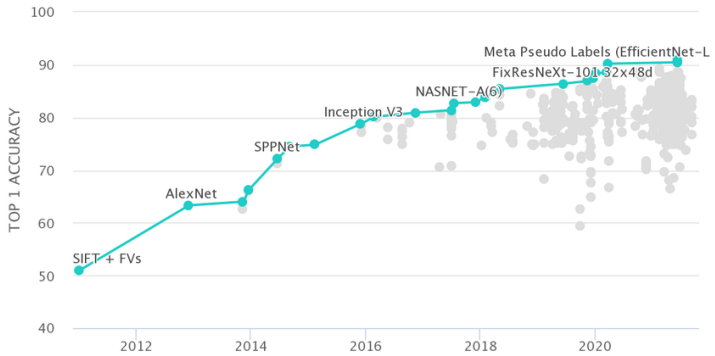
Fonds National de la Recherche Luxembourg

Context

The success of deep learning

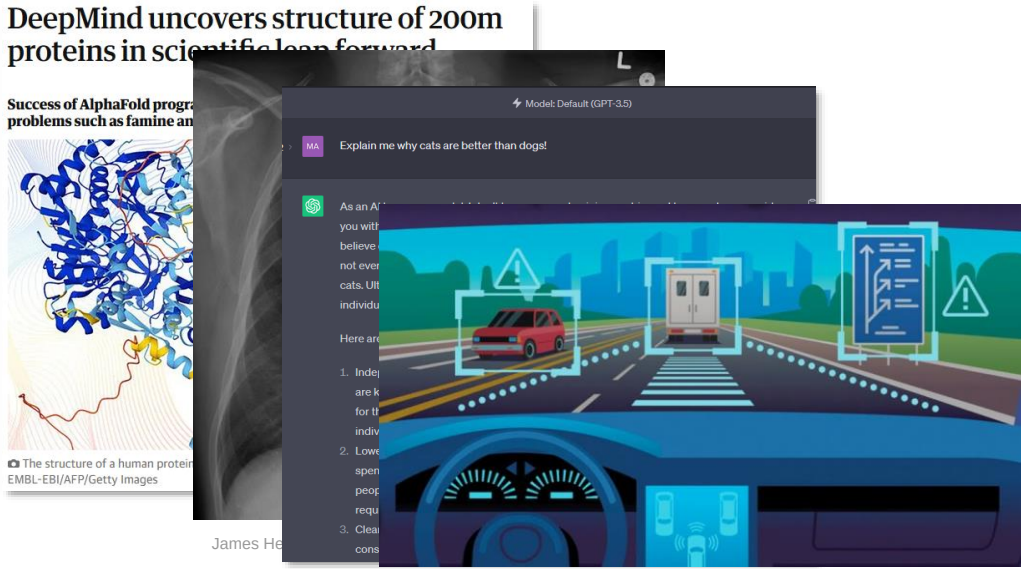
Deep Neural Networks (DNNs) have made huge progress in numerous fields.

Benchmark datasets



Evolution of top-1 accuracy on ImageNet (paperwithcode)

Real world applications



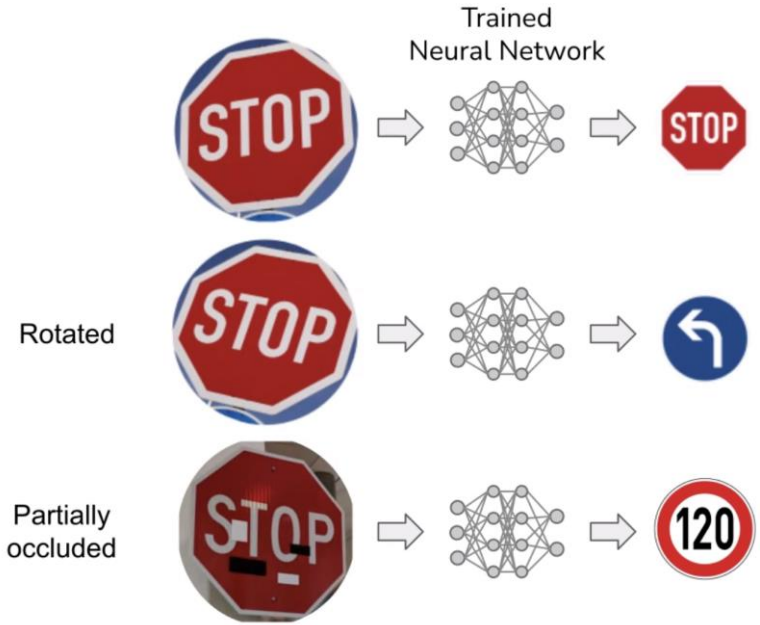
James He

Illustration from freeCodeCamp.org

Context

Critical failures

But deep neural networks can fail critically when faced with unexpected data.



Content filtering

Surveillance system

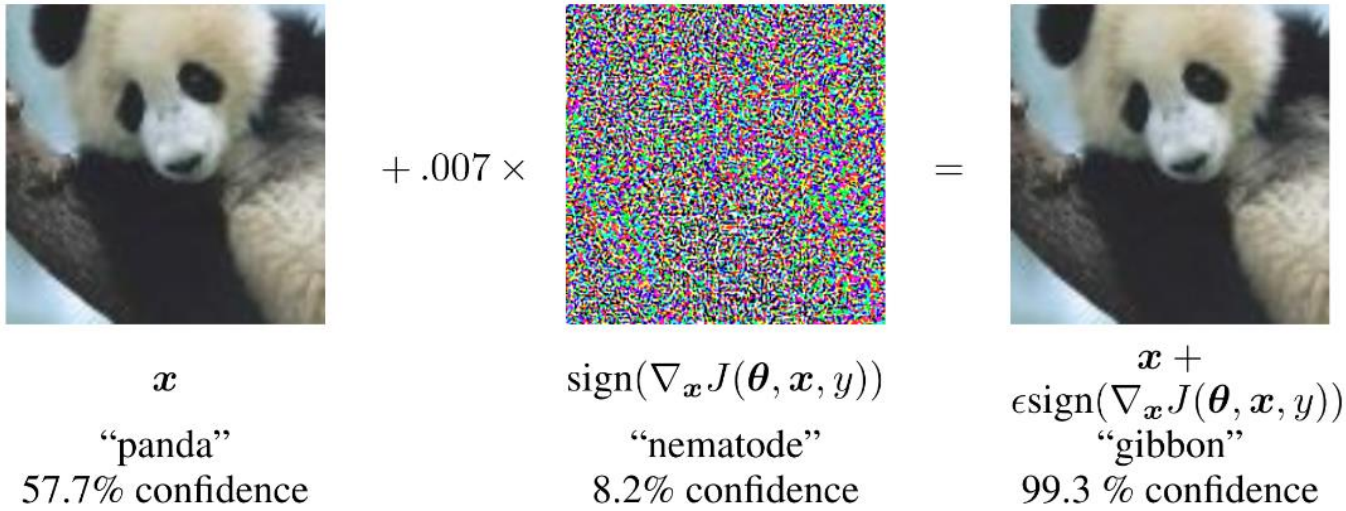
Fischer et al., Certified Defense to Image Transformations via Randomized Smoothing, CoRR 2019
K. Eykholt, et al. Robust Physical-World Attacks on Deep Learning Visual Classification, IEEE 2018
Illustrations from Mark Müller

The New York Times, 'On YouTube Kids, Startling Videos Slip Past Filters'
Wired, 'Children's YouTube is still churning out blood, suicide and cannibalism'
The Guardian, 'Hiding in plain sight: activists don camouflage to beat Met surveillance'.
Photograph: Cocoa Laney/The Observer

Background

Adversarial examples

Carefully crafted imperceptible perturbations applied to inputs to create large changes in output.



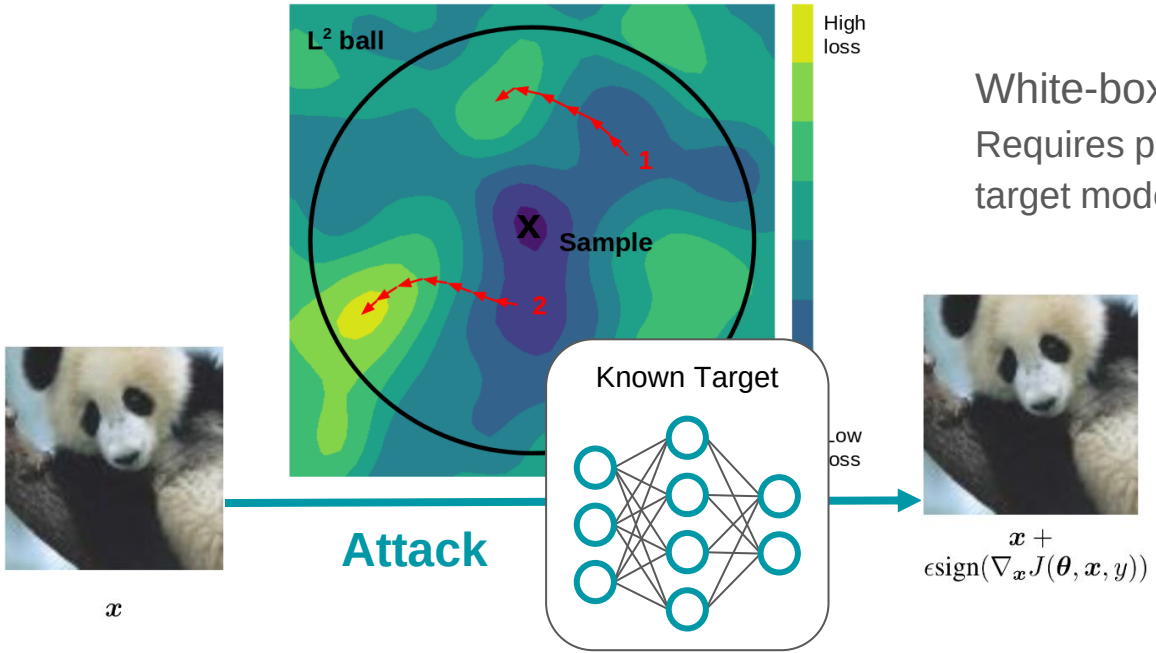
→ Worst-case distributional shift

Goodfellow, I. et al. (2014). Explaining and Harnessing Adversarial Examples

Background

PGD

Projected gradient ascent in the input space.



White-box attack

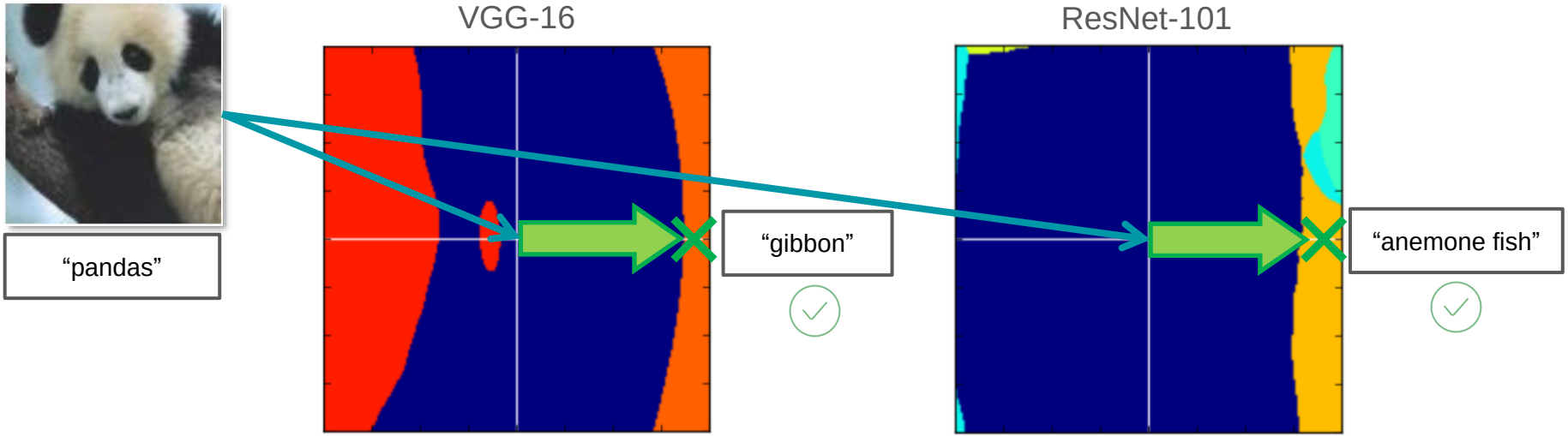
Requires perfect knowledge of the target model

Source Image: towardsdatascience.com

Background

Transferability

An adversarial example against a model is likely to be also adversarial against another model.



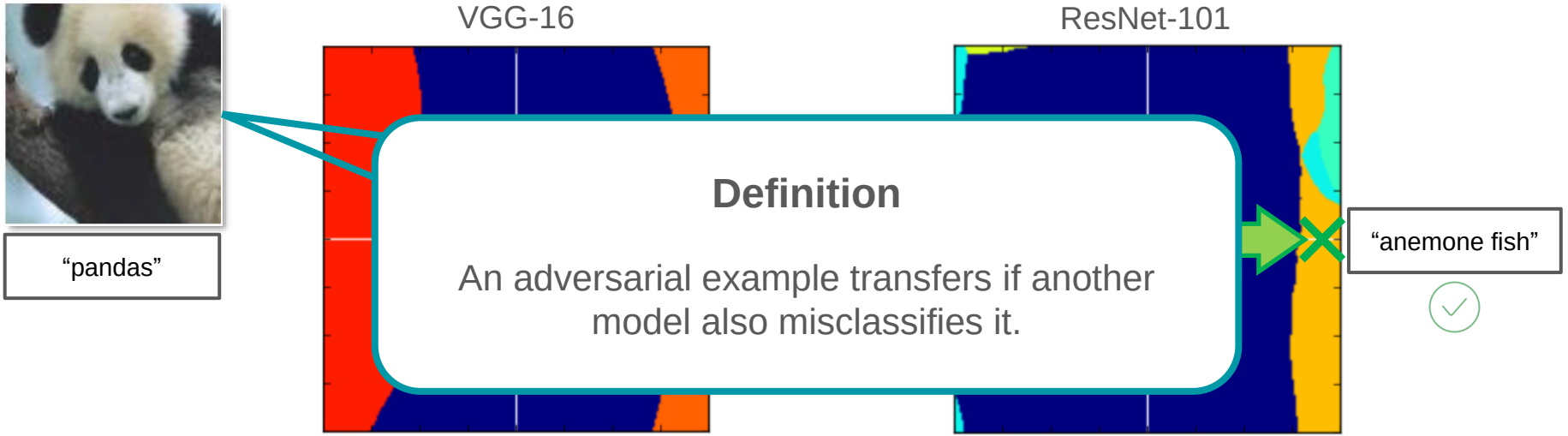
Decision boundaries around an ImageNet example. One color per predicted label.

Y. Liu, et al., Delving into Transferable Adversarial Examples and Black-box Attacks, ICLR 2017

Background

Transferability

An adversarial example against a model is likely to be also adversarial against another model.



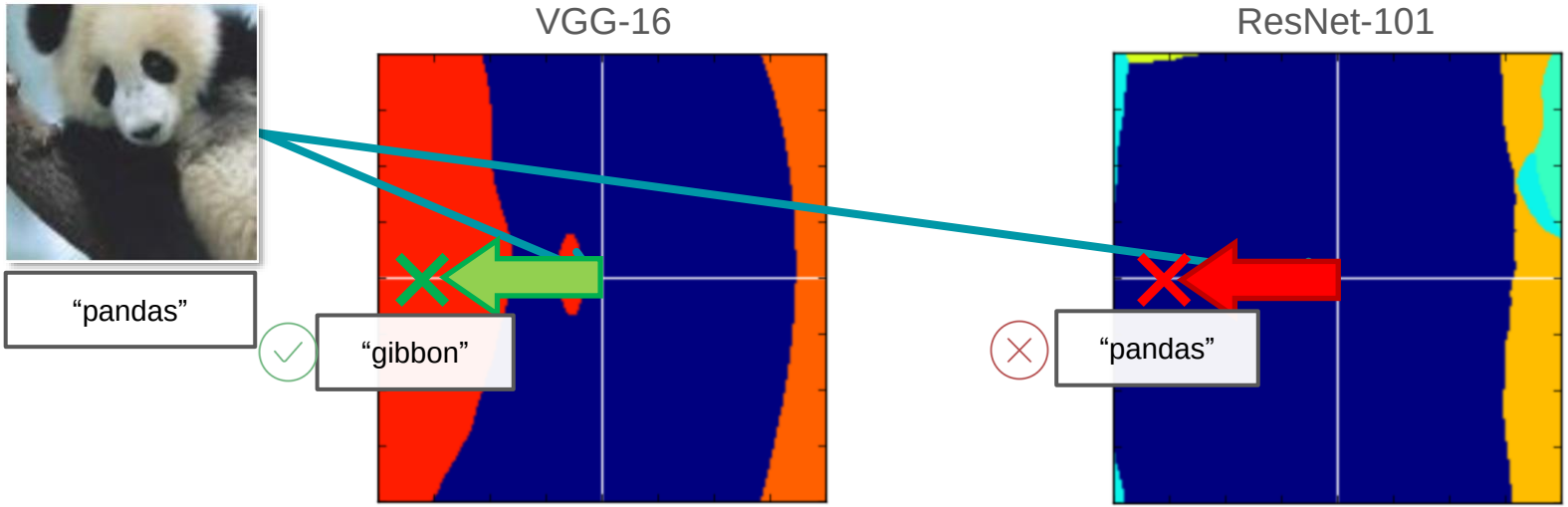
Decision boundaries around an ImageNet example. One color per predicted label.

Y. Liu, et al., Delving into Transferable Adversarial Examples and Black-box Attacks, ICLR 2017

Background

But transfer may fail

Adversarial examples easily overfit the surrogate model



Decision boundaries around an ImageNet example. One color per predicted label.

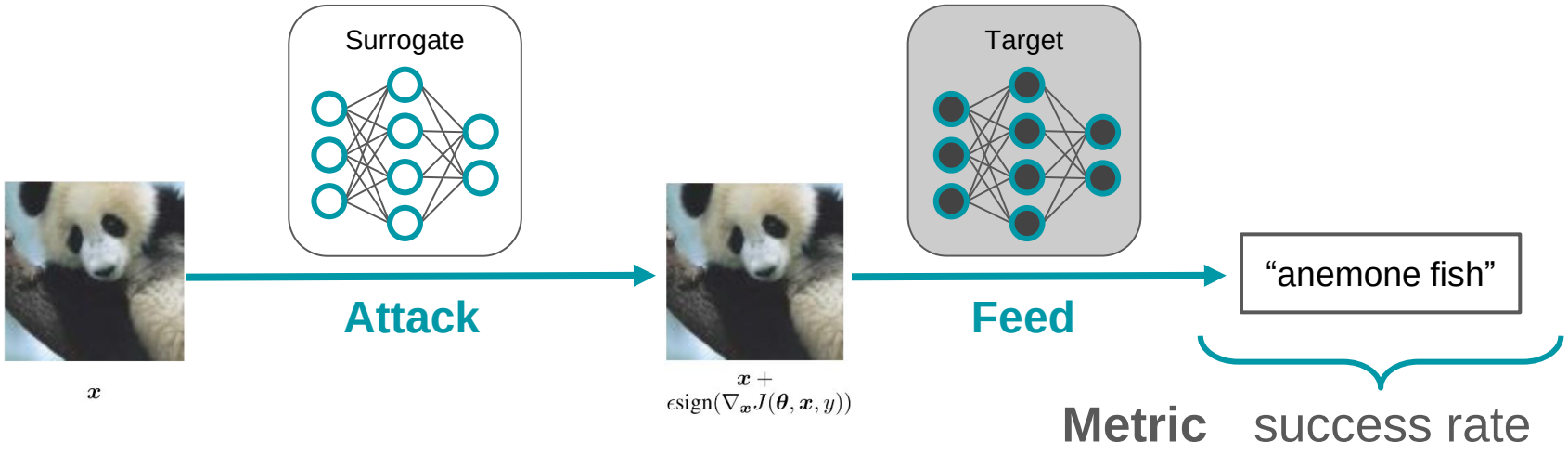
Y. Liu, et al., Delving into Transferable Adversarial Examples and Black-box Attacks, ICLR 2017

Background

Transfer-based black-box attack

More realistic attacks require less knowledge on the target.

Does **not** require to query the model.

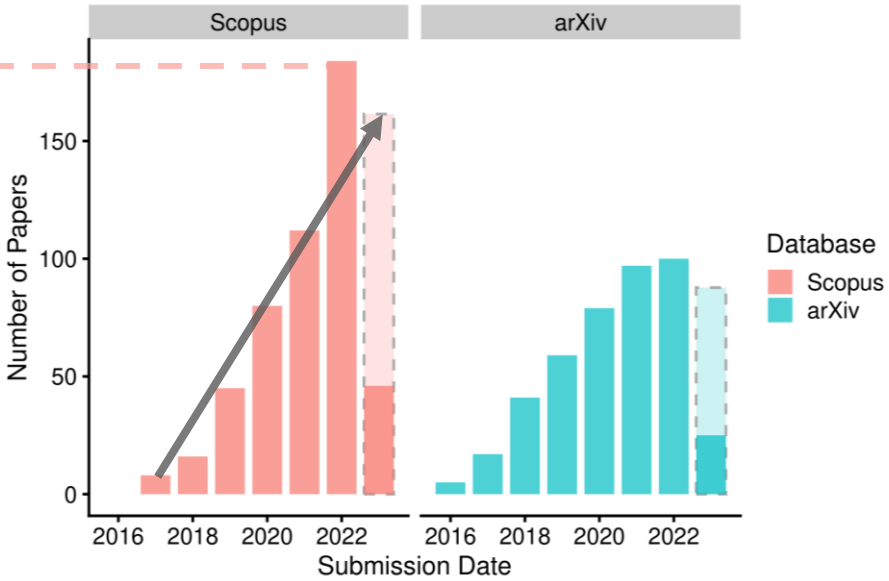


Background

Transferability, a prolific topic

184 published articles in 2022

Annual growth rate: 33.9%



Number of published (Scopus)
and submitted (arXiv) articles on transferability

Challenges

1. Knowledge gap

In 2021, a single publication studied how to improve the training of surrogate models: Liu et al. (2017) show that ensembling architectures improves transferability.

→ No prior contribution about how to train each surrogate architecture for transferability

2. High training computational cost

Training the surrogate model is at least **two** orders of magnitude more costly than attacking it.

→ Training the surrogate is the most costly

3. Lack of insights

The field tends to be a collection of scattered techniques with limited scientific value.

→ Lack of in-depth knowledge

4. Low success rate of small perturbations

Large perturbations are more transferable, but more visible and may even change the label.

Example of large L∞ norm perturbations that change label (Addepalli et al., 2022)

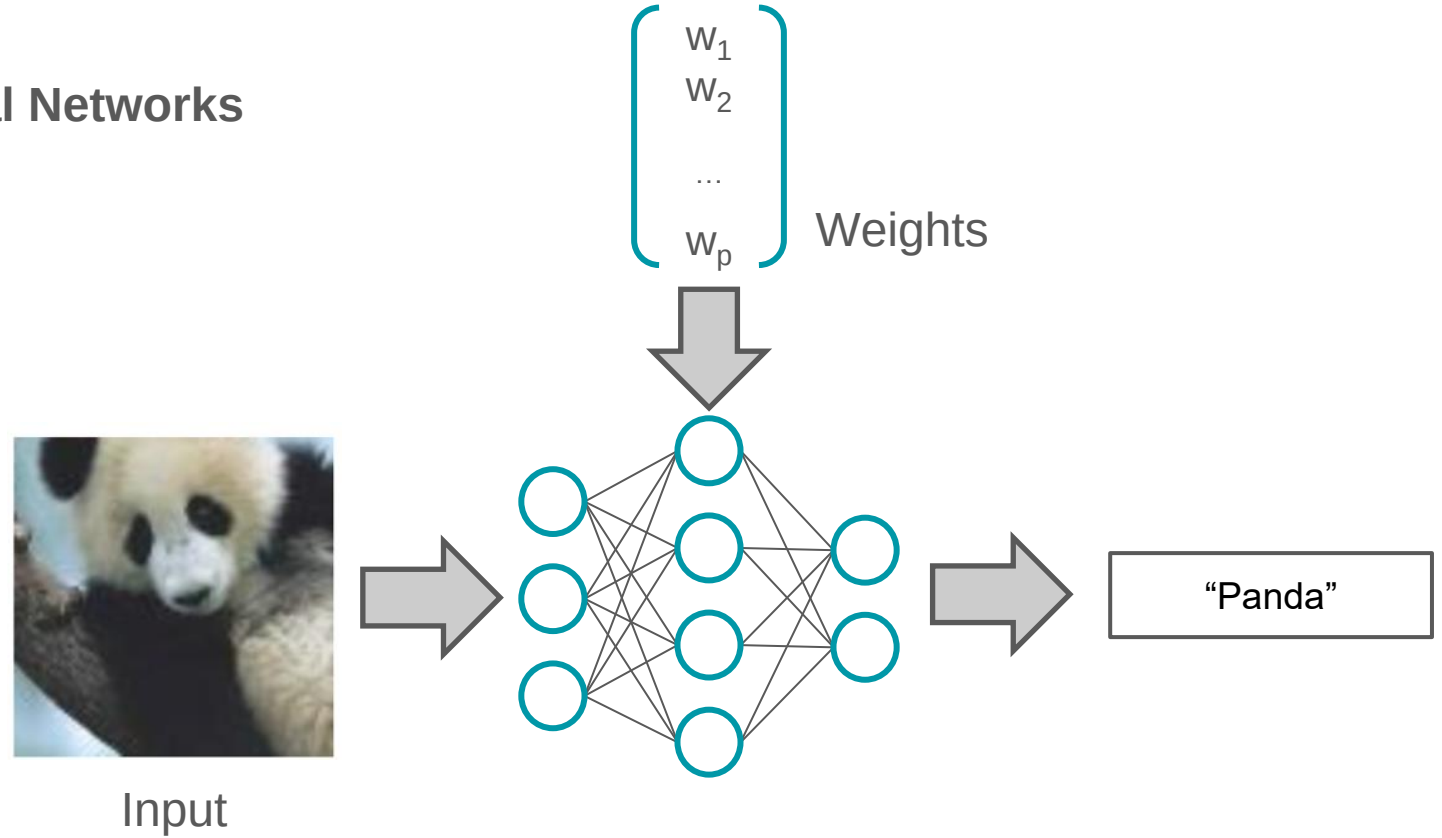
5. Over-specific transferability techniques

Many transferability techniques exploit specific components (skip connection, dropout).

→ Lack of exploitation of generic principles

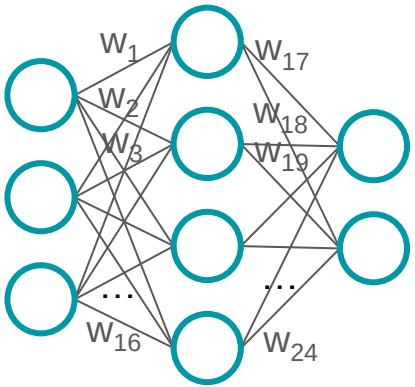
Context

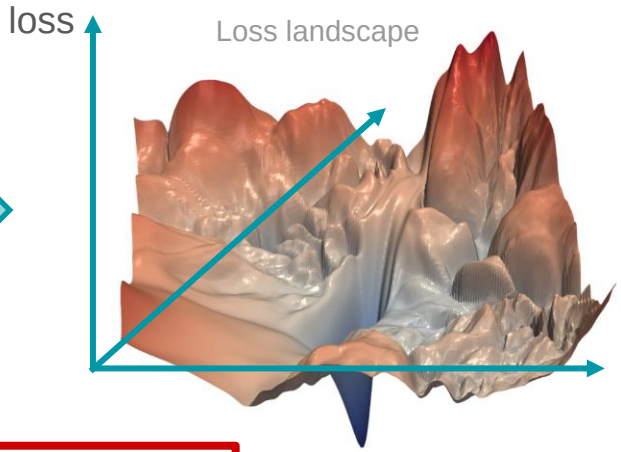
Neural Networks



Weight space

The weight space (or the parameter space) is the Euclidian space formed by all the weights a neural network.



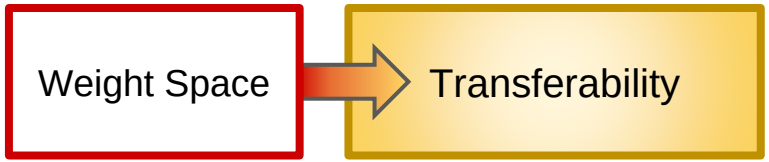


Weight Space

Hao Li, et al. (2018). Visualizing the Loss Landscape of Neural Nets. *NeurIPS 2018.*

Scope & goal

Goal



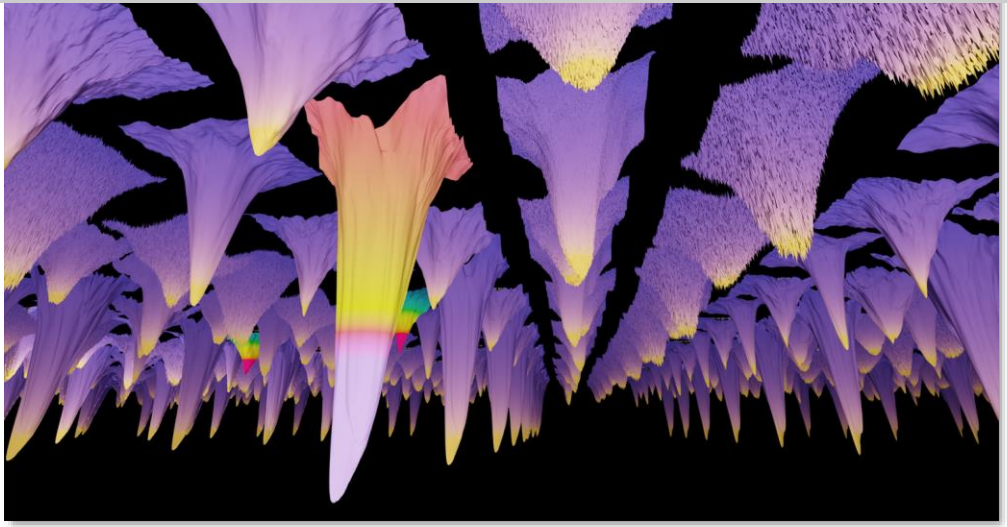
Scope

How to train surrogate models for black-box evasion attacks

Weight space exploration

Multiple modes
of the loss landscape.

losslandscape.com



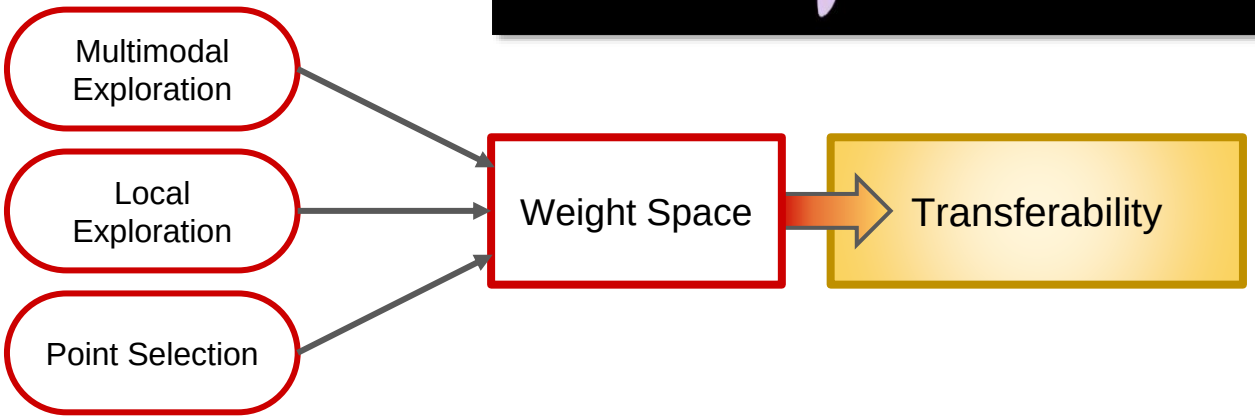
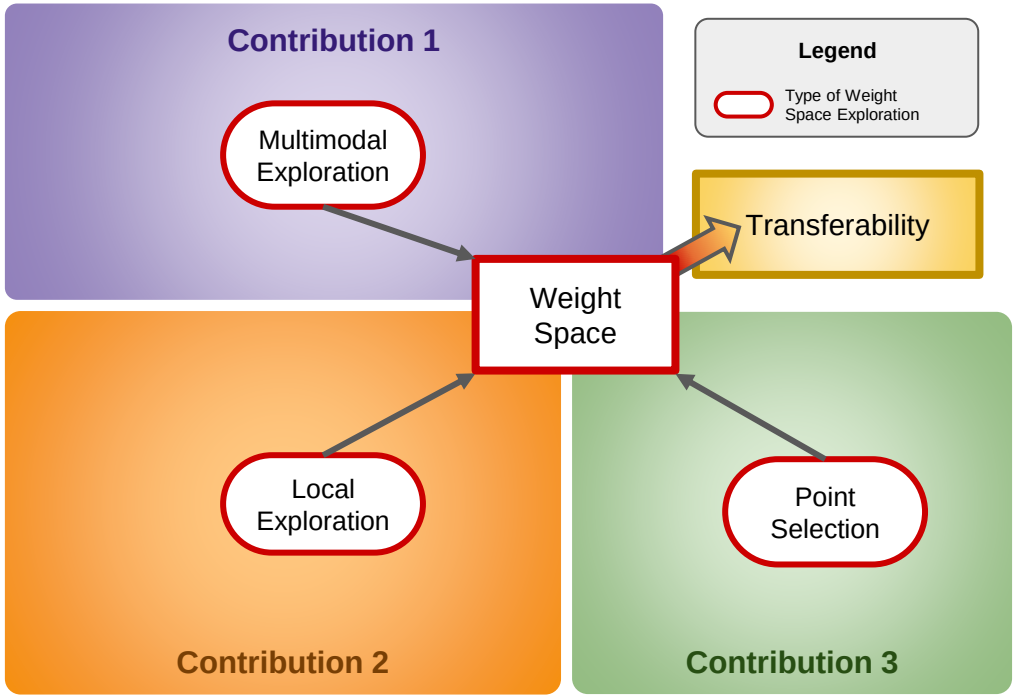


Table of contents



Probabilistic Transferability

Transferability from
Deep Ensemble &
Bayesian Neural Network

Efficient and Transferable Adversarial Examples from Bayesian Neural Networks

Martin Gubri[1] **Maxime Cordy**[1] **Mike Papadakis**[1] **Yves Le Traon**[1] **Koushik Sen**[2]

[1]Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, LU
[2]University of California, Berkeley, CA, USA

Abstract

An established way to improve the transferability of black-box evasion attacks is to craft the adversarial examples on an ensemble-based surrogate to increase diversity. We argue that transferability is fundamentally related to uncertainty. Based on a state-of-the-art Bayesian Deep Learning technique, we propose a new method to efficiently build a surrogate by sampling approximately from the posterior distribution of neural network weights, which represents the belief about the value of each parameter. Our extensive experiments on ImageNet and CIFAR-10 show that our approach improves the transfer rates of four state-of-the-art attacks significantly (up to 62.1 percentage points), in both intra-architecture and inter-architecture cases. On ImageNet, our approach can reach 94% of transfer rate while reducing training computations from 11.6 to 2.4 exaflops, compared to an ensemble of independently trained DNNs. Our vanilla surrogate achieves 87.5% of the time higher transferability than 3 test-time techniques designed for this purpose. Our work demonstrates that the way to train a surrogate has been overlooked, although it is an important element of transfer-based attacks. We are, therefore, the first to review the effectiveness of several training methods in increasing transferability. We provide new directions to better understand the transferability phenomenon and offer a simple but strong baseline for future work.

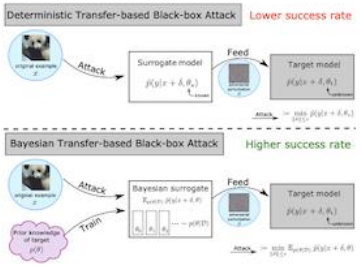


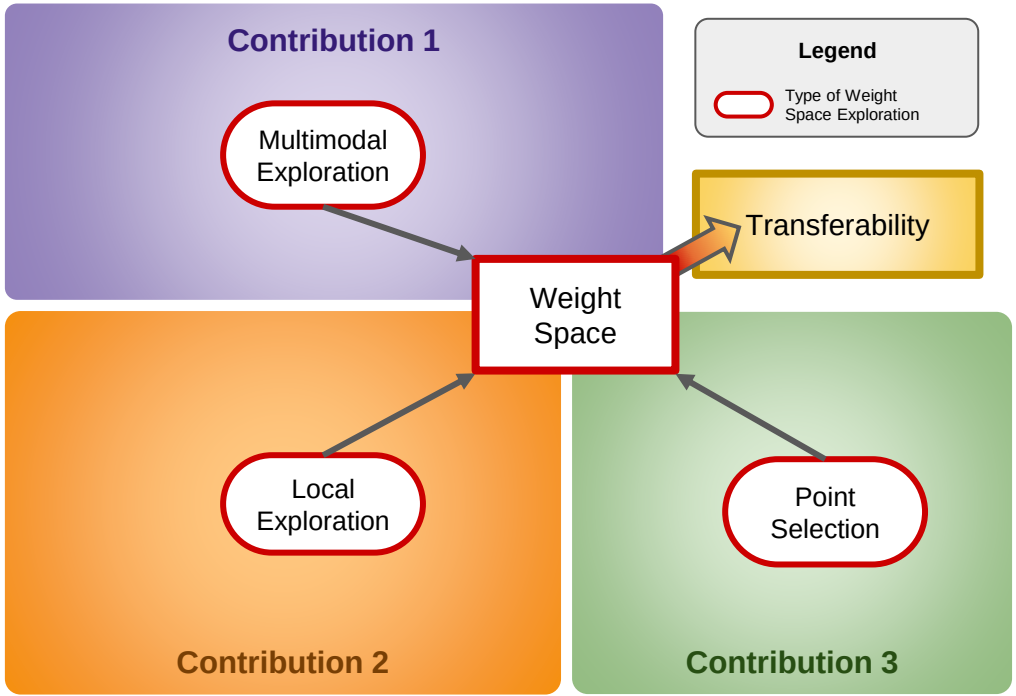
Figure 1: Illustration of the proposed approach.

els is that they are vulnerable to adversarial examples, i.e., misclassified examples that result from slightly altering a well-classified example at test time [Biggio et al., 2013, Szegedy et al., 2013]. This constitutes a critical security threat, as a malicious third party may exploit this property to enforce some desired outcome.

Such *adversarial attacks* have been primarily designed in white-box settings, where the attacker is assumed to have complete knowledge of the target DNN (incl. its weights). While studying such worst-case scenarios is essential for proper security assessment, in practice the attacker should have limited knowledge of the target model. In such a case, the adversarial attack is applied to a surrogate model, with the hope that the crafted adversarial examples *transfer to* (i.e., are also misclassified by) the target DNN.

Accepted at UAI 2022

Scope and goal



Scope and goal

Focus

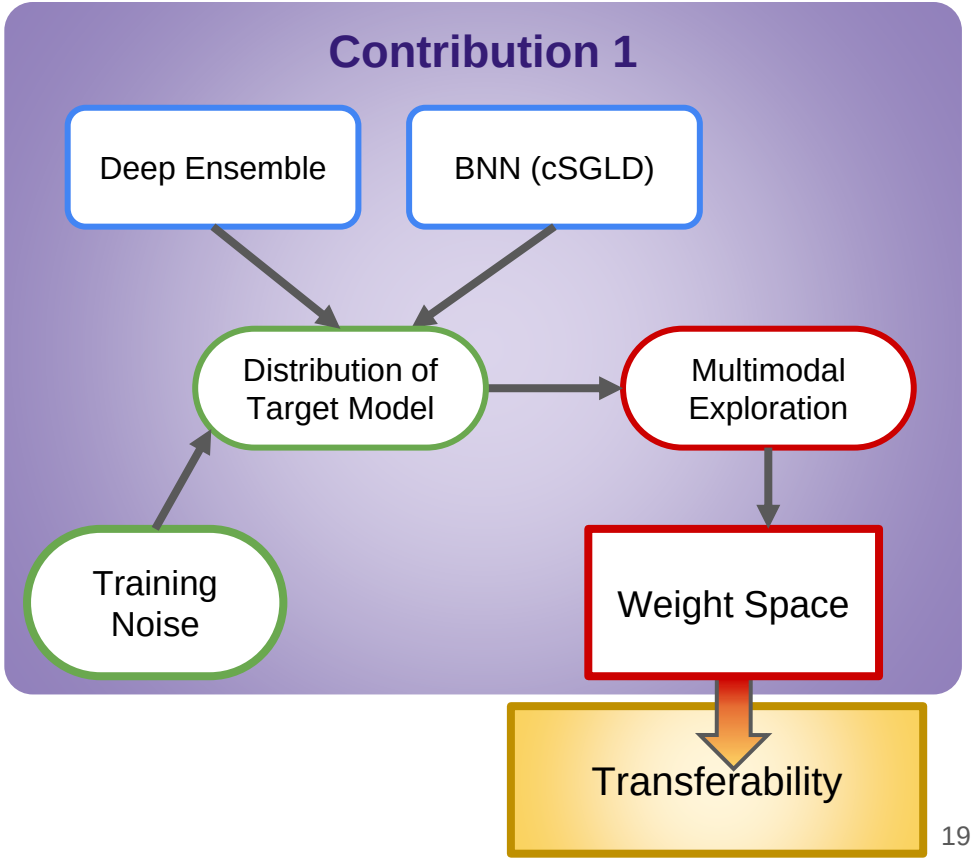
Train surrogate models from several vicinities of the weight space

Question

Are representations from different modes different enough to hinder transferability?

What matters?

Training noise through probabilistic distribution of the target



Probabilistic view on transferability

Threat model

Known on the target model

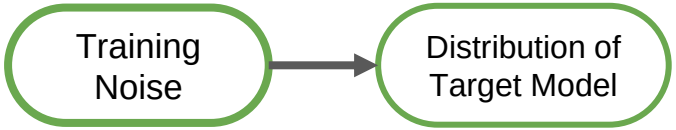
- its **architecture**,
- its training **dataset**,
- its (maximum likelihood) **optimizer**,
- its **prior** of its parameters (weight decay).

Unknown on the target model

- its **weights**.

Uncertainty of the target weights arises from stochasticity in training (random batches & random initialization).

The weights of the target model admit a probability distribution.



Extension to unknown architecture in the thesis.

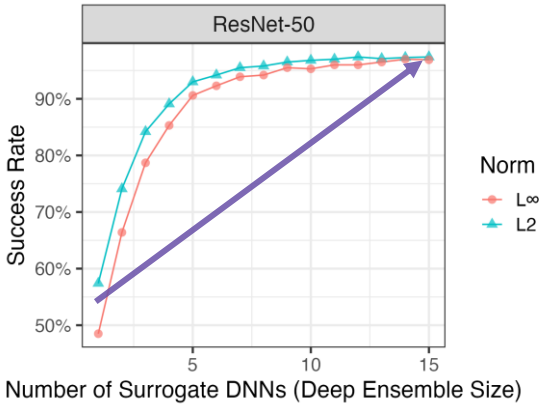
Deep Ensemble

Deep Ensemble (Lakshminarayanan, 2017) ensemble of independently trained DNNs (same architecture).

Sample from the distribution of the target model under our threat model → Deep ensemble surrogate

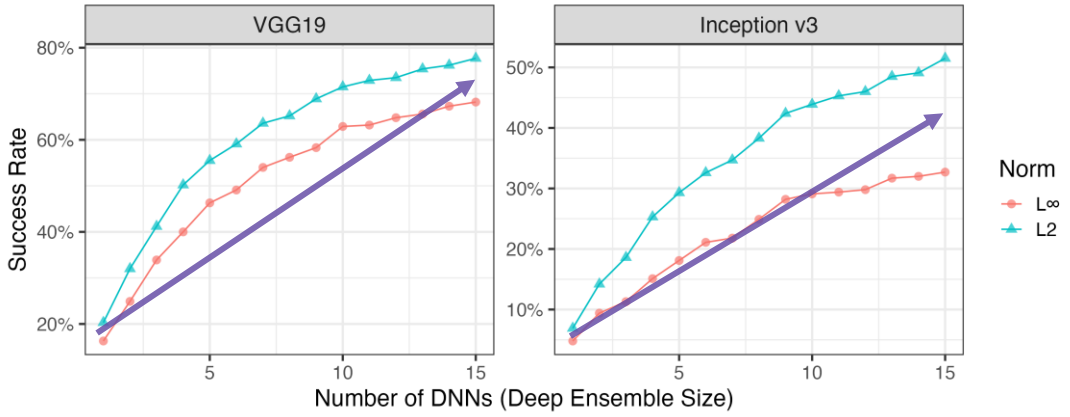
Intra-architecture transferability

From ResNet-50 to ResNet-50



Inter-architecture transferability

From ResNet-50 to other architectures



Same conclusion on six other target architectures

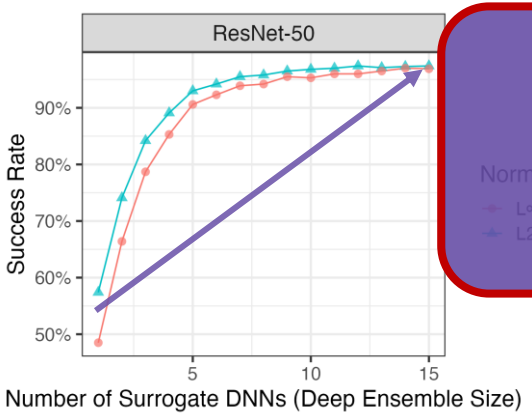
Deep Ensemble

Deep Ensemble (Lakshminarayanan, 2017) ensemble of independently trained DNNs (same architecture).

Sample from the distribution of the target model under our threat model → Deep ensemble surrogate

Intra-architecture transferability

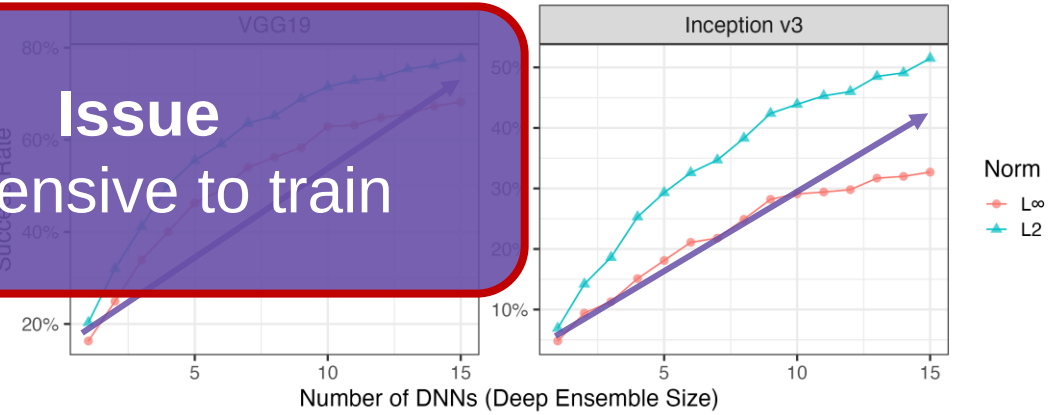
From ResNet-50 to ResNet-50



Issue
Expensive to train

Inter-architecture transferability

From ResNet-50 to other architectures

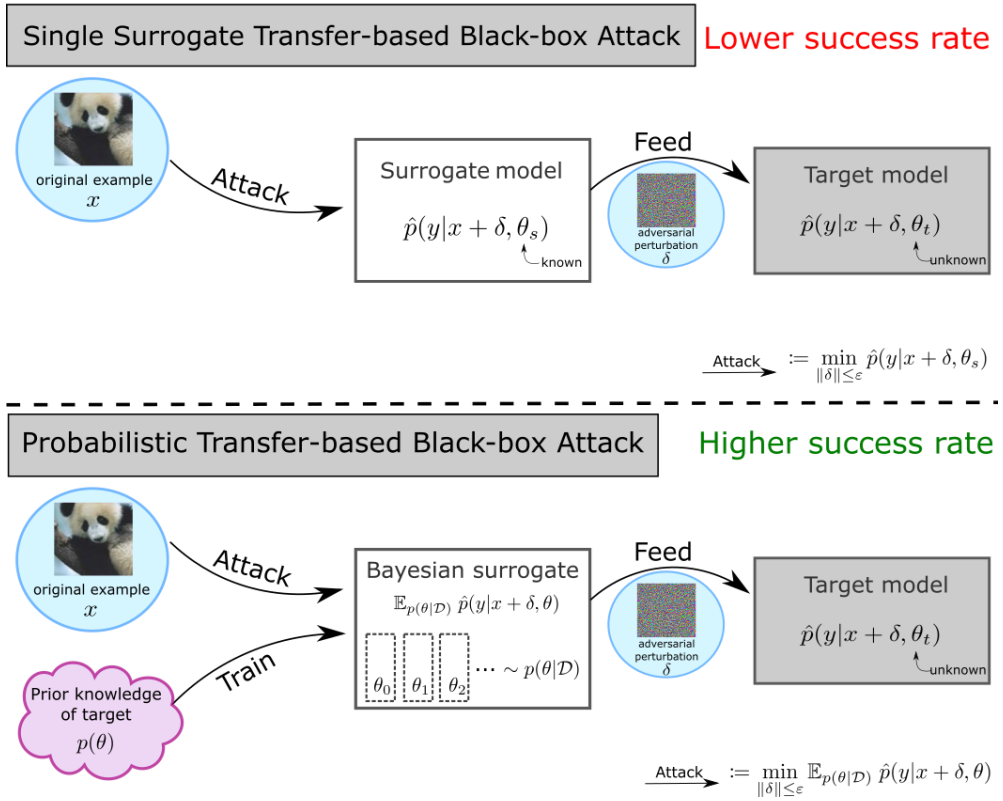


Same conclusion on six other target architectures

Bayesian Neural Networks

Mingard et al. [2020] observes a strong correlation between the probability to obtain a function consistent with the training set by SGD, and the Bayesian posterior.

The target weights are approximately distributed according to the posterior.



Bayesian Neural Networks

cSGLD (Zhang, 2020) Bayesian method that adds noise to the weights to sample from the posterior.

Efficient

At least **2.5 times** less computational cost to achieve the same success rate.

Dataset	Attack	Norm	T-DEE	Flops Ratio
ImageNet	I-FGSM	L2	4.91 ±0.11	2.84 ±0.06
		L∞	4.34 ±0.13	2.51 ±0.08
	MI-FGSM	L2	4.69 ±0.18	2.71 ±0.10
		L∞	4.38 ±0.03	2.53 ±0.02
	PGD	L2	5.00 ±0.11	2.89 ±0.06
		L∞	4.42 ±0.16	2.56 ±0.09
	FGSM	L2	5.81 ±0.34	3.35 ±0.19
		L∞	5.98 ±0.03	3.46 ±0.02
CIFAR10	I-FGSM	L2	>15 ±nan	>15 ±nan
		L∞	3.76 ±0.08	3.76 ±0.08
	MI-FGSM	L2	5.56 ±0.80	5.56 ±0.80
		L∞	2.88 ±0.03	2.87 ±0.03
	PGD	L2	>15 ±nan	>15 ±nan
		L∞	3.74 ±0.12	3.74 ±0.12
	FGSM	L2	>15 ±nan	>15 ±nan
		L∞	8.72 ±0.01	8.72 ±0.01
MNIST	I-FGSM	L2	>15 ±nan	>15 ±nan
		L∞	3.42 ±0.17	3.42 ±0.17
	MI-FGSM	L2	>15 ±nan	>15 ±nan
		L∞	2.79 ±0.07	2.79 ±0.07
	PGD	L2	>15 ±nan	>15 ±nan
		L∞	3.26 ±0.28	3.26 ±0.28
	FGSM	L2	>15 ±nan	>15 ±nan
		L∞	>15 ±nan	>15 ±nan

Transferability techniques

1. Competitive

→ Exploring the weight space is of utmost importance

Norm	Surrogate	Target Architecture: RN50	RNX50	DN121	MNASN	EffNetB0
L2	1 DNN	56.60 ±0.71	41.09 ±0.61	29.73 ±0.30	28.13 ±0.17	16.64 ±0.33
	+ DI	83.15 ±0.30	73.17 ±0.80	61.24 ±0.58	58.16 ±0.36	⋆42.10 ±0.36
	+ SGM	65.64 ±0.88	52.75 ±0.42	38.58 ±0.55	43.40 ±0.61	29.11 ±0.30
	+ GN	78.84 ±0.46	62.46 ±0.38	45.76 ±0.02	41.44 ±0.58	25.77 ±0.11
	+ MI	†52.53 ±0.80	†37.15 ±0.76	†26.33 ±0.48	†25.21 ±0.42	†14.74 ±0.31
	+ DI	80.81 ±0.72	69.55 ±0.83	56.73 ±0.39	54.16 ±0.05	37.07 ±0.03
	+ SGM	65.65 ±0.95	53.25 ±0.18	38.79 ±0.62	44.33 ±0.63	29.45 ±0.28
	+ GN	71.50 ±0.12	53.45 ±0.65	37.39 ±0.47	34.53 ±0.69	20.29 ±0.36
	cSGLD	84.83 ±0.55	74.73 ±0.82	71.45 ±0.56	60.14 ±0.44	39.71 ±0.20
	+ DI	**93.87 ±0.19**	**89.12 ±0.24**	**88.52 ±0.16**	**82.78 ±0.28**	**66.13 ±0.35**
	+ SGM	†83.17 ±0.85	†72.79 ±1.06	†66.19 ±0.89	71.71 ±0.41	52.66 ±0.31
	+ GN	92.99 ±0.13	85.69 ±0.24	82.81 ±0.42	72.88 ±0.30	50.30 ±0.29
	+ MI	†82.44 ±0.19	†70.93 ±1.04	†66.19 ±0.56	†55.51 ±0.59	†34.49 ±0.59
	+ DI	93.48 ±0.23	87.87 ±0.15	86.81 ±0.33	80.37 ±0.20	60.26 ±0.02
	+ SGM	†82.35 ±0.10	†71.54 ±0.58	†64.50 ±0.18	70.47 ±0.22	50.80 ±0.23
	+ GN	90.11 ±0.18	80.35 ±0.61	75.10 ±0.67	64.08 ±0.12	39.85 ±0.52
L∞	1 DNN	47.81 ±1.09	32.29 ±0.64	23.43 ±0.32	22.52 ±0.45	12.77 ±0.32
	+ DI	76.55 ±1.01	62.57 ±0.56	50.17 ±0.33	49.31 ±0.18	⋆32.64 ±0.09
	+ SGM	66.36 ±0.50	51.60 ±0.36	39.05 ±0.24	45.60 ±0.72	30.69 ±0.03
	+ GN	67.02 ±0.17	46.74 ±0.63	32.57 ±0.17	31.12 ±0.77	17.68 ±0.05
	+ MI	55.12 ±0.82	38.47 ±0.82	28.19 ±0.14	27.55 ±0.67	16.34 ±0.37
	+ DI	⋆82.47 ±0.41	⋆69.69 ±0.81	57.79 ±0.57	⋆55.99 ±0.37	⋆38.63 ±0.29
	+ SGM	68.39 ±0.53	54.57 ±0.60	41.48 ±0.37	47.97 ±0.41	⋆33.16 ±0.37
	+ GN	71.27 ±0.54	51.46 ±0.84	36.91 ±0.48	34.54 ±0.32	20.51 ±0.30
	cSGLD	78.71 ±1.19	65.11 ±1.45	61.49 ±0.59	51.81 ±1.45	31.11 ±0.99
	+ DI	90.03 ±0.10	82.13 ±0.45	81.19 ±0.34	74.48 ±0.39	53.51 ±0.39
	+ SGM	81.37 ±0.72	69.88 ±1.31	65.20 ±0.75	71.68 ±0.53	52.15 ±0.32
	+ GN	87.33 ±0.73	76.00 ±1.33	71.67 ±0.97	61.45 ±0.25	37.19 ±0.68
	+ MI	82.89 ±0.70	70.42 ±1.26	66.39 ±0.74	56.68 ±0.97	36.00 ±1.15
	+ DI	**93.97 ±0.26**	**87.69 ±0.44**	**86.78 ±0.16**	**81.08 ±0.14**	**60.87 ±0.48**
	+ SGM	84.19 ±0.21	73.14 ±0.99	67.35 ±0.26	74.36 ±0.47	55.30 ±0.16
	+ GN	89.53 ±0.05	78.69 ±0.19	73.33 ±0.58	63.56 ±0.35	39.79 ±0.52

Transferability techniques

1. Competitive

→ Exploring the weight space is of utmost importance

2. Complementary

→ The weight space leverages other properties of transferability

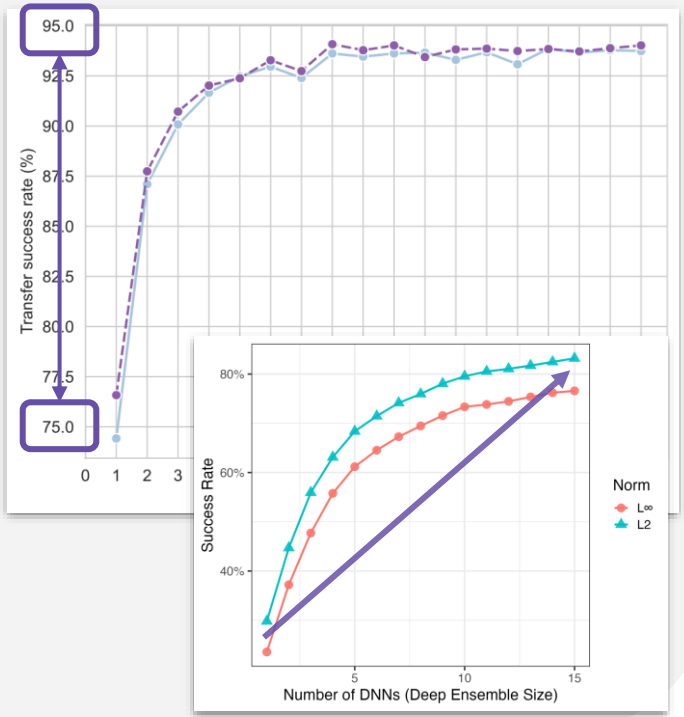
Norm	Surrogate	Target Architecture: RN50	RNX50	DN121	MNASN	EffNetB0
	1 DNN	56.60 ±0.71	41.09 ±0.61	29.73 ±0.30	28.13 ±0.17	16.64 ±0.33
	+ DI	83.15 ±0.30	73.17 ±0.80	61.24 ±0.58	58.16 ±0.36	⋆42.10 ±0.36
	+ SGM	65.64 ±0.88	52.75 ±0.42	38.58 ±0.55	43.40 ±0.61	29.11 ±0.30
	+ GN	78.84 ±0.46	62.46 ±0.38	45.76 ±0.02	41.44 ±0.58	25.77 ±0.11
	+ MI	†52.53 ±0.80	†37.15 ±0.76	†26.33 ±0.48	†25.21 ±0.42	†14.74 ±0.31
	+ DI	80.81 ±0.72	69.55 ±0.83	56.73 ±0.39	54.16 ±0.05	37.07 ±0.03
	+ SGM	65.65 ±0.95	53.25 ±0.18	38.79 ±0.62	44.33 ±0.63	29.45 ±0.28
	+ GN	71.50 ±0.12	53.45 ±0.65	37.39 ±0.47	34.53 ±0.69	20.29 ±0.36
L2	cSGLD	84.83 ±0.55	74.73 ±0.82	71.45 ±0.56	60.14 ±0.44	39.71 ±0.20
	+ DI	**93.87 ±0.19**	**89.12 ±0.24**	**88.52 ±0.16**	**82.78 ±0.28**	**66.13 ±0.35**
	+ SGM	†83.17 ±0.85	†72.79 ±1.06	†66.19 ±0.89	71.71 ±0.41	52.66 ±0.31
	+ GN	92.99 ±0.13	85.69 ±0.24	82.81 ±0.42	72.88 ±0.30	50.30 ±0.29
	+ MI	†82.44 ±0.19	†70.93 ±1.04	†66.19 ±0.56	†55.51 ±0.59	†34.49 ±0.59
	+ DI	93.48 ±0.23	87.87 ±0.15	86.81 ±0.33	80.37 ±0.20	60.26 ±0.02
	+ SGM	†82.35 ±0.10	†71.54 ±0.58	†64.50 ±0.18	70.47 ±0.22	50.80 ±0.23
	+ GN	90.11 ±0.18	80.35 ±0.61	75.10 ±0.67	64.08 ±0.12	39.85 ±0.52
	1 DNN	47.81 ±1.09	32.29 ±0.64	23.43 ±0.32	22.52 ±0.45	12.77 ±0.32
	+ DI	76.55 ±1.01	62.57 ±0.56	50.17 ±0.33	49.31 ±0.18	⋆32.64 ±0.09
	+ SGM	66.36 ±0.50	51.60 ±0.36	39.05 ±0.24	45.60 ±0.72	30.69 ±0.03
	+ GN	67.02 ±0.17	46.74 ±0.63	32.57 ±0.17	31.12 ±0.77	17.68 ±0.05
	+ MI	55.12 ±0.82	38.47 ±0.82	28.19 ±0.14	27.55 ±0.67	16.34 ±0.37
	+ DI	⋆82.47 ±0.41	⋆69.69 ±0.81	57.79 ±0.57	⋆55.99 ±0.37	⋆38.63 ±0.29
	+ SGM	68.39 ±0.53	54.57 ±0.60	41.48 ±0.37	47.97 ±0.41	⋆33.16 ±0.37
	+ GN	71.27 ±0.54	51.46 ±0.84	36.91 ±0.48	34.54 ±0.32	20.51 ±0.30
L∞	cSGLD	78.71 ±1.19	65.11 ±1.45	61.49 ±0.59	51.81 ±1.45	31.11 ±0.99
	+ DI	90.03 ±0.10	82.13 ±0.45	81.19 ±0.34	74.48 ±0.39	53.51 ±0.39
	+ SGM	81.37 ±0.72	69.88 ±1.31	65.20 ±0.75	71.68 ±0.53	52.15 ±0.32
	+ GN	87.33 ±0.73	76.00 ±1.33	71.67 ±0.97	61.45 ±0.25	37.19 ±0.68
	+ MI	82.89 ±0.70	70.42 ±1.26	66.39 ±0.74	56.68 ±0.97	36.00 ±1.15
	+ DI	**93.97 ±0.26**	**87.69 ±0.44**	**86.78 ±0.16**	**81.08 ±0.14**	**60.87 ±0.48**
	+ SGM	84.19 ±0.21	73.14 ±0.99	67.35 ±0.26	74.36 ±0.47	55.30 ±0.16
	+ GN	89.53 ±0.05	78.69 ±0.19	73.33 ±0.58	63.56 ±0.35	39.79 ±0.52

The surrogate
weight space
matters for transferability.

Weight space exploration

1. Multimodal exploration

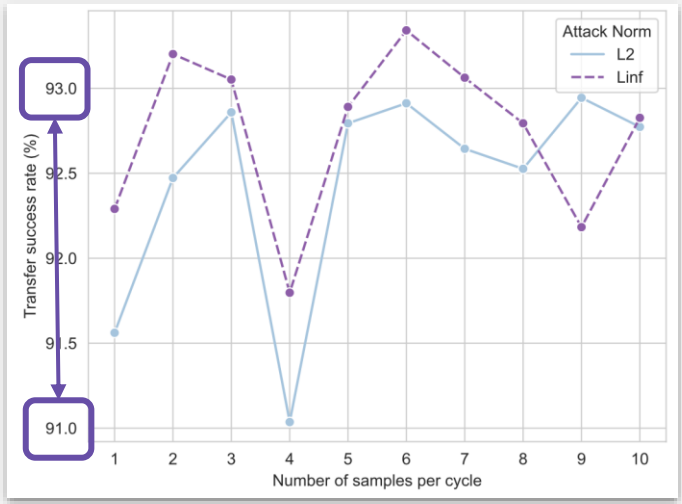
cSGLD



Deep Ensemble

2. Local Exploration

cSGLD



Deep Ensemble

Single weight per vicinity

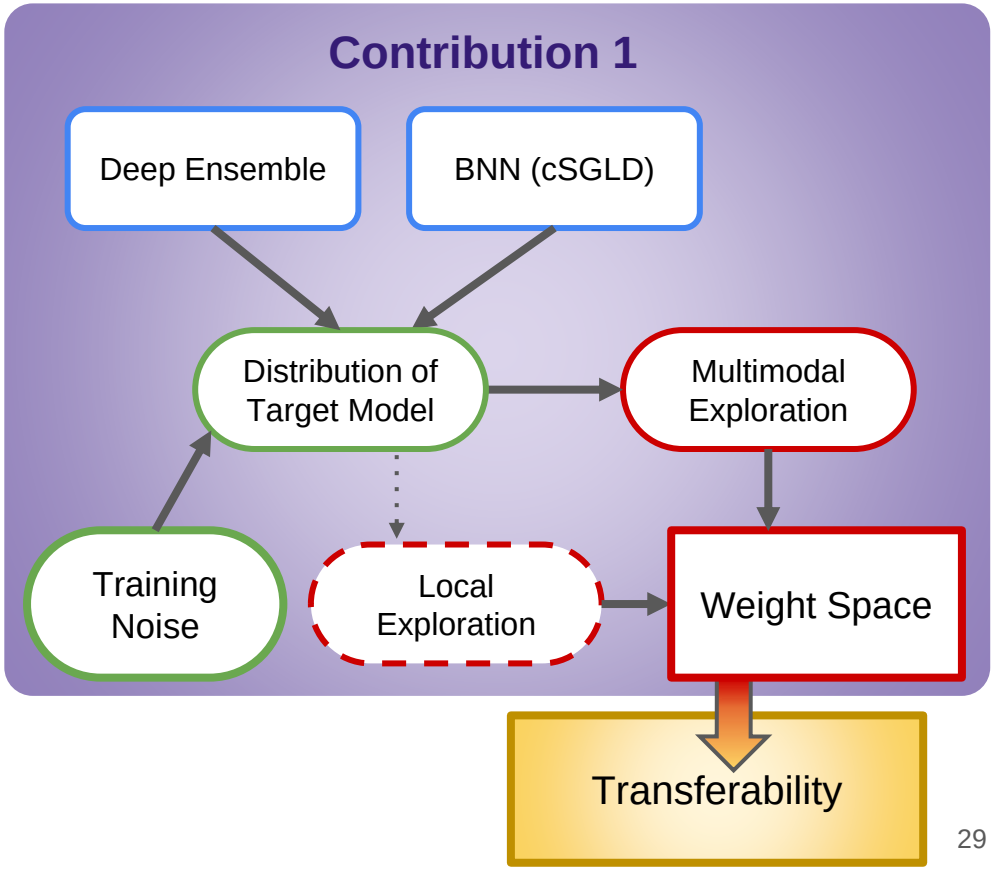
Conclusion

Summary

The unknown weights of the target admit a probability distribution, where the randomness comes from the training noise.

Sampling (approximately) from this distribution in multiple vicinities improves transferability.

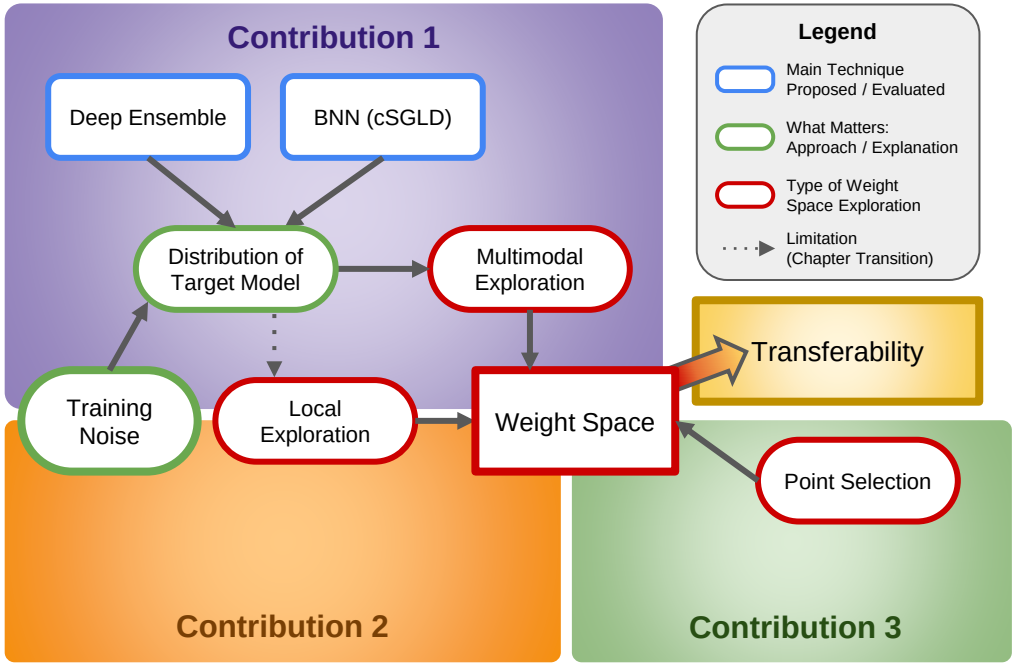
However, both techniques poorly explore locally the weight space.



Conclusion

Open challenge

Can we increase transferability from the local exploration around a surrogate model?



LGV

Transferability from
Large Geometric Vicinity

LGV: Boosting Adversarial Example Transferability from Large Geometric Vicinity

Martin Gubri[1], Maxime Cordy[1], Mike Papadakis[1], Yves Le Traon[1], and Koushik Sen[2]

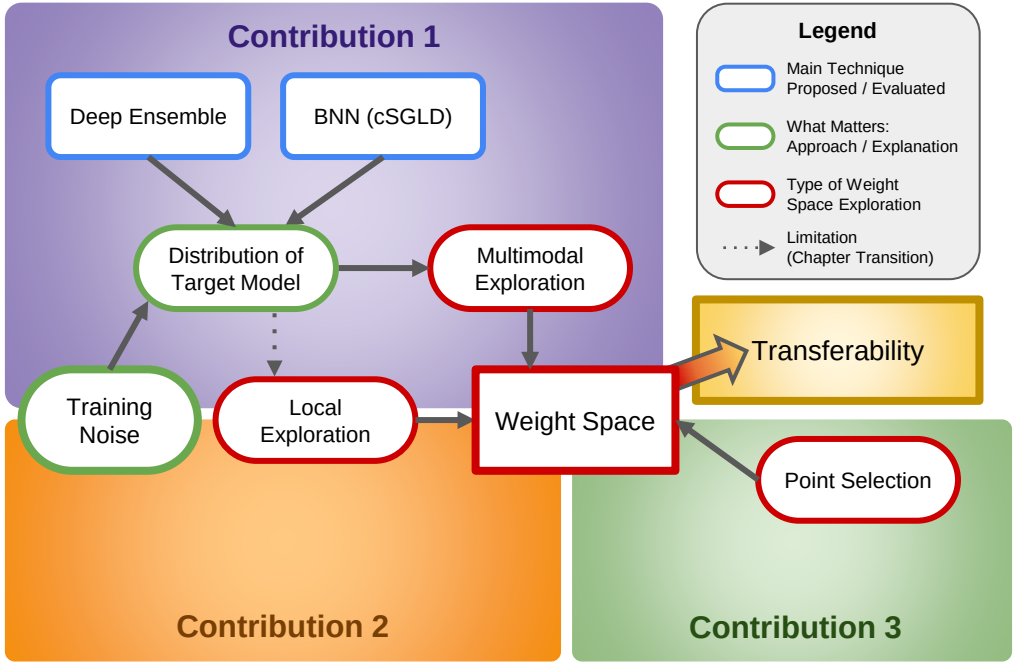
[1] Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, LU firstname.lastname@uni.lu
[2] University of California, Berkeley, CA, USA

Abstract. We propose transferability from Large Geometric Vicinity (LGV), a new technique to increase the transferability of black-box adversarial attacks. LGV starts from a pretrained surrogate model and collects multiple weight sets from a few additional training epochs with a constant and high learning rate. LGV exploits two geometric properties that we relate to transferability. First, models that belong to a wider weight optimum are better surrogates. Second, we identify a subspace able to generate an effective surrogate ensemble among this wider optimum. Through extensive experiments, we show that LGV alone outperforms all (combinations of) four established test-time transformations by 1.8 to 59.9 percentage points. Our findings shed new light on the importance of the geometry of the weight space to explain the transferability of adversarial examples.

Keywords: Adversarial Examples, Transferability, Loss Geometry, Machine Learning Security, Deep Learning

Accepted at ECCV 22

Scope and goal



Scope and goal

Focus

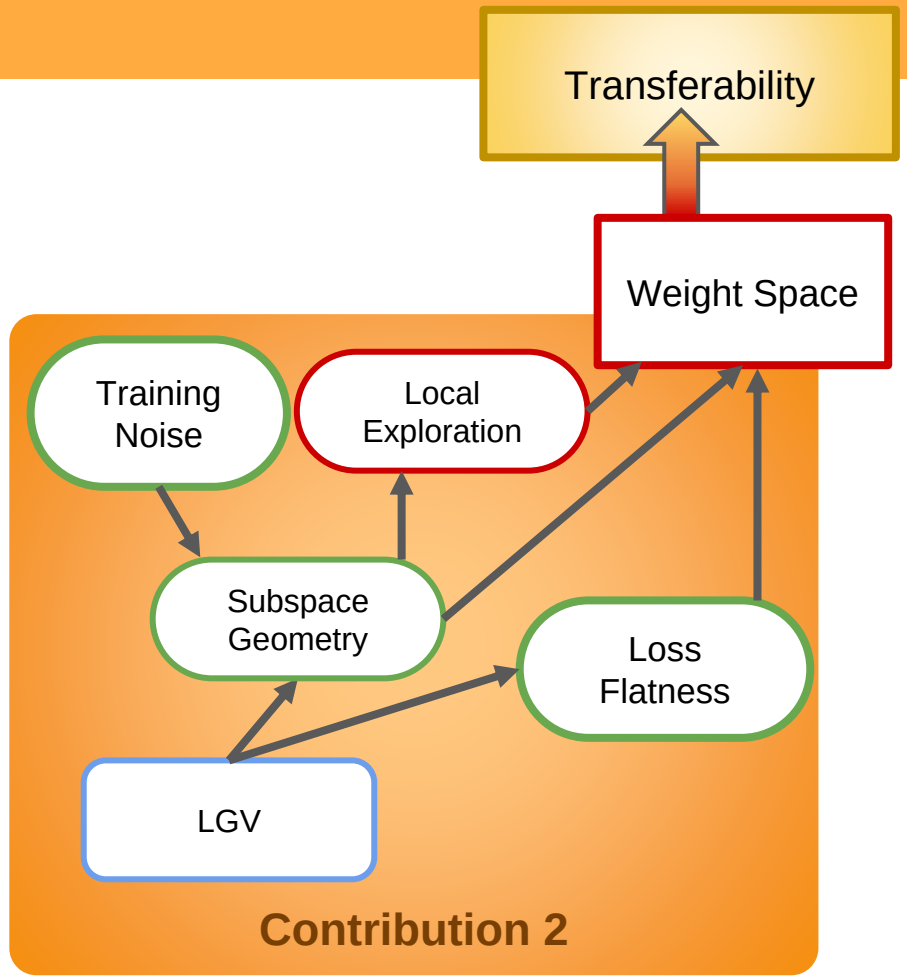
Augment a surrogate model with the local exploration of the weight space

Question

Are representations from a single vicinity too similar to help transferability?

What matters?

Training noise & loss flatness



Motivation

Random directions in the **weight space** increase transferability.

$$\nabla_x \mathcal{L}(x'_k;\, y, w_0 + e_k) \text{ with } e_k \sim \mathcal{N}(\mathbf{0},\, \sigma^2 I_p)$$

DNN weights

Gaussian noise on weights

	Target							
Surrogate	RN50	RN152	RNX50	WRN50	DN201	VGG19	IncV1	IncV3
1 DNN (baseline)	$45.3_{\pm 2.4}$	$29.6_{\pm 0.9}$	$28.8_{\pm 0.2}$	$31.5_{\pm 1.6}$	$17.5_{\pm 0.6}$	$16.6_{\pm 0.9}$	$10.4_{\pm 0.5}$	$5.3_{\pm 1.0}$
+ **RD Weights**	$\mathbf{60.6}_{\pm 1.5}$	$\mathbf{40.5}_{\pm 3.0}$	$\mathbf{39.9}_{\pm 0.2}$	$\mathbf{44.4}_{\pm 3.2}$	$\mathbf{22.9}_{\pm 0.8}$	$\mathbf{22.7}_{\pm 0.5}$	$\mathbf{13.9}_{\pm 0.2}$	$\mathbf{6.6}_{\pm 0.7}$
+ RD Inputs	$46.4_{\pm 1.8}$	$29.0_{\pm 2.2}$	$28.7_{\pm 1.2}$	$32.7_{\pm 1.5}$	$17.5_{\pm 0.6}$	$17.5_{\pm 0.6}$	$10.3_{\pm 0.7}$	$5.6_{\pm 0.7}$

Random directions in the input space do **not**.

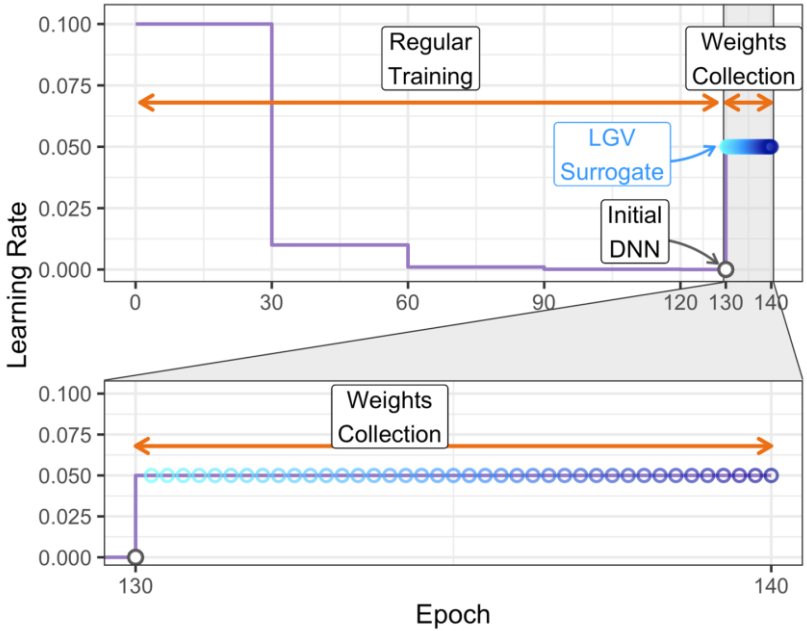
$$\nabla_x \mathcal{L}(x'_k;\, y, w_0) + e'_k \text{ with } e'_k \sim \mathcal{N}(\mathbf{0},\, \sigma'^2 I_d)$$

The vicinity
in the weight space
matters for transferability.

LGV

Phase 1

Collect models during a few additional epochs with a **high learning rate**



Phase 2

Attack one random collected model per iteration

Algorithm 2 I-FGSM Attack on LGV

Input: (x, y) natural example, $(w_1, \ldots, w_K)$ LGV weights, n_{iter} number of iterations, ε p-norm perturbation, α step-size

Output: x_{adv} adversarial example

Shuffle $(w_1, \ldots, w_K)$ ▷ Shuffle weights

$x_{\text{adv}} \leftarrow x$

for $i \leftarrow 1$ **to** n_{iter} **do**

  $x_{\text{adv}} \leftarrow x_{\text{adv}} + \alpha \nabla_x \mathcal{L}(x_{\text{adv}}; y, w_{i \bmod K})$ ▷ Compute the input gradient of the loss of a randomly picked LGV model

  $x_{\text{adv}} \leftarrow \text{project}(x_{\text{adv}}, B_\varepsilon[x])$ ▷ Project in the p-norm ball centred on x of ε radius

  $x_{\text{adv}} \leftarrow \text{clip}(x_{\text{adv}}, 0, 1)$ ▷ Clip to pixel range values

end for

Evaluation

LGV alone beats all (combinations of) four state-of-the-art techniques.

	Target							
Surrogate	RN50	RN152	RNX50	WRN50	DN201	VGG19	IncV1	IncV3
Baselines (1 DNN)								
1 DNN	45.3±2.4	29.6±0.9	28.8±0.2	31.5±1.6	17.5±0.6	16.6±0.9	10.4±0.5	5.3±1.0
MI	53.0±2.2	36.3±1.5	34.7±0.4	38.1±2.0	22.0±0.1	21.1±0.3	13.9±0.4	7.3±0.8
GN	63.9±2.4	43.8±2.4	43.3±1.3	47.4±0.9	24.8±0.3	24.1±1.0	14.6±0.3	6.8±1.2
GN+MI	68.4±2.3	49.3±2.5	47.9±1.2	52.1±1.7	28.4±0.8	28.0±0.7	17.5±0.5	8.7±0.5
DI	75.0±0.2	56.4±1.9	59.6±1.5	61.6±2.4	41.6±1.1	39.7±0.9	27.7±1.0	15.2±1.0
DI+MI	81.2±0.3	63.8±1.9	67.6±0.9	68.9±1.5	49.3±0.7	46.7±0.4	33.0±1.0	19.4±0.9
SGM	64.4±0.8	49.1±3.1	48.9±0.6	51.7±2.8	30.7±0.9	33.6±1.3	22.5±1.5	10.7±0.9
SGM+MI	66.0±0.6	51.3±3.5	50.9±0.9	54.3±2.3	32.5±1.3	35.8±0.7	24.1±1.0	12.1±1.2
SGM+DI	76.8±0.5	62.3±2.7	63.6±1.7	65.3±1.4	45.5±0.9	49.9±0.8	36.0±0.7	19.2±1.7
SGM+DI+MI	80.9±0.7	66.9±2.5	68.7±1.2	70.0±1.7	50.9±0.6	56.0±1.4	42.1±1.4	23.6±1.6
Our techniques								
RD	60.6±1.5	40.5±3.0	39.9±0.2	44.4±3.2	22.9±0.8	22.7±0.5	13.9±0.2	6.6±0.7
LGV (ours)	95.4±0.1	85.5±2.3	83.7±1.2	82.1±2.4	69.3±1.0	67.8±1.2	58.1±0.8	25.3±1.9

How to explain the success of LGV?

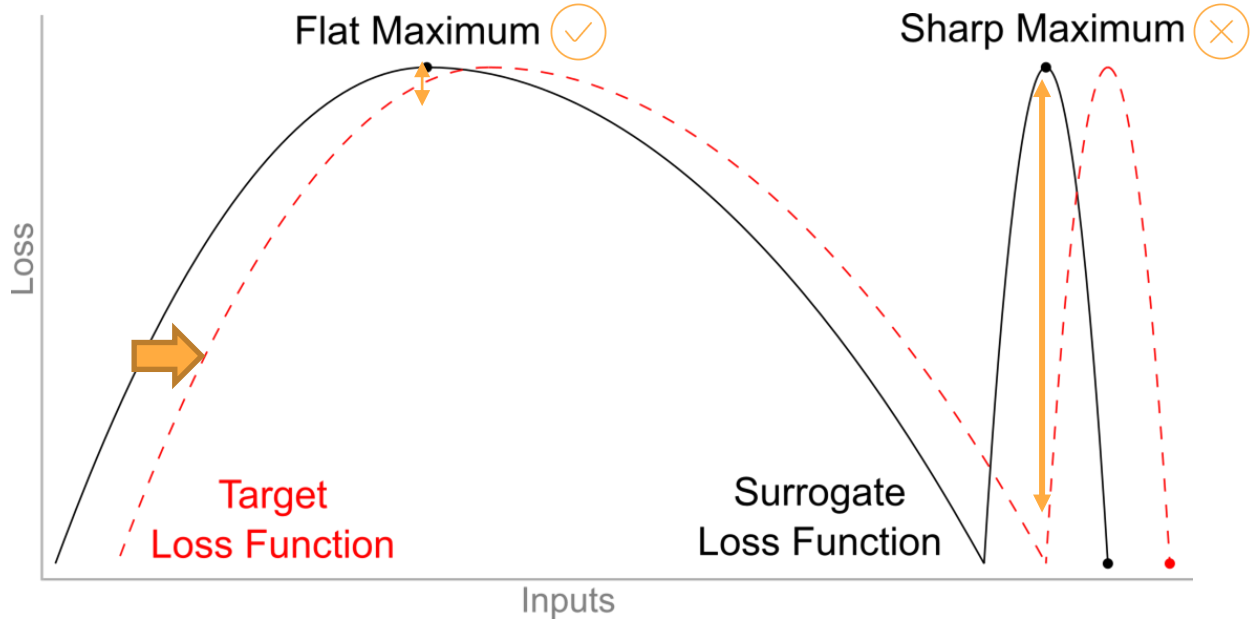
RQ Why do weights from a vicinity help to attack a model from another vicinity?

Two keys:

1. LGV produces flatter adversarial examples.
2. The LGV subspace embeds geometric properties relevant for transferability.

I - The surrogate-target misalignment hypothesis

Flatter adversarial examples may be more robust to misalignment between surrogate and target.



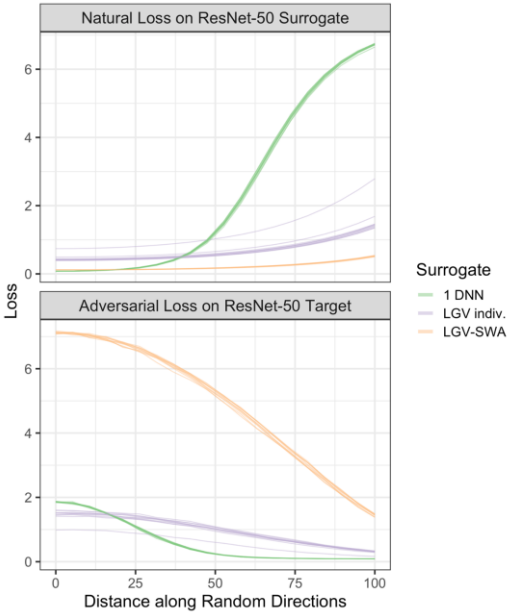
LGV → Flatness in the **weight** space

LGV collects models in flatter regions of the weight space…

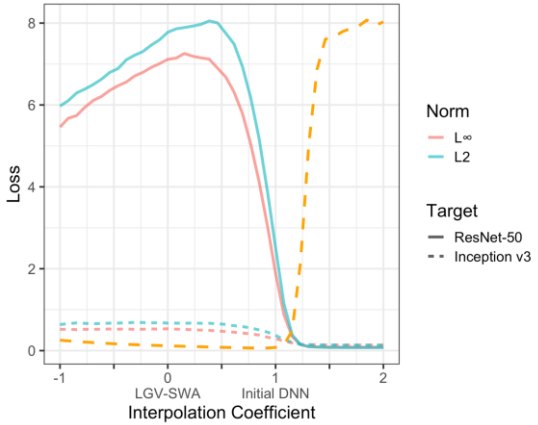
Hessian-based sharpness metrics

Model	Hessian	
	Max EV	Trace
1 DNN	558 ±57	16258 ±725
LGV indiv.	168 ±127	4295 ±517

Random rays



Interpolation in the weight space

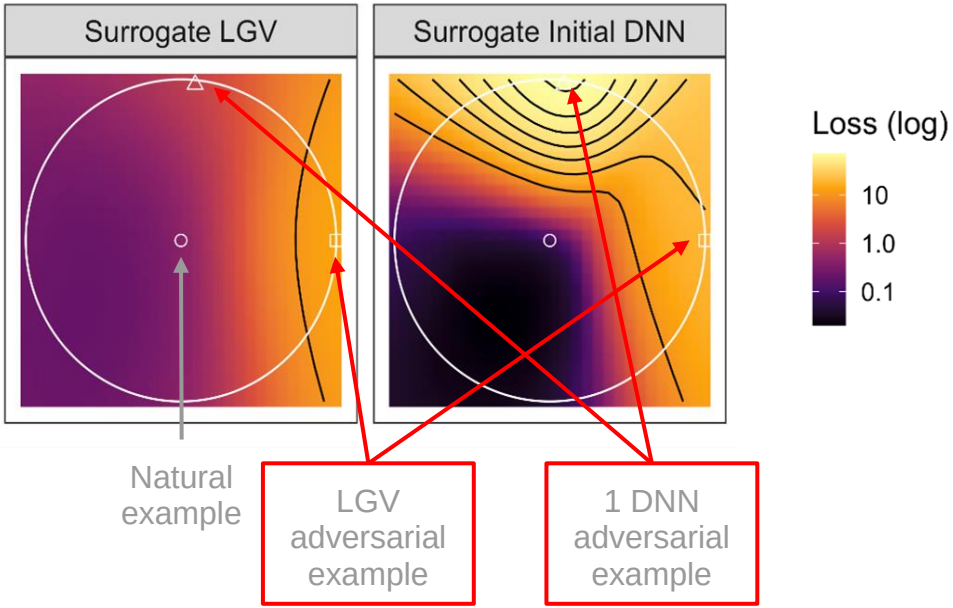


Among others

Optimal lengths of random and LGV deviations vectors

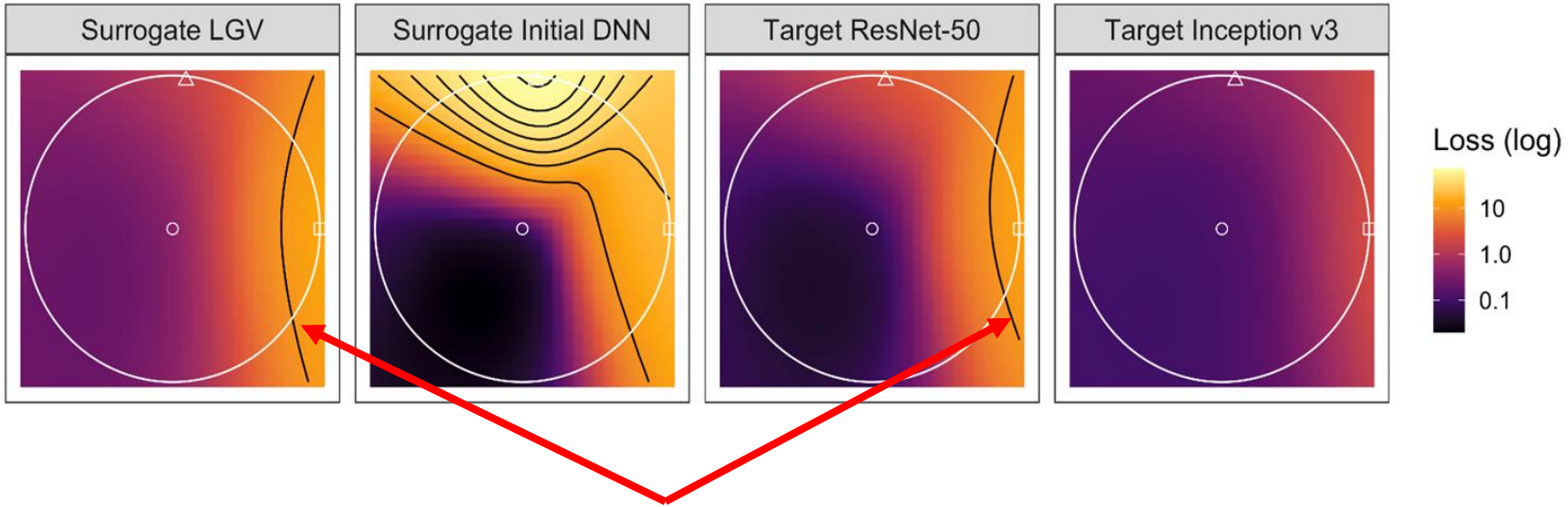
LGV → Flatness in the **input** space

…as a result, LGV produces adversarial examples flatter in the input space.



The surrogate-target (mis)alignment

LGV appears particularly well aligned with the target.



Loss contours have similar shape which appear shifted

II - LGV subspace properties

We consider at the weight subspace defined by deviations of LGV weights from their average:

$$\mathcal{S} = \{w \,|\, w = w_{\mathrm{SWA}} + \mathbf{P}z\}\,,$$

where w_{SWA} is called the shift vector, $\mathbf{P} = (w_1 - w_{\mathrm{SWA}}, \ldots, w_K - w_{\mathrm{SWA}})^\intercal$ is the projection matrix of LGV weights deviations from their mean, and $z \in \mathbb{R}^K$.

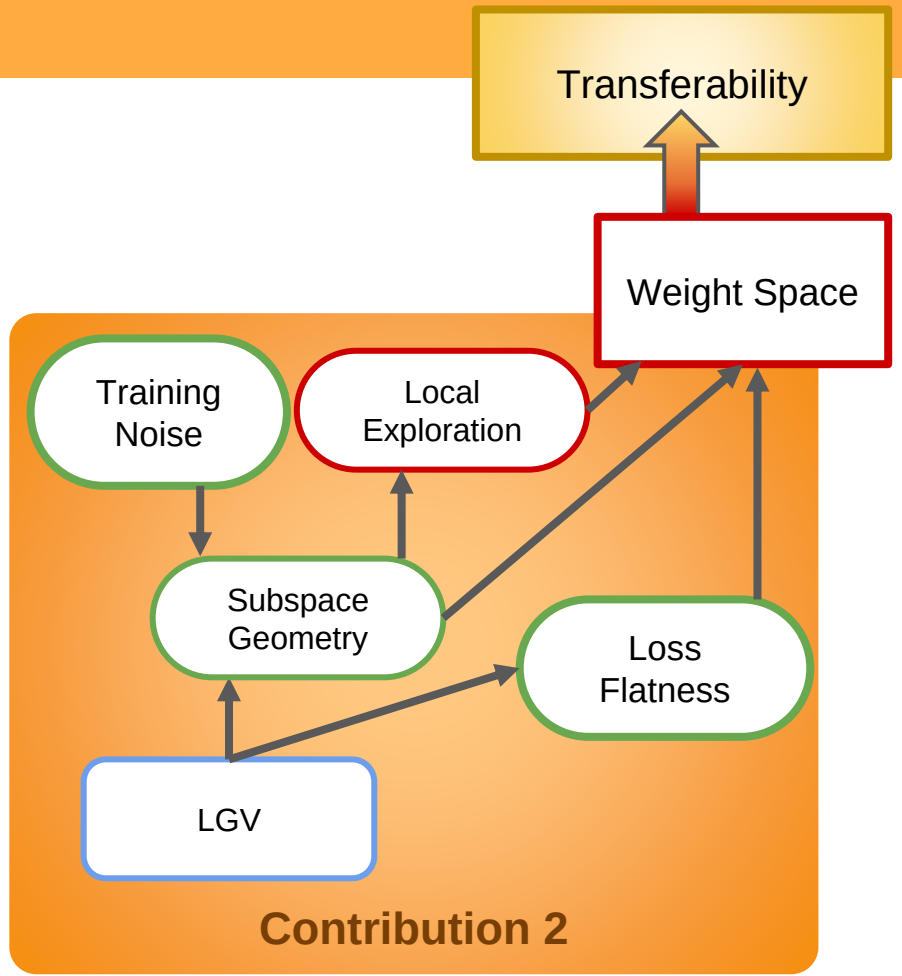
The subspace $\mathcal{S}$:

1. is densely composed good surrogates, i.e., it strongly relates to transferability,
2. is composed of directions whose relative importance correlates with geometrical properties, i.e., its geometry is relevant for transferability,
3. can be shifted to other solutions, i.e., its geometry captures generic properties.

Conclusion

LGV is simple yet effective to enhance transferability from the local exploration of the weight space.

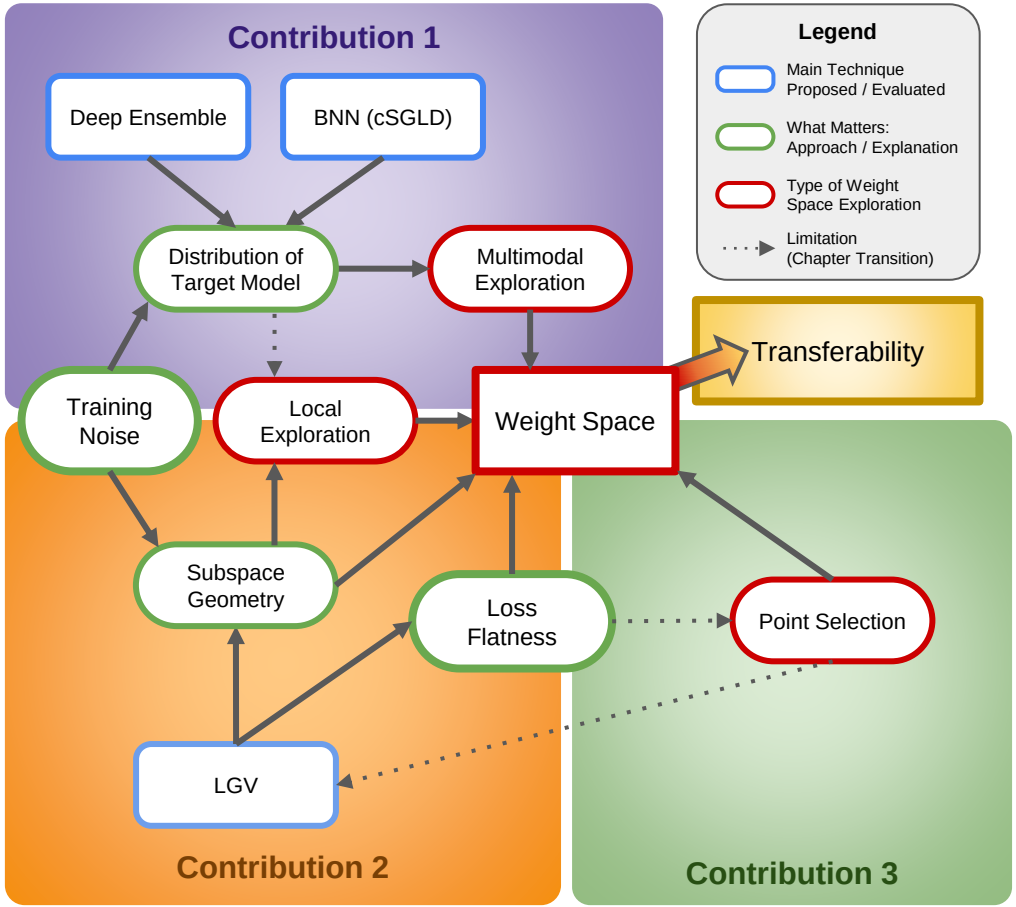
Overall, the improved transferability of LGV comes from the **geometry** of the subspace formed by LGV weights in a **flatter** region of the loss.



Conclusion

Open challenges

- How to train a better base surrogate model?
- Does *explicitly* maximizing flatness improve transferability?



RFN

Transferability of
Representations from Flat Neighbourhood

Going Further: Flatness at the Rescue of Early Stopping for Adversarial Example Transferability

Martin GUBRI, Maxime CORDY & Yves LE TRAON
University of Luxembourg
firstname.name@uni.lu

Abstract

Transferability is the property of adversarial examples to be misclassified by other models than the surrogate model for which they were crafted. Previous research has shown that transferability is substantially increased when the training of the surrogate model has been early stopped. A common hypothesis to explain this is that the later training epochs are when models learn the non-robust features that adversarial attacks exploit. Hence, an early stopped model is more robust (hence, a better surrogate) than fully trained models.

We demonstrate that the reasons why early stopping improves transferability lie in the side effects it has on the learning dynamics of the model. We first show that early stopping benefits transferability even on models learning from data with non-robust features. We then establish links between transferability and the exploration of the loss landscape in the parameter space, on which early stopping has an inherent effect. More precisely, we observe that transferability peaks when the learning rate decays, which is also the time at which the sharpness of the loss significantly drops.

This leads us to propose RFN, a new approach for transferability that minimizes loss sharpness during training in order to maximize transferability. We show that by searching for large flat neighborhoods, RFN always improves over early stopping (by up to 47 points of transferability rate) and is competitive to (if not better than) strong state-of-the-art baselines.

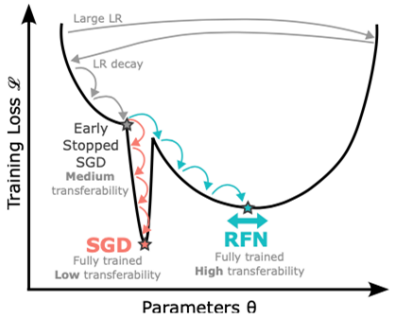
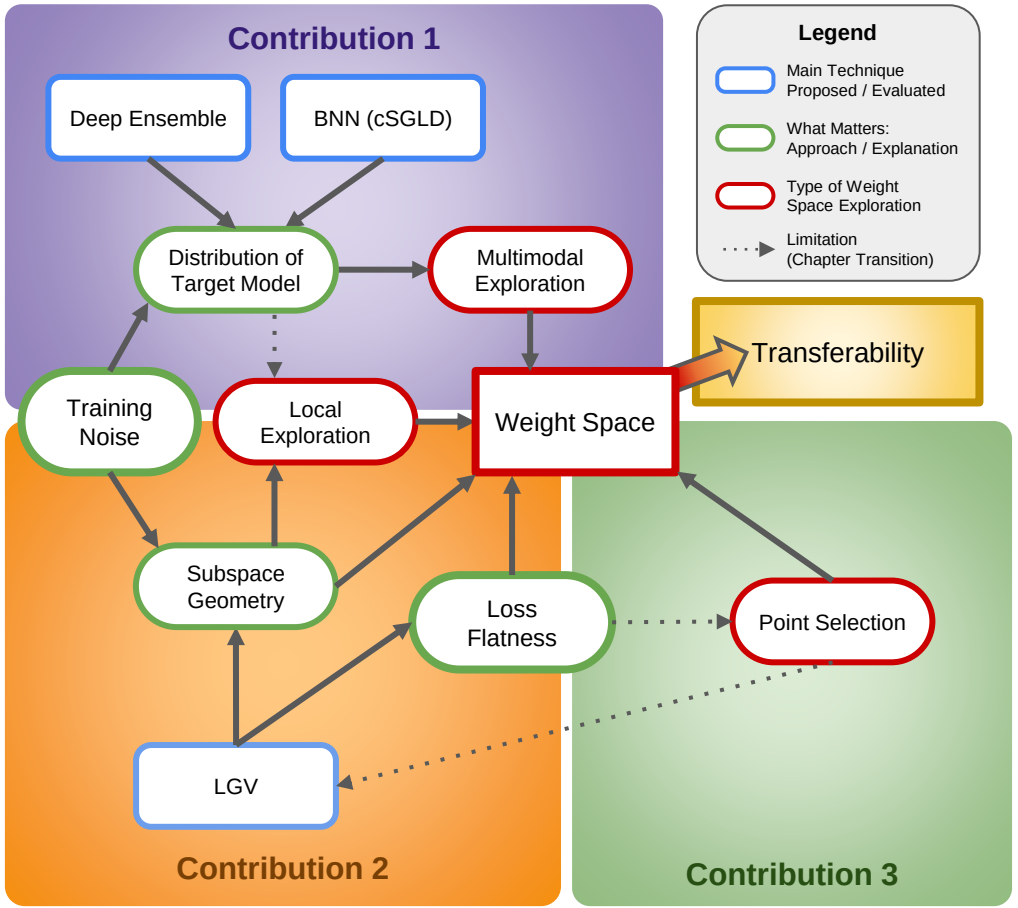


Figure 1. Illustration of the relation between the training dynamics of the surrogate model, sharpness, and transferability. Before the learning rate decays, training is in a "crossing the valley" phase for both SGD and RFN (gray) with plateauing transferability. A few iterations after the decay of the learning rate, early stopped SGD achieves its best transferability. In the following epochs, SGD falls progressively into deep, sharp holes in the parameter space with poor transferability (red). RFN (blue) avoids these holes by minimizing the maximum loss around an unusually large neighbourhood (thick blue arrow).

This observation leads to the discovery of the *transferability* of adversarial examples, i.e., an adversarial exam-

Under Review

Scope and goal



Scope and goal

Focus

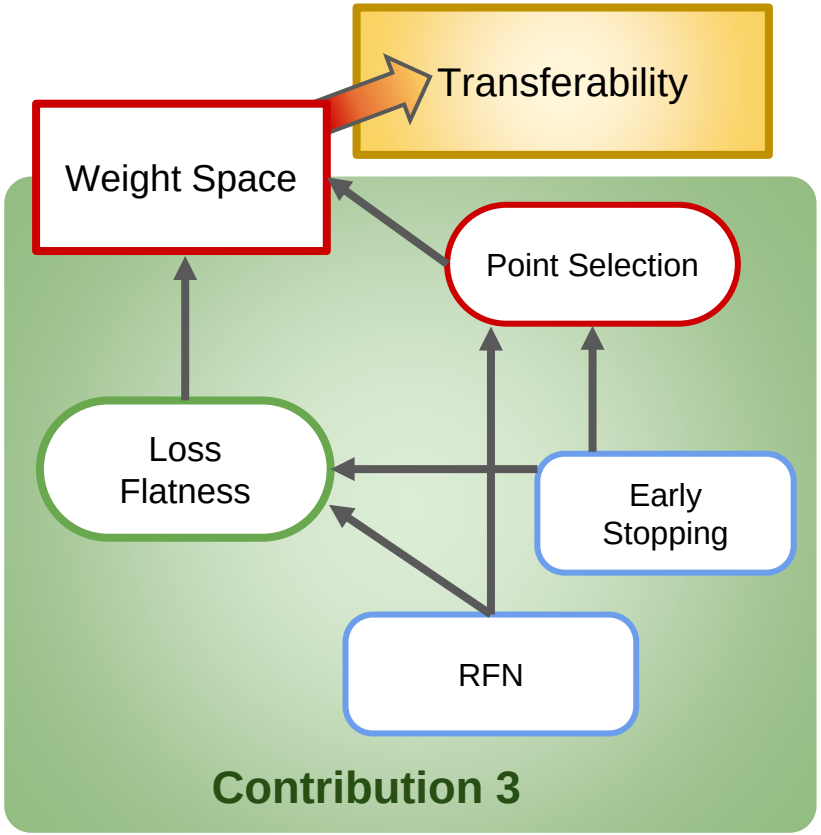
Transferable representation, i.e., single surrogate model

Question

Are vicinities of the surrogate weight space better than others?

What matters?

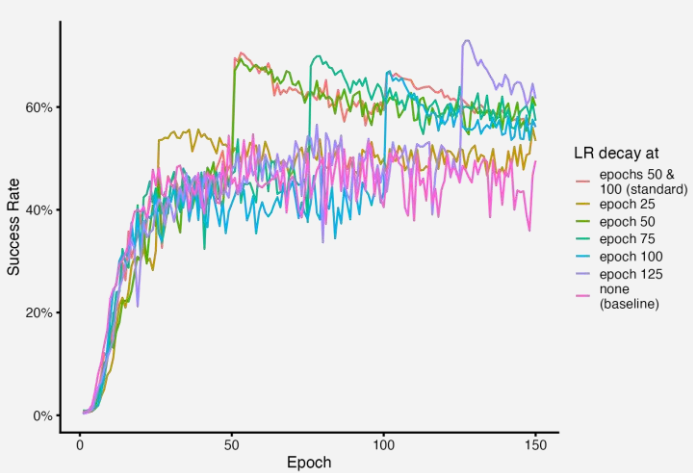
Loss flatness



Transferability and training dynamics

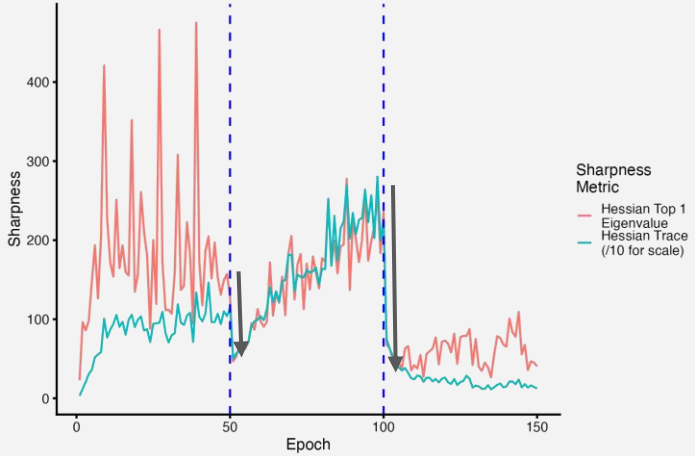
When the learning rate decays…

1. Transferability peaks



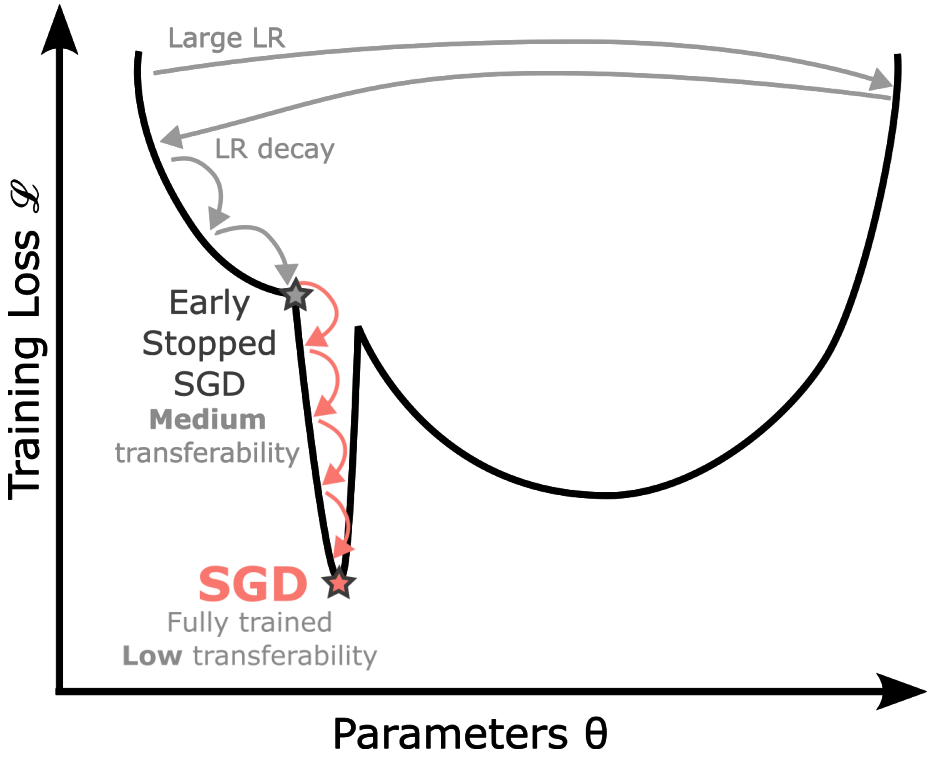
The peak of transferability when the learning rate decays is consistent across epochs.

2. Sharpness drops



Worst and average case sharpness for all training epochs on CIFAR-10.

Transferability and training dynamics



Relation between the training dynamics of the surrogate model, sharpness, and transferability.

1. Before the learning rate decays, training is in a "crossing the valley" phase. Transferability plateaus.
2. After the learning rate decays, training goes down the valley. Soon after, SGD achieves its best transferability (early stopping ☆). Sharpness is reduced.
3. When learning continues, SGD falls progressively into deep, sharp holes in the weight space, where the representations are too specific, thus have poor transferability (fully trained SGD ★).

Explicitly minimizing sharpness

RQ Does explicitly minimizing the sharpness of the surrogate model improve transferability?

Explicitly minimizing sharpness

Sharpness-Aware Minimization (SAM) is an SGD variant that minimizes both the loss value and the loss sharpness, by solving a min-max optimization problem. SAM seeks neighbourhoods of size ρ with uniformly low loss.

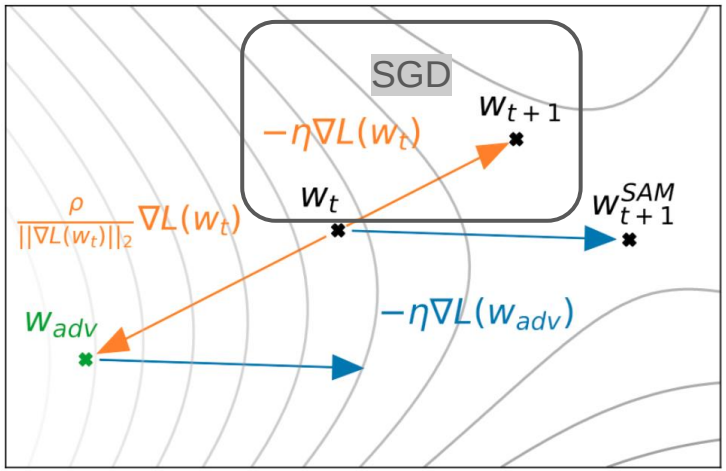


Illustration of the SAM weight update

P. Foret, et al. (2021). Sharpness-Aware Minimization for Efficiently Improving Generalization. *ICLR 2021.*

Explicitly minimizing sharpness

Sharpness-Aware Minimization (SAM) is an SGD variant that minimizes both the loss value and the loss sharpness, by solving a min-max optimization problem. SAM seeks neighbourhoods of size ρ with uniformly low loss.

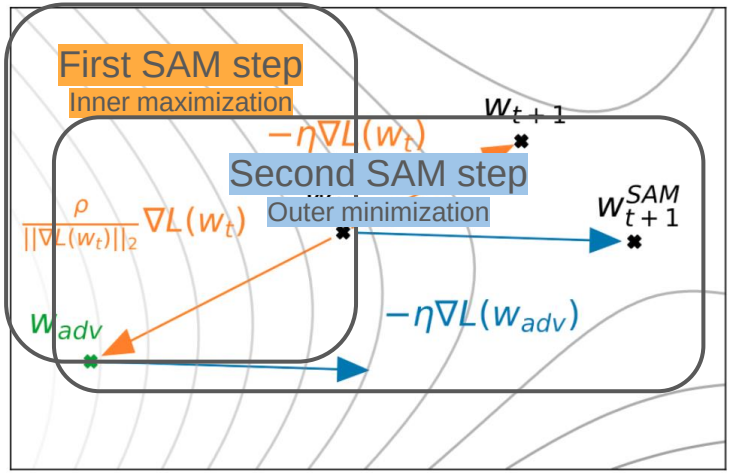
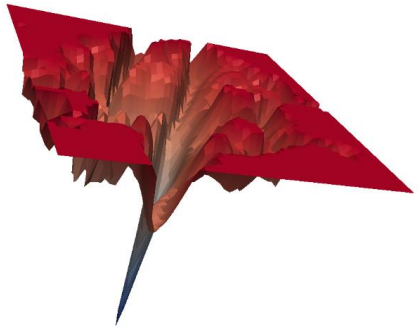


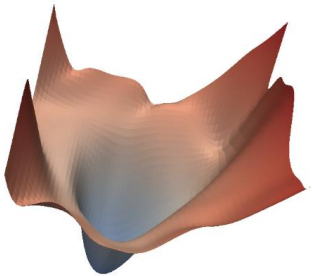
Illustration of the SAM weight update

SGD



A sharp minimum obtained with SGD

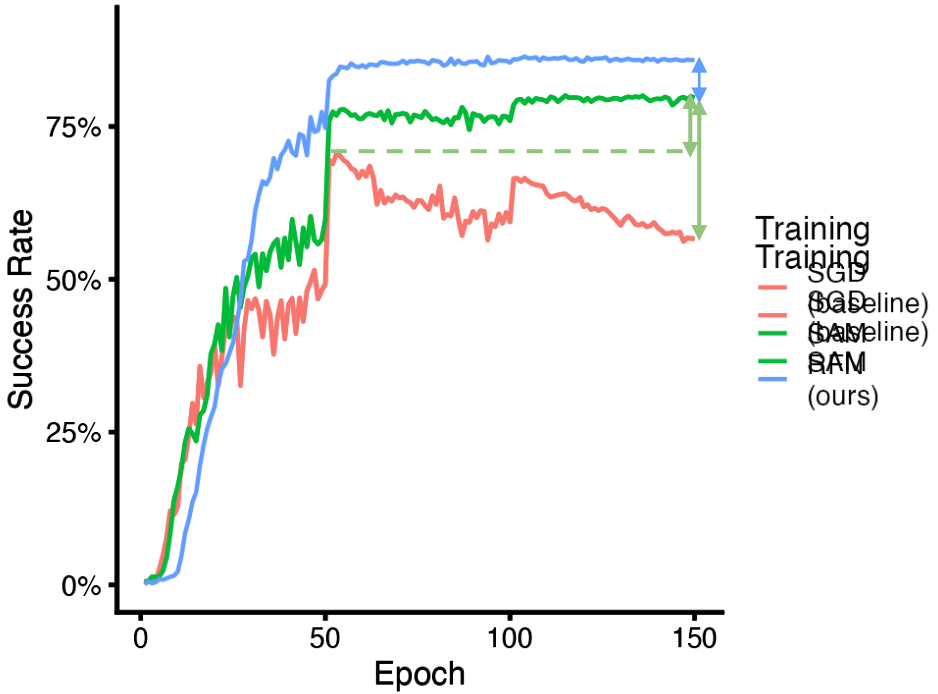
SAM



A flat minimum obtained with SAM

P. Foret, et al. (2021). Sharpness-Aware Minimization for Efficiently Improving Generalization. *ICLR 2021.*

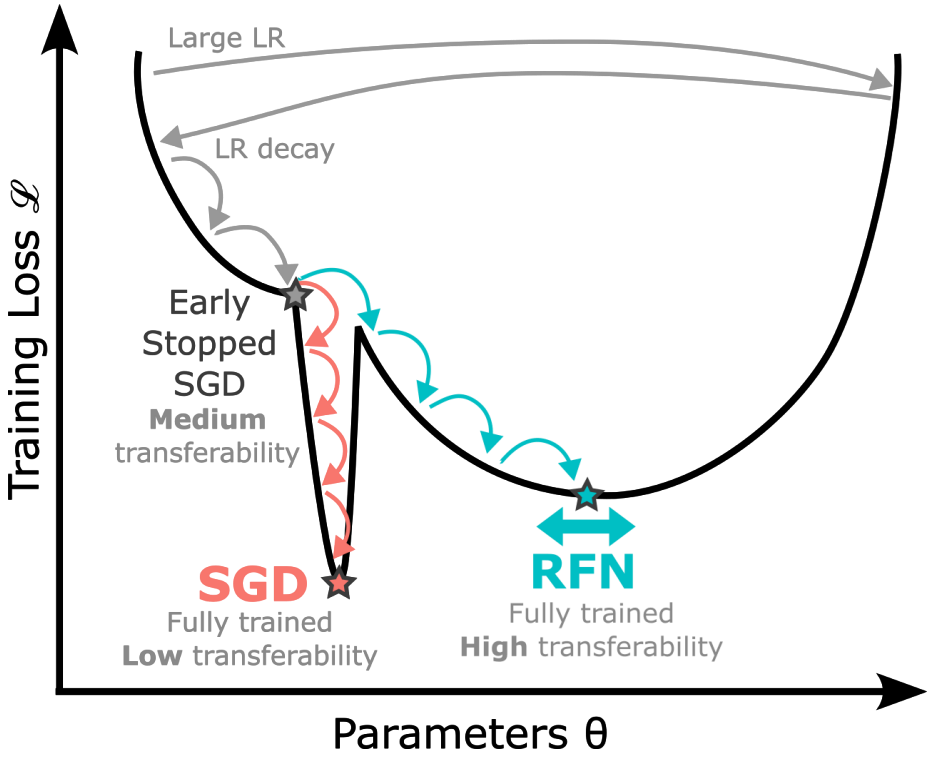
Explicitly minimizing sharpness



Transferability per epoch of SGD, SAM and RFN on CIFAR-10 (average over nine targets)

- Minimizing sharpness improves transferability
- Large flat neighbourhoods are optimal for transferability
  RFN ρ=0.4 > SAM ρ=0.05
- RFN benefits specifically to transferability, not natural accuracy
- Same observations on ImageNet

Explicitly minimizing sharpness



RFN (★) avoids deep sharp holes by minimizing the maximum loss around an unusually large neighbourhood (⟷).

Evaluation

1. RFN is competitive to train a single surrogate model.

RFN is best in 43 out of 57 cases.

Success rate of competitive techniques on CIFAR-10

Surrogate	Target								
	RN18	RN50	RN101	DN161	DN201	VGG13	VGG19	IncV3	WRN28
Fully Trained SGD	57.9	81.2	70.6	70.8	66.1	27.8	26.3	49.4	66.5
Early Stopped SGD	73.3	87.8	82.1	81.4	78.3	45.5	44.3	66.8	79.5
SAT [17]	66.3	76.2	73.6	66.9	66.1	49.8	48.5	57.9	67.8
RFN **(ours)**	**89.7**	**97.3**	**95.5**	**95.7**	**94.0**	**63.6**	**60.6**	**87.3**	**93.0**

Success rate of competitive techniques on ImageNet

Surrogate	Target									
	RN50	RN152	RNX50	WRN50	VGG19	DN201	IncV1	IncV3	ViT B	SwinS
Fully Trained SGD	44.5	25.2	24.8	27.1	16.2	16.4	9.8	8.0	1.8	3.3
Early Stopped SGD	51.5	27.4	27.7	28.0	18.4	18.7	10.8	10.4	2.2	2.7
LGV-SWA [8]	82.5	56.8	58.5	54.0	40.9	42.4	28.3	15.1	3.1	5.7
SAT [17]	76.3	62.5	66.8	63.4	48.1	**59.0**	**47.9**	**40.8**	**17.4**	**16.8**
RFN **(ours)**	**85.7**	**70.3**	**73.3**	**73.2**	**58.2**	55.6	37.9	20.5	4.0	8.2

2. RFN is a better base model for complementary techniques.

Success rate of complementary techniques on ImageNet

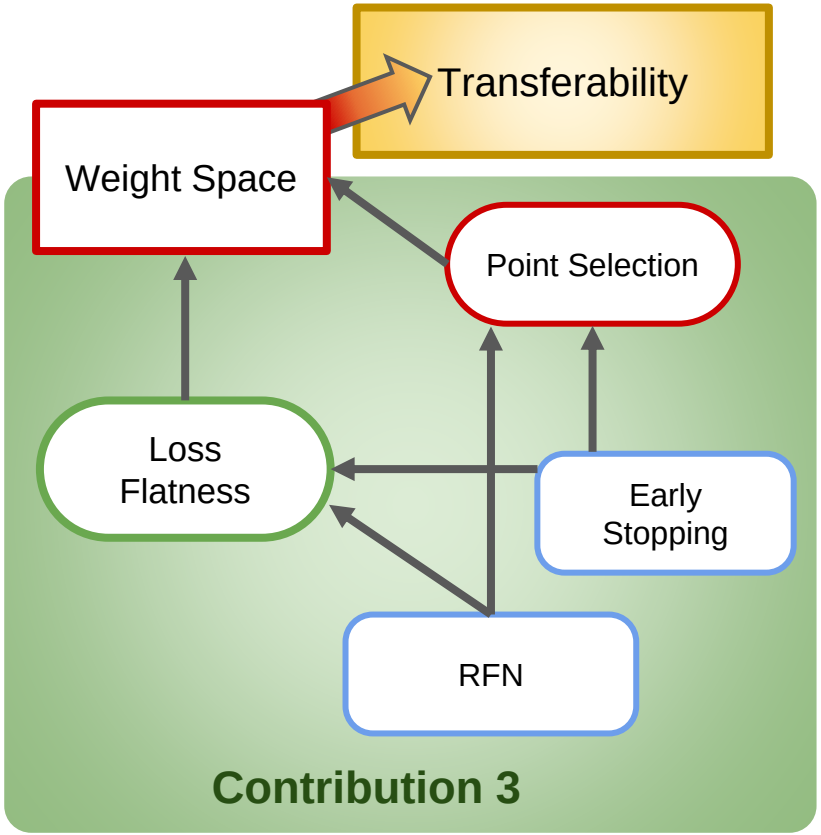
Attack	Target									
	RN50	RN152	RNX50	WRN50	VGG19	DN201	IncV1	IncV3	ViT B	SwinS
Model Augmentation Techniques										
GN [12]	68.0	43.1	41.3	44.1	24.8	27.2	14.3	9.9	1.9	3.8
GN+RFN	89.6	76.6	79.4	79.9	65.7	60.3	42.2	22.4	3.8	7.8
SGM [21]	62.8	40.6	41.5	43.5	31.9	28.0	19.3	13.2	4.1	7.9
SGM+RFN	83.2	68.7	71.5	73.0	67.0	56.2	48.9	26.6	6.2	13.6
LGV [8]	93.3	78.1	75.3	73.1	64.4	61.6	49.3	28.8	5.0	6.5
LGV+RFN	88.7	74.3	75.7	75.7	70.3	61.9	56.8	31.5	4.5	7.3
Data Augmentation Techniques										
DI [22]	83.1	60.5	68.1	67.3	45.4	57.9	41.4	30.7	5.7	9.9
DI+RFN	95.0	89.7	90.7	91.6	85.3	87.8	87.5	64.2	14.2	19.0
SI [13]	60.0	37.9	37.3	40.0	23.9	30.0	19.6	13.5	2.6	3.8
SI+RFN	89.2	76.6	80.1	79.1	65.2	69.8	58.0	35.8	5.0	8.5
VT [20]	58.6	35.0	35.2	38.5	23.9	24.7	14.9	11.0	2.3	4.9
VT+RFN	92.0	81.2	82.4	82.9	72.3	72.3	56.7	33.6	7.0	13.5
Attack Optimizers										
MI [3]	56.8	37.4	37.5	38.9	27.0	29.3	18.4	14.6	3.5	4.8
MI+RFN	89.4	79.3	80.4	80.8	71.5	71.1	60.1	39.3	8.5	15.2
NI [13]	53.7	33.1	32.9	35.1	20.5	20.8	12.2	9.4	1.8	3.9
NI+RFN	83.9	67.3	69.8	71.4	56.1	52.5	35.6	17.6	3.8	7.0

The flatness
of the neighbourhood
matters for transferability.

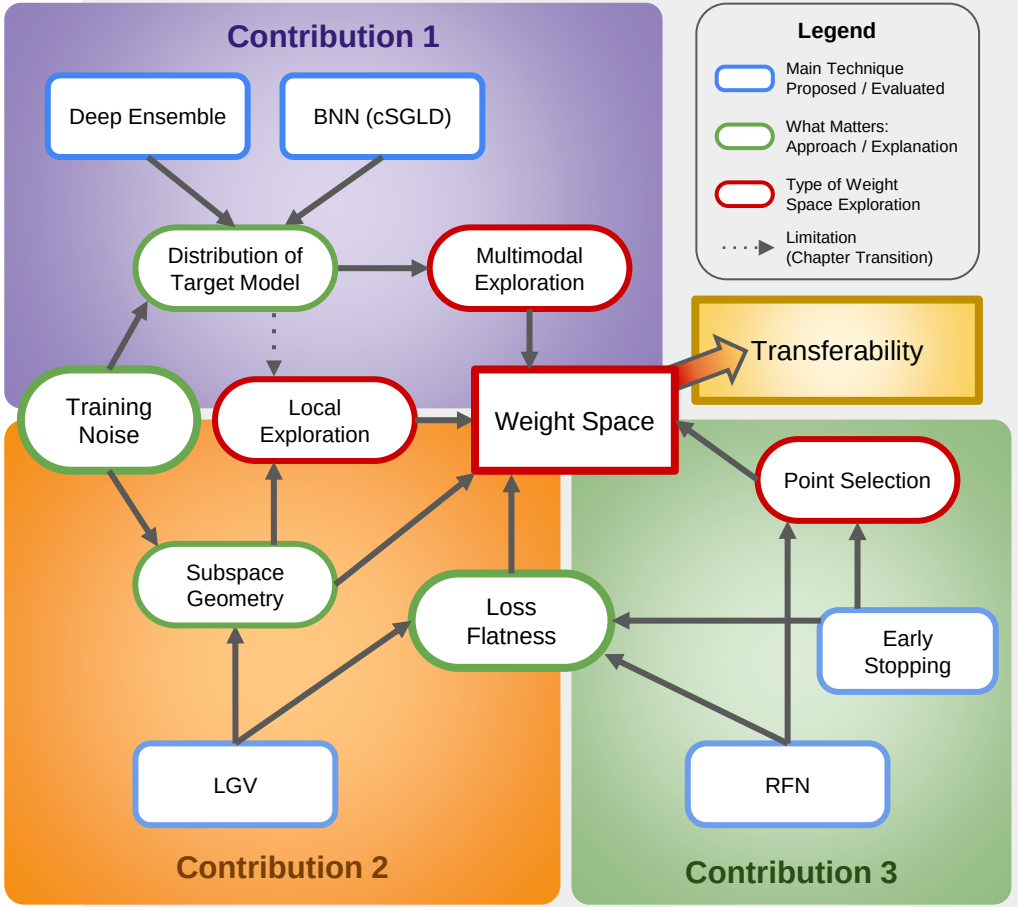
Conclusion

Explicitly maximizing flatness finds transferable representations

Early stopping has an implicit effect on flatness.

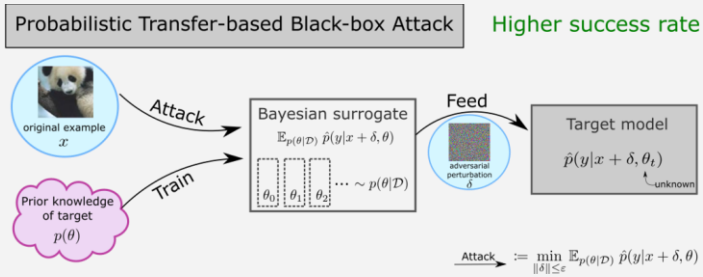


Conclusion

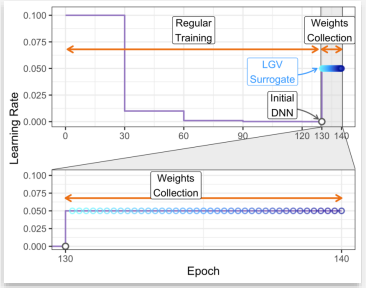

Contribution 1
Deep Ensemble
BNN (cSGLD)
Distribution of Target Model
Multimodal Exploration
Training Noise
Local Exploration
Weight Space
Transferability
Legend
Main Technique Proposed / Evaluated
What Matters: Approach / Explanation
Type of Weight Space Exploration
Limitation (Chapter Transition)
Point Selection
Subspace Geometry
Loss Flatness
Early Stopping
LGV
RFN
Contribution 2
Contribution 3

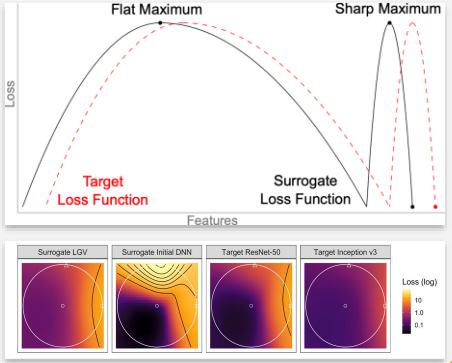
Summary

1. Deep ensemble & Bayesian NN

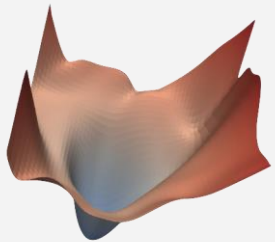


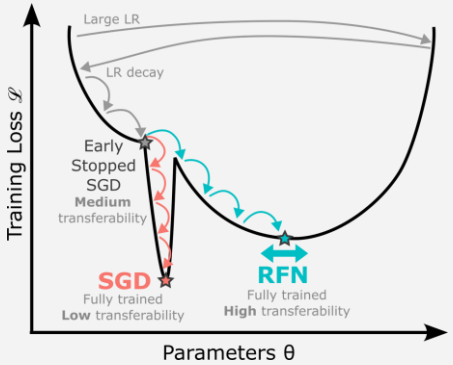
2. LGV





3. RFN



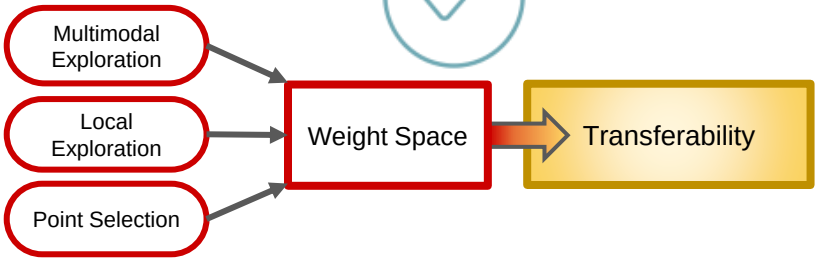


Papers

1. Efficient and Transferable Adversarial Examples from Bayesian Neural Networks. **UAI** 2022
2. LGV: Boosting Adversarial Example Transferability from Large Geometric Vicinity. **ECCV** 2022
3. Going Further: Flatness at the Rescue of Early Stopping for Adversarial Example Transferability. Under Review, 2023

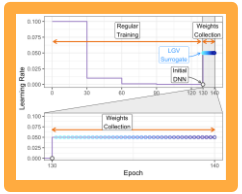
Challenges

1. Knowledge gap



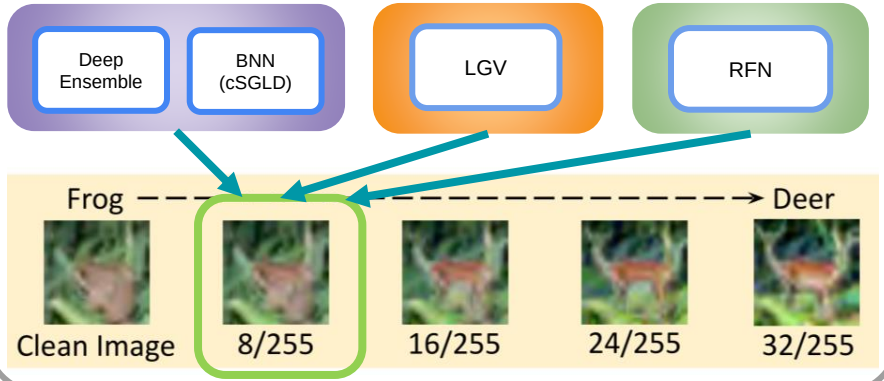
2. High training computational cost

cSGLD
(Zhang, 2020).



3. Lack of insights

4. Low success rate of small perturbations

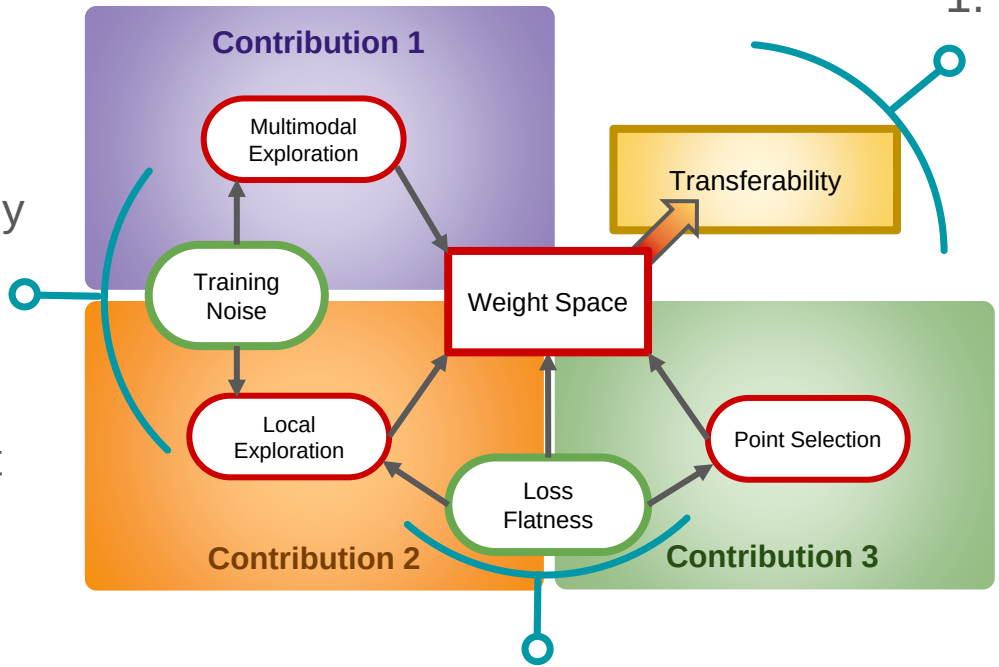


5. Over-specific transferability techniques

Takeaways

1. The surrogate weight space is relevant for transferability.

2. Training noise induces significantly different representations, which hinders transferability if not controlled.



3. The transferability of flat neighbourhoods gives another look at the relation between flatness and generalization.

Limitations and perspectives

Transferability and uncertainty

How different threat models affect the type of uncertainty?

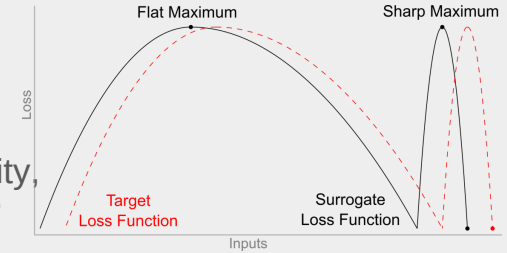
- Unknown training dataset → aleatoric uncertainty
- Distributional shift → epistemic / aleatoric uncertainty

Bayesian perspectives on LGV & RFN

- Mandt et al. (2017): SGD with a constant learning rate approximates the Bayesian posterior.
- Möllenhoff and Khan (2022): SAM is an optimal relaxation of the Bayes objective

Controversy about the loss flatness

Why some explicit or implicit regularizations improve transferability, while others do not?



Extensions of experimental settings

- Physical domain
- Broader applications, including NLP.
  E.g., evading LLM text detectors
- Non-L_p attacks

Other contributions

Additional Papers

1. Adversarial Perturbation Intensity Achieving Chosen Intra-Technique Transferability Level for Logistic Regression. 2018
2. Search-Based Adversarial Testing and Improvement of Constrained Credit Scoring Systems. **FSE** 2020
3. Influence-Driven Data Poisoning in Graph-Based Semi- Supervised Classifiers. **CAIN** 2022

Tools

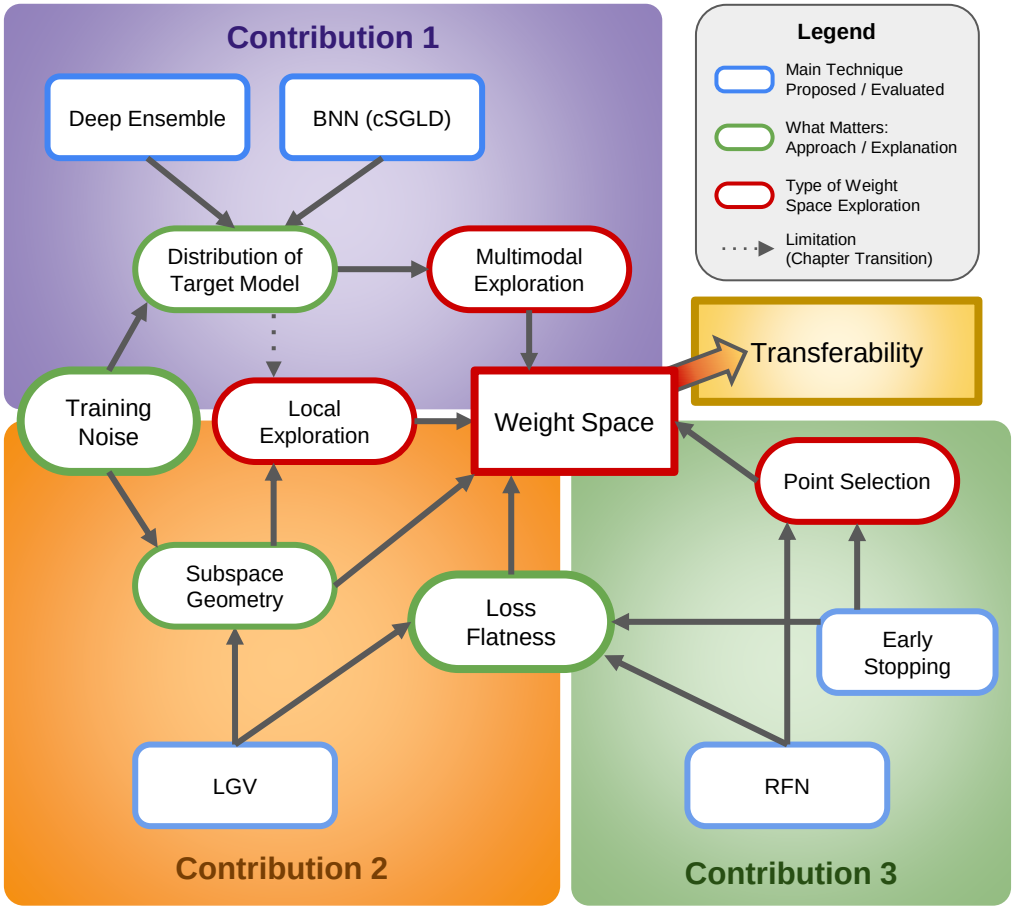
- Torchattacks library: refactor, merge and showcase LGV code, demo notebook, bug fix

```
from torchattacks import LGV, BIM

atk = LGV(base_model, trainloader, lr=0.05, epochs=10, nb_models_epoch=4,
          wd=1e-4, attack_class=BIM, eps=4/255, alpha=4/255/10, steps=50)
```

- Open code source
- ART library: bug fix

Thanks!



Open science

1. BNN
   github.com/Framartin/transferable-bnn-adv-ex
2. LGV
   github.com/Framartin/lgv-geometric-transferability
3. RFN
   github.com/Framartin/rfn-flatness-transferability

Additional materials available on gubri.eu

LGV

Additional slides

LGV: Boosting Adversarial Example Transferability from Large Geometric Vicinity

Martin Gubri[1], Maxime Cordy[1], Mike Papadakis[1], Yves Le Traon[1], and Koushik Sen[2]

[1] Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, Luxembourg, LU firstname.lastname@uni.lu
[2] University of California, Berkeley, CA, USA

Abstract. We propose transferability from Large Geometric Vicinity (LGV), a new technique to increase the transferability of black-box adversarial attacks. LGV starts from a pretrained surrogate model and collects multiple weight sets from a few additional training epochs with a constant and high learning rate. LGV exploits two geometric properties that we relate to transferability. First, models that belong to a wider weight optimum are better surrogates. Second, we identify a subspace able to generate an effective surrogate ensemble among this wider optimum. Through extensive experiments, we show that LGV alone outperforms all (combinations of) four established test-time transformations by 1.8 to 59.9 percentage points. Our findings shed new light on the importance of the geometry of the weight space to explain the transferability of adversarial examples.

Keywords: Adversarial Examples, Transferability, Loss Geometry, Machine Learning Security, Deep Learning

Accepted at ECCV 22

Motivation

Random directions in the **weight space** increase transferability.

$$\nabla_x \mathcal{L}(x'_k;\, y, w_0 + e_k) \text{ with } e_k \sim \mathcal{N}(\mathbf{0},\, \sigma^2 I_p)$$

Equivalent to adding **input** noise structured by local variations of input gradients in the weight space:

$$\mathcal{N}\left(\nabla_x \mathcal{L}(x'_k;\, y, w_0),\ \sigma^2\, \mathbf{J}_{\nabla_x \mathcal{L}(x'_k;\, y, \cdot)}(w_0)\, \mathbf{J}_{\nabla_x \mathcal{L}(x'_k;\, y, \cdot)}(w_0)^T\right)$$

Usage

Implemented in the *torchattacks* library.

```
[ ]  from torchattacks import LGV, BIM

     atk = LGV(base_model, trainloader, lr=0.05, epochs=10, nb_models_epoch=4,
               wd=1e-4, attack_class=BIM, eps=4/255, alpha=4/255/10, steps=50)
     # atk.load_models(list_models)  # uncomment to load our publicly available pretrained models
     atk.collect_models()  # or collect models yourself
     atk.save_models('models/lgv')

     report_success_rate(atk)
```

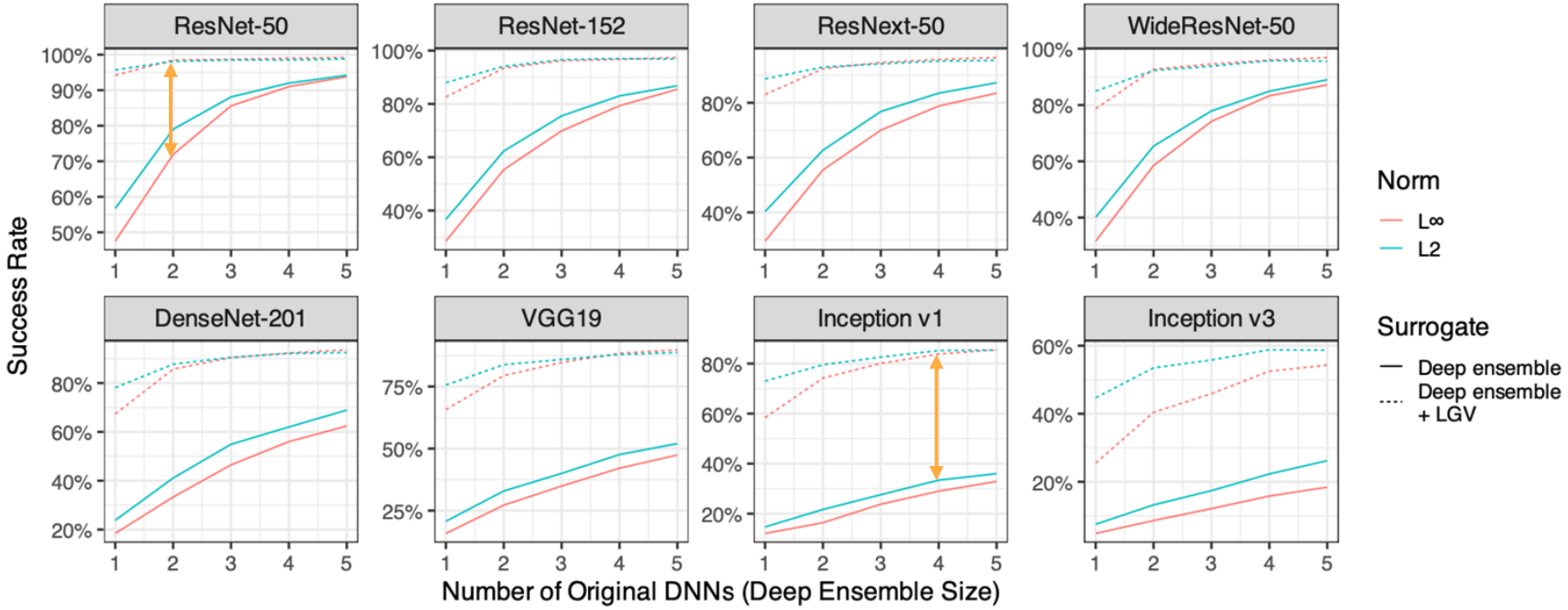
```
Phase 2: craft adversarial examples with BIM
Success rate of LGV-BIM: 97.6%
```

Possible to combine with any other attack.

Demo notebook available.

Complementarity with deep ensemble

LGV can be applied multiple times from each independently trained model



Types of high learning rate

Typology of high learning rates

1. The highest possible learning rate that does not make the model leave the current local minimum;
2. The highest possible learning rate that makes the model jump between different local minima but does not cause deterministic chaos
3. The highest possible learning rate that causes deterministic chaos but does not lead to numerical divergence

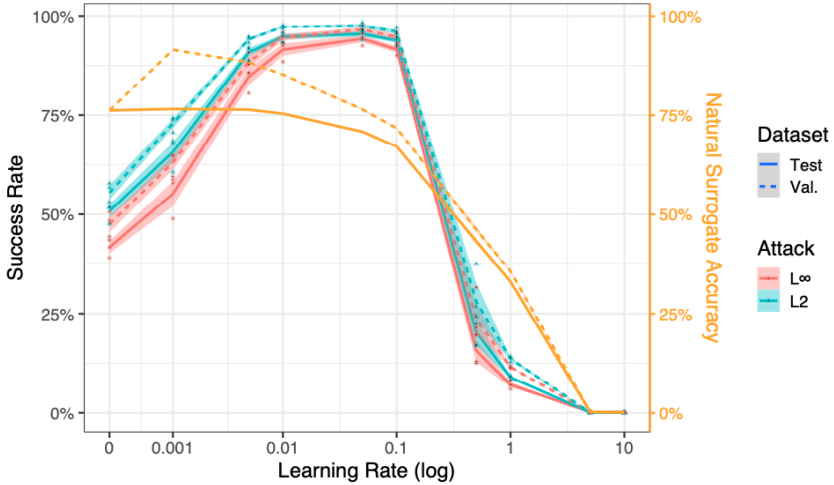


Figure 5.5: Transfer success rate against the ResNet-50 target (*red, blue*) and natural test accuracy (*orange*) of the LGV surrogate trained with a wide range of constant learning rate, in pseudo-log scale. The null learning rate refers to the initial DNN.

I - The surrogate-target misalignment hypothesis

Background about flatness and (natural) generalization

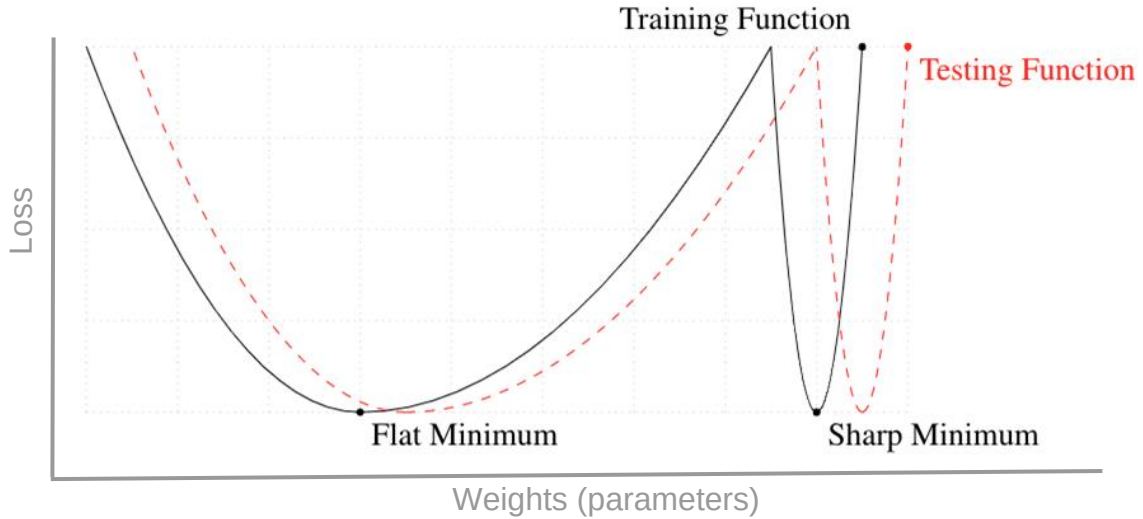


Figure 1: A Conceptual Sketch of Flat and Sharp Minima. The Y-axis indicates value of the loss function and the X-axis the variables (parameters)

Keskar, N. S., et al. (2017). On large-batch training for deep learning: Generalization gap and sharp minima. *ICLR 2017.*

Background - SGD Subspace

Despite the high dimensionality of the weight space, SGD updates are concentrated in a tiny subspace

---

GRADIENT DESCENT HAPPENS IN A TINY SUBSPACE

Guy Gur-Ari[*]
School of Natural Sciences
Institute for Advanced Study
Princeton, NJ 08540, USA
guyg@ias.edu

Daniel A. Roberts[*]
Facebook AI Research
New York, NY 10003, USA
danr@fb.com

Ethan Dyer
Johns Hopkins University
Baltimore, MD 21218, USA
edyer4@jhu.edu

ABSTRACT

We show that in a variety of large-scale deep learning scenarios the gradient dynamically converges to a very small subspace after a short period of training. The subspace is spanned by a few top eigenvectors of the Hessian (equal to the number of classes in the dataset), and is mostly preserved over long periods of training. A simple argument then suggests that gradient descent may happen mostly in this subspace. We give an example of this effect in a solvable model of classification, and we comment on possible implications for optimization and learning.

II - A) A subspace **useful** for transferability

The LGV subspace is significantly better than a random subspace

→ Specific relation to transferability

Table 9: Transfer success rate of random directions sampled in LGV deviations subspace.

Norm	Surrogate	Target							
		RN50	RN152	RNX50	WRN50	DN201	VGG19	IncV1	IncV3
L∞	LGV	$95.5_{\pm 0.1}$	$85.5_{\pm 2.1}$	$83.6_{\pm 1.1}$	$82.2_{\pm 2.4}$	$69.6_{\pm 1.0}$	$67.8_{\pm 0.9}$	$58.4_{\pm 0.6}$	$25.6_{\pm 1.7}$
L∞	LGV-SWA + RD in $\mathcal{S}$	$96.0_{\pm 0.2}$	$85.6_{\pm 2.5}$	$83.6_{\pm 0.6}$	$82.1_{\pm 2.8}$	$68.6_{\pm 1.1}$	$65.7_{\pm 1.5}$	$54.5_{\pm 0.9}$	$23.5_{\pm 0.4}$
L∞	LGV-SWA + RD	$90.4_{\pm 0.3}$	$71.9_{\pm 3.4}$	$70.0_{\pm 1.2}$	$69.2_{\pm 3.4}$	$50.0_{\pm 1.0}$	$47.4_{\pm 1.9}$	$34.9_{\pm 0.4}$	$13.4_{\pm 0.7}$
L2	LGV	$96.3_{\pm 0.1}$	$90.1_{\pm 1.0}$	$88.8_{\pm 0.4}$	$87.5_{\pm 1.6}$	$79.8_{\pm 1.1}$	$78.1_{\pm 1.6}$	$71.9_{\pm 0.6}$	$43.1_{\pm 0.6}$
L2	LGV-SWA + RD in $\mathcal{S}$	$96.6_{\pm 0.3}$	$90.1_{\pm 1.4}$	$88.7_{\pm 0.5}$	$87.3_{\pm 2.0}$	$77.6_{\pm 1.0}$	$75.6_{\pm 1.5}$	$67.4_{\pm 1.9}$	$37.4_{\pm 0.4}$
L2	LGV-SWA + RD	$91.9_{\pm 0.6}$	$78.2_{\pm 2.9}$	$76.2_{\pm 1.3}$	$75.4_{\pm 2.5}$	$58.1_{\pm 0.3}$	$55.8_{\pm 1.6}$	$42.7_{\pm 0.6}$	$20.0_{\pm 0.6}$

Random directions in full weight space vs. LGV

II - A) A subspace **useful** for transferability

Sampling random directions in the subspace have results close to LGV

→ Densely related to transferability

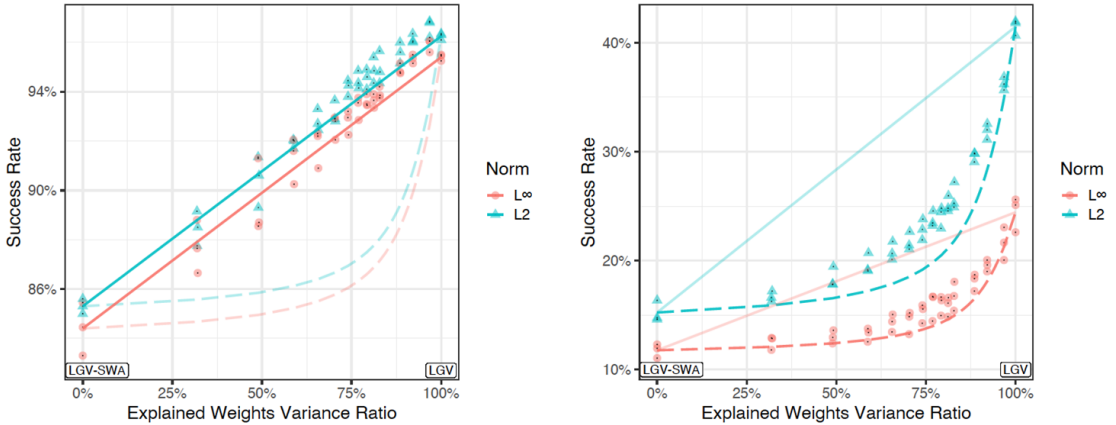
Table 9: Transfer success rate of random directions sampled in LGV deviations subspace.

Norm	Surrogate	Target							
		RN50	RN152	RNX50	WRN50	DN201	VGG19	IncV1	IncV3
L∞	LGV	95.5±0.1	85.5±2.1	83.6±1.1	82.2±2.4	69.6±1.0	67.8±0.9	58.4±0.6	25.6±1.7
L∞	LGV-SWA + RD in $\mathcal{S}$	96.0±0.2	85.6±2.5	83.6±0.6	82.1±2.8	68.6±1.1	65.7±1.5	54.5±0.9	23.5±0.4
L∞	LGV-SWA + RD	90.4±0.3	71.9±3.4	70.0±1.2	69.2±3.4	50.0±1.0	47.4±1.9	34.9±0.4	13.4±0.7
L2	LGV	96.3±0.1	90.1±1.0	88.8±0.4	87.5±1.6	79.8±1.1	78.1±1.6	71.9±0.6	43.1±0.6
L2	LGV-SWA + RD in $\mathcal{S}$	96.6±0.3	90.1±1.4	88.7±0.5	87.3±2.0	77.6±1.0	75.6±1.5	67.4±1.9	37.4±0.4
L2	LGV-SWA + RD	91.9±0.6	78.2±2.9	76.2±1.3	75.4±2.5	58.1±0.3	55.8±1.6	42.7±0.6	20.0±0.6

Random directions in LGV subspace vs. LGV

II - B) Relevance of **Geometry**

The subspace is composed of directions whose relative importance depends on the functional similarity between surrogate and target



(a) ResNet-50 Target

(b) Inception v3 Target

Fig. 5: Success rate of the LGV surrogate projected on an increasing number of dimensions with the corresponding ratio of explained variance in the weight space. Hypothetical average cases of proportionality to variance (*solid*) and equal contributions of all subspace dimensions (*dashed*). Scales not shared.

II - C) **Generic** Geometry Properties

LGV deviations can be shifted in the weight space and significantly outperform random directions

Table 10: Transfer success rate of LGV deviations shifted to other independent solutions, for target architectures in the ResNet family.

Norm	Surrogate	Target			
		RN50	RN152	RNX50	WRN50
L∞	LGV-SWA + (LGV' - LGV-SWA')	$94.3_{\pm 0.5}$	$81.5_{\pm 2.3}$	$79.1_{\pm 1.4}$	$78.1_{\pm 2.4}$
L∞	LGV-SWA + RD	$90.4_{\pm 0.3}$	$71.9_{\pm 3.4}$	$70.0_{\pm 1.2}$	$69.2_{\pm 3.4}$
L∞	LGV (ours)	$95.4_{\pm 0.1}$	$85.3_{\pm 2.1}$	$83.7_{\pm 1.1}$	$82.1_{\pm 2.5}$
L∞	1 DNN + γ (LGV' - LGV-SWA')	$73.3_{\pm 2.0}$	$52.8_{\pm 2.9}$	$52.6_{\pm 1.6}$	$56.6_{\pm 2.8}$
L∞	1 DNN + RD	$60.8_{\pm 1.6}$	$40.8_{\pm 2.7}$	$40.2_{\pm 0.3}$	$44.8_{\pm 2.7}$
L2	LGV-SWA + (LGV' - LGV-SWA')	$95.2_{\pm 0.5}$	$86.1_{\pm 1.9}$	$84.2_{\pm 1.0}$	$82.7_{\pm 1.6}$
L2	LGV-SWA + RD	$92.0_{\pm 0.5}$	$77.9_{\pm 3.0}$	$76.2_{\pm 1.4}$	$75.2_{\pm 2.8}$
L2	LGV (ours)	$96.3_{\pm 0.1}$	$90.2_{\pm 1.1}$	$88.6_{\pm 0.6}$	$87.6_{\pm 1.7}$
L2	1 DNN + γ (LGV' - LGV-SWA')	$84.2_{\pm 0.8}$	$68.7_{\pm 2.6}$	$70.0_{\pm 1.3}$	$72.4_{\pm 1.5}$
L2	1 DNN + RD	$74.6_{\pm 0.5}$	$55.8_{\pm 3.1}$	$56.1_{\pm 0.6}$	$59.9_{\pm 3.2}$

Conclusion

LGV is simple yet effective to enhance black-box attack.

1. LGV collect weights from flatter regions, which create **flatter adversarial examples** more robust to surrogate-target misalignment.
2. LGV weights span a dense **weight subspace** whose geometry is intrinsically connected to transferability.