LGV: Boosting Adversarial Example Transferability from Large Geometric Vicinity (Appendix)

In appendix of "LGV: Boosting Adversarial Example Transferability from Large Geometric Vicinity", we provide the following:

- Background and related work
- Theoretical developments of:
  - The connection between Gaussian noise in feature space and Gaussian noise in the weight space
  - The connection between attacking LGV-SWA and attacking the ensemble of LGV weights
- Additional experimental results:
  - Additional details on the experimental protocol and the studied models
  - The preliminary experiences on adding Gaussian noise in the feature space, or in the weight space, for transferability
  - Comparison to state of the art for the L_2 attack
  - The flatness in the weight space with random directions
  - The flatness in the weight space with weights interpolation
  - The flatness in feature space
  - The transferability of individual LGV weights
  - The construction of the "LGV-SWA + RD" surrogate
  - The subspace spanned by LGV weights densely related to transferability
  - Details on the decomposition of the deviation matrix
  - The shift of the subspace spanned by LGV weights to solutions
- Discussion and selection of LGV and attack hyperparameters:
  - The LGV learning rate
  - The number of LGV epochs
  - The number of collected LGV weights per epoch
  - The number of I-FGSM attack iterations

A Background and Related Work

Adversarial attacks and transferability. [24] observes early the transferability of adversarial examples: an adversarial example for one model is likely to be adversarial for another one. [20] formalizes the leverage of this property in black-box threat models. White-box attacks are applied on a surrogate model to craft adversarial examples for the unknown target model. The I-FGSM attack [15] is the workhorse of adversarial machine learning for both white-box and transfer black-box attacks. It performs gradient ascent steps projected into the L_p ball of radius ϵ centred on the original example to optimize the loss with respect to the input. See Algorithm 2 for the complete description.

Techniques for transferability. Several techniques prove useful to increase transferability. Each one provide a different perspective on the phenomenon. Up to our knowledge, no previous work boosts transferability using geometrical properties of the loss. As revealed by Section 3, our LGV approach alone consistently beats all combinations of the four following techniques. [27] favours the gradients from skip connections rather than residual modules, and claims that the formers are of first importance to generate highly transferable adversarial examples. We find the local loss geometry to have such relevance. In line with our results, residual connections flatten the natural loss [31] and increase transferability. [17] use dropout or skip connection erosion to generate Ghost Networks, and identify the diversity of surrogate models as key. Neither of these improves LGV, suggesting that they may be poor local loss geometry proxies. [29] suggest that input diversity, i.e., random transformations applied to inputs at each iteration, is a strong baseline to study transferability. [5] adds momentum to the attack gradients to stabilize them and escape from local maxima with poor transferability. We find a more effective way to do so. Overall, we shed a new major light on the transferability of adversarial examples.

Table 3: Combinations of transferability techniques evaluated by previous work.

Reference	Combinations of Techniques Evaluated
MI [5]	MI
GN [17]	GN, GN+MI
DI [29]	DI, MI, DI+MI
SGM [27]	MI, DI, SGM, MI+SGM, DI+SGM, MI+DI, MI+DI+SGM

Geometry of transferability. Previous studies [26,3] analyse the geometry of transferable adversarial examples in the input space without proposing an actionable method, whereas we study them in the weight space and provide insights to improve surrogates. On MNIST, [26] shows that among the 44 dimensions adversarial input space, a dense 25 dimensions subspace is shared between models, thus enabling transferability. [3] proves with a geometric perspective that transferable adversarial directions exist with high probability for linear classifiers trained on independent sets drawn from the same distribution.

Geometry of DNNs. Numerous work study the generalization of DNNs and SGD from a geometric perspective. [16] establishes that the intrinsic dimension of the objective landscapes is smaller than expected by applying SGD in a randomly oriented parameter subspace. [10] observes that SGD happens in a tiny parameter subspace, which is mostly preserved during training. [14] correlates large-batch SGD to both sharp solutions and a generalization gap compared to small-batch SGD. [13] shows that averaging weights along the trajectory of SGD

iterates lead to wider optima and better natural generalization than SGD. Some techniques explicitly minimizes the loss sharpness for natural [7] or robust generalization [28]. Up to our knowledge, no previous work relates loss flatness to transferability.

SGD with constant learning rate (cSGD). LGV rests upon sampling weights along the trajectory of SGD with constant learning rate. This idea has been explored extensively to improve natural accuracy or calibration in deep learning [19,13,18]. [19] proves that under some assumptions, cSGD simulates a Markov chain with a stationary distribution, which can be tuned to approximate the Bayesian posterior. Our results corroborate the relationship between the posterior predictive distribution and transferability established by [9], with a new plug-in technique. LGV is inspired by the SGD trajectories used in SWA [13], SWAG [18], and SI [12]. A key difference is that LGV uses a higher learning rate to improve attack transferability that degrades the natural accuracy of the surrogate ensemble (Figure 21). We analyse extensively SWA on top our LGV surrogate in Section 4.

B Theoretical developments

B.1 Connection between white noise in the weight space and in feature space

We develop the theoretical relation between the two DNN-based attack variants studied empirically as preliminaries in Appendix C.2: the addition of Gaussian white noise to the gradients in feature space, and the addition of Gaussian white noise in the weight space. We suppose that the loss function $\mathcal{L}(x;\, y, w)$ is twice continuously differentiable both with respect to x in the L_p ball $B_\varepsilon[x]$, and to w at w_0. To understand the failure of noise in feature space and the success of noise in the weight space, we consider the linear approximation of the input loss gradient function $\nabla_x \mathcal{L}(x'_k;\, y, \cdot) : \mathbb{R}^p \longrightarrow \mathcal{X}$, around w_0,

$$\nabla_x \mathcal{L}(x'_k;\, y, w_0 + e_k) = \nabla_x \mathcal{L}(x'_k;\, y, w_0) + \mathbf{J}_{\nabla_x \mathcal{L}(x'_k;\, y, \cdot)}(w_0)\, e_k + o(\|e_k\|), \quad (8)$$

with $\mathbf{J}_{\nabla_x \mathcal{L}(x'_k;\, y, \cdot)}(w)$ the Jacobian matrix of the input loss gradient function at w, x'_k the adversarial example at iteration k, and $e_k \sim \mathcal{N}(\mathbf{0},\, \sigma^2 I_p)$. Empirically σ is set to 5×10^{-3}, justifying the local approximation. So, at the first order approximation, the attack gradient is approximately sampled from:

$$\mathcal{N}\left(\nabla_x \mathcal{L}(x'_k;\, y, w_0),\; \sigma^2\, \mathbf{J}_{\nabla_x \mathcal{L}(x'_k;\, y, \cdot)}(w_0)\, \mathbf{J}_{\nabla_x \mathcal{L}(x'_k;\, y, \cdot)}(w_0)^T\right) \quad (9)$$

Only the noise covariance matrix changes compared to Gaussian white noise directly added in feature space. This *structured feature noise* induced by local variations of input gradients in the weight space improves transferability (Appendix C.2).

B.2 Connection between LGV-SWA and LGV ensemble surrogates

We demonstrate that the gradient of LGV-SWA approximates the gradient of the ensemble of LGV models. We show empirically in Section 3 that LGV-SWA is a good single model surrogate. We develop here its relation to the LGV weights. We extend the analysis from the original SWA paper [13] on the connection between the natural generalization of SWA and the one of local ensemble methods. Here, we suppose the loss function $\mathcal{L}(x;\, y, w)$ to be twice continuously differentiable both with respect to x in the L_p ball $B_\varepsilon[x]$, and to w at every w_k, for k in $[\![1, K]\!]$.

We perform a local analysis, since by construction, the weights collected by LGV w_k are close in the weight space and concentrated around their mean w_{SWA}. We consider the linear approximation of the input loss gradient function $\nabla_x \mathcal{L}(x;\, y, \cdot) : \mathbb{R}^p \longrightarrow \mathcal{X}$ around w_k,

$$\nabla_x \mathcal{L}(x;\, y, w_k) = \nabla_x \mathcal{L}(x;\, y, w_{\text{SWA}}) + \mathbf{J}_{\nabla_x \mathcal{L}(x;\, y, \cdot)}(w_{\text{SWA}})(w_k - w_{\text{SWA}}) + o(\|w_k - w_{\text{SWA}}\|)\,,$$

with $\mathbf{J}_{\nabla_x \mathcal{L}(x;\, y, \cdot)}(w)$ the Jacobian matrix of $\nabla_x \mathcal{L}(x;\, y, w)$ at w. The gradient of the ensemble of LGV models is the ensemble of individual gradients, $\overline{\nabla}_x \coloneqq \nabla_x \frac{1}{K} \sum_{k=1}^K \mathcal{L}(x;\, y, w_k) = \frac{1}{K} \sum_{k=1}^K \nabla_x \mathcal{L}(x;\, y, w_k)$. Then, the difference between the average of gradients and the gradient of the weights average is

$$\begin{aligned}
&\overline{\nabla}_x - \nabla_x \mathcal{L}(x;\, y, w_{\text{SWA}}) \\
&= \frac{1}{K} \sum_{k=1}^K \left[\mathbf{J}_{\nabla_x \mathcal{L}(x;\, y, \cdot)}(w_{\text{SWA}})(w_k - w_{\text{SWA}}) + o(\|w_k - w_{\text{SWA}}\|) \right] \\
&= \mathbf{J}_{\nabla_x \mathcal{L}(x;\, y, \cdot)}(w_{\text{SWA}}) \left(\frac{1}{K} \sum_{k=1}^K w_k - w_{\text{SWA}} \right) + o(\|\Delta_w\|) \\
&= o(\|\Delta_w\|),
\end{aligned}$$

with $\Delta_w = \max_{k=1}^K (\|w_k - w_{\text{SWA}}\|)$. It follows that LGV-SWA is a good single-model approximation of the ensemble of LGV models for gradient-based attacks. It captures some diversity of gradients in the vicinity of the weight space.

C Additional Experimental Results

C.1 Experimental Settings

Target models We select 8 pretrained models distributed by the *torchvision* library [21]. The architectures are diverse and belong to different families. We choose ResNet-50 to study the intra-architecture transferability, ResNet-152 for the effect of increasing the number of layers, ResNeXt-50 32×4d and WideResNet-50-2 for other variants in the ResNet family, DenseNet-201, VGG19, Inception v1 (GoogLeNet) and Inception v3 to represent other families. Table 4 contains their natural accuracy and negative log likelihood (NLL).

Table 4: Natural accuracy and loss of target models computed on the test set.

Name	Architecture	Test Accuracy	Loss (NLL)
RN50	ResNet-50	76.01%	0.963
RN152	ResNet-152	78.25%	0.876
RNX50	ResNext-50	77.63%	0.941
WRN50	WideResNet-50	78.46%	0.883
DN201	DenseNet-201	76.93%	0.926
VGG19	VGG19	72.36%	1.115
IncV1	Inception v1 (GoogLeNet)	69.74%	1.283
IncV3	Inception v3	76.25%	1.041

Surrogate models We retrieve the independently trained ResNet-50 DNNs from [1]. All DNNs share the same hyperparameters, and have different random initializations. For each experiment run, we select without replacement a random DNN, and call it interchangeably "1 DNN", the initial DNN, or the DNN with weights w_0. LGV starts from the weights w_0 of this DNN. . "1 DNN + RD" is defined in Appendix C.2 and "LGV-SWA" in Section 2. Table 5 contains their natural accuracy and negative log likelihood.

Table 5: Natural accuracy and loss of surrogate models computed on the test set.

Method	Test Accuracy	Loss (NLL)	Number of models
1 DNN	76.14% ± 0.14	0.945 ± 0.003	1
1 DNN + RD	76.17% ± 0.10	0.948 ± 0.003	50
LGV-SWA	72.17% ± 0.10	1.128 ± 0.002	1
LGV (ours)	70.83% ± 0.10	1.310 ± 0.011	40

Threat model We study untargeted adversarial examples, the attack objective is misclassification. We consider the less restrictive threat model for transfer-based attack, where no query access to the target model is granted. Therefore, we do not compare with query-based black-box attacks. This experimental setup is standard for transfer-based attack evaluation.

Attack The I-FGSM attack performs 50 iterations with a step size equal to one tenth of the maximum perturbation norm ε. For the L_∞ attack ε equals $\frac{4}{255}$, and for the L_2 one it equals 3. The number of iterations is selected on a validation set for both the initial DNN surrogate and its resulting LGV surrogate (Figure 25). Each iteration compute a single gradient on a randomly selected model if several are available for the method considered. If the number of iterations is higher than the number of models, we cycle on models in the same order. Therefore,

the attack cost, measured as the number of backward passes, is kept constant regardless of the number of the size of the surrogate. We do not consider the ensemble of models for fairness with the single model surrogate baselines.

Batch normalisation Each time our experiments change weights (e.g. when we apply random directions), we perform an additional forward pass over 10% of the training data to update the batch-normalization statistics, following previous studies [12,18]. Translations in the weight space cause internal covariate shift, which may causes our surrogate to fail, regardless of the quality of the corresponding point in the weight space. We control this undesired experimental artefact by updating batch-normalization statistics. LGV does not need such extra computational cost, since regular training updates batch-normalization statistics on the fly.

Figures Figures containing multiple subplots report the success rate on the target indicated in subplot title of adversarial examples crafted against the surrogate indicated in legend or in caption. In all figures containing lines surrounded with a lighter coloured area, the lines are smoothed means[5] over 3 independent runs, and the coloured areas correspond to one standard deviation around the mean.

Implementation All experiments source code and models are available on GitHub[6]. We adapt the I-FGSM attack from the Python ART library to support the four state-of-the-art transferability techniques. The training of LGV and some experiments are adapted from the code of [12] on PyTorch. We use the following software versions: Python 3.8.8, PyTorch 1.7.1, Torchvision 0.8.2, Adversarial Robustness Toolbox 1.6.0, and Scikit-learn 0.23.2.

Infrastructure The GPU used for the experiments is Tesla V100-DGXS-32GB, on a server with the following specifications: 256GB RDIMM DDR4, CUDA version 10.1, Linux (Ubuntu) operating system.

Hyperparameters We use the hyperparameters for training indicated in Table 6, and for the attack in Table 7.

[5] The smooth means are local polynomial regressions computed by the "loess()" function of the R stats package.

[6] URL redacted for review. Reproducible code is provided as supplementary materials.

Table 6: Hyperparameters used to train LGV and the initial DNN.

Hyperparameter	1 DNN	LGV
Learning rate schedule	Step size decay $\times 0.1$ each 30 epochs	Constant
Initial learning rate	0.1	0.05
Number of epochs	130	10
Optimizer	SGD	SGD
Momentum	0.9	0.9
Batch-size	256	256
Weight decay	1×10^{-4}	1×10^{-4}

Table 7: Hyperparameters of the I-FGSM attack and transferability techniques.

Attack	Hyperparameter	Values
I-FGSM	Perturbation norm ε	3 for L_2, $\frac{4}{255}$ for L_∞
I-FGSM	Step-size α	$\frac{\varepsilon}{10}$
I-FGSM	Number iterations	50
Momentum (MI)	Momentum term	0.9
Ghost Network (GN)	Skip connection erosion random range	$[1 - 0.22, 1 + 0.22]$
Input Diversity (DI)	Minimum resize ratio	90%
Input Diversity (DI)	Probability transformation	0.5
Skip Gradient Method (SGM)	Residual gradient decay γ	0.2

C.2 Preliminaries: Transferability from the Weight Space

We aim to show the importance of the geometry of the surrogate loss in improving transferability. We experimentally demonstrate that adding random directions in the weight space to an existing surrogate increases transferability, whereas random directions in the feature space applied on gradients do not. We conclude that the structure of gradient noise from the local variations in weight space of the surrogate architecture is key to improving transferability (Appendix B.1).

White Noise on Features We first establish that random directions in feature space do not increase transferability. [25] observe that adding a random step to the single-step FGSM attack hinders transferability. We extend this conclusion to the I-FGSM attack. We add Gaussian white noise to the input gradients of the loss function during the attack, $\nabla_x \mathcal{L}(x'_k; y, w_0) + e''_k$ with $e''_k \sim \mathcal{N}(\mathbf{0}, \sigma''^2 I_d)$ where x'_k is the adversarial example at the kth attack iteration. Gradient noise does not improve the success rate (over all considered architectures), regardless of the standard deviation value σ'' used (from 1×10^{-7} to 1×10^{-2} ; Figure 6).

White Noise on Weights Next, we demonstrate that sampling random directions in the surrogate weight space increases transferability. At each I-FGSM iteration k, we add Gaussian white noise to every weight w_0 to compute the input gradient, $\nabla_x \mathcal{L}(x'_k; y, w_0 + e_k)$ with $e_k \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_p)$. As weights belong to high dimensions[7], the resulting surrogate is approximately uniformly distributed on the sphere centred on w_0 with radius $\sigma\sqrt{p}$. We found a consistent and significant improvement of transferability – from 1.1 to 20.8 percentage points of success rate – for all eight target architectures and both L_2 and L_∞ attacks, compared to the initial weights w_0. We call the attack *RD* for random directions in Table 1 (L_∞) and Table 8 (L_2). RD is reported with standard deviation σ equal to 5×10^{-3} and one random direction per attack iteration. The noise standard deviation is selected by cross-validation on a validation set (Figure 7). Due to computational limitations to update the batch normalisation statistics, we sample only 10 random directions for cross-validation and cycle between samples during the 50 attack iterations.

In Appendix B.1, we show that sampling random directions in the weight space increases transferability due to the *structured feature noise* induced by the surrogate architecture. We develop the connection between feature space noise and weight space noise in Appendix B.1 and show that the latter boils down to adding a structured Gaussian noise in feature space with a covariance matrix based on local variations in weight space (to the first order approximation).

As a side note and in line with our results in Section 4.1, we observe in Appendix C.6 that "RD" produces flatter adversarial examples than its vanilla DNN counterpart.

[7] The number of ResNet-50 weights p is 25 557 032.

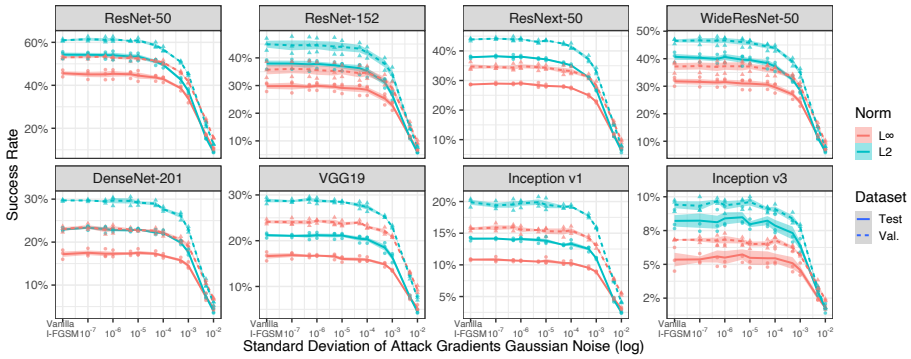

Fig. 6: Transfer success rate of I-FGSM with respect to the standard deviation of the Gaussian white noise added to the inputs gradients (pseudo-log scale). The null standard deviation is vanilla I-FGSM. The subplot title is the target architecture. The first subplot is intra-architecture transferability.

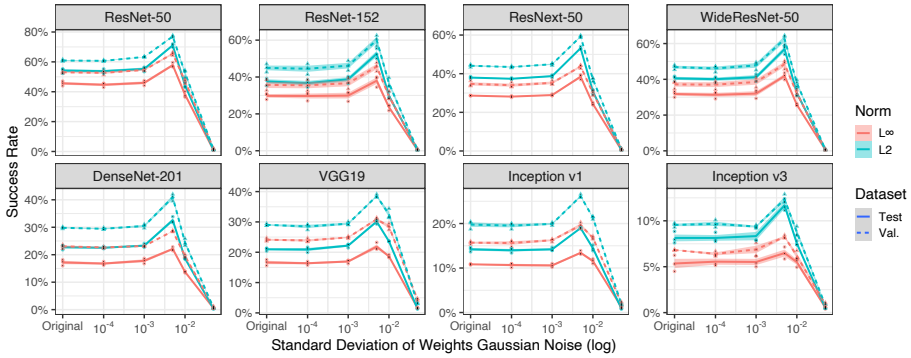

Fig. 7: Transfer success rate of I-FGSM with respect to the standard deviation of the Gaussian white noise added to the weight of the initial DNN. Ten random directions are sampled in weight space. The subplot title is the target architecture. The first subplot is intra-architecture transferability.

C.3 Comparison to State of the Art

Similarly to Table 1, we report here the L_2 attack success rates of LGV, its variants, four state-of-the-art methods, and their combination with LGV. As for the L_∞ attack, LGV alone improves over all (combinations of) other techniques (simple underline). Contrary to the L_∞ case, the vanilla LGV intra-architecture L_2 attack outperforms all techniques applied on LGV, with a margin larger than the sum of the respective standard deviations. In six out of seven inter-architecture targets, input diversity (DI) on top of LGV is best, and in one case, vanilla LGV is.

Table 8: Success rates of baselines, state-of-the-art and LGV under the L2 attack. Simple underline is best without LGV combinations, double is best overall. Gray is LGV-based techniques worse than vanilla LGV. "RD" stands for random directions in the weight space. In %.

	Target							
Surrogate	RN50	RN152	RNX50	WRN50	DN201	VGG19	IncV1	IncV3
Baselines (1 DNN)								
1 DNN	53.9$_{\pm 2.0}$	37.9$_{\pm 2.0}$	37.9$_{\pm 0.1}$	40.5$_{\pm 2.0}$	22.7$_{\pm 0.5}$	21.1$_{\pm 0.6}$	13.6$_{\pm 0.2}$	7.9$_{\pm 0.7}$
MI	48.7$_{\pm 1.2}$	33.5$_{\pm 1.3}$	33.7$_{\pm 0.7}$	36.5$_{\pm 1.8}$	19.9$_{\pm 0.3}$	19.3$_{\pm 0.8}$	12.0$_{\pm 0.5}$	6.6$_{\pm 0.5}$
GN	77.2$_{\pm 0.9}$	60.1$_{\pm 1.3}$	59.6$_{\pm 2.0}$	63.4$_{\pm 2.0}$	37.3$_{\pm 1.4}$	33.6$_{\pm 0.5}$	21.1$_{\pm 0.8}$	12.1$_{\pm 1.0}$
GN+MI	68.4$_{\pm 2.0}$	50.4$_{\pm 1.9}$	49.8$_{\pm 1.4}$	53.3$_{\pm 1.2}$	29.0$_{\pm 1.2}$	27.3$_{\pm 0.2}$	16.5$_{\pm 0.5}$	9.0$_{\pm 1.0}$
DI	82.2$_{\pm 0.6}$	68.0$_{\pm 1.6}$	71.8$_{\pm 0.6}$	72.5$_{\pm 1.9}$	53.8$_{\pm 0.4}$	49.8$_{\pm 1.1}$	37.5$_{\pm 0.9}$	25.4$_{\pm 1.3}$
DI+MI	79.2$_{\pm 0.4}$	63.7$_{\pm 1.1}$	66.8$_{\pm 0.6}$	68.2$_{\pm 1.4}$	47.3$_{\pm 0.9}$	45.8$_{\pm 1.0}$	32.0$_{\pm 0.7}$	20.6$_{\pm 0.9}$
SGM	63.3$_{\pm 0.5}$	49.7$_{\pm 3.1}$	50.1$_{\pm 0.7}$	51.6$_{\pm 1.7}$	30.2$_{\pm 0.7}$	33.3$_{\pm 1.3}$	20.6$_{\pm 0.9}$	11.1$_{\pm 0.7}$
SGM+MI	63.3$_{\pm 0.4}$	49.2$_{\pm 3.7}$	50.1$_{\pm 0.6}$	51.9$_{\pm 1.5}$	29.6$_{\pm 0.3}$	33.9$_{\pm 1.4}$	21.4$_{\pm 0.5}$	11.6$_{\pm 0.8}$
SGM+DI	79.5$_{\pm 0.7}$	67.5$_{\pm 2.3}$	69.0$_{\pm 0.9}$	69.6$_{\pm 1.1}$	50.1$_{\pm 0.2}$	54.6$_{\pm 1.3}$	40.5$_{\pm 0.8}$	25.8$_{\pm 1.0}$
SGM+DI+MI	78.5$_{\pm 0.8}$	66.4$_{\pm 2.4}$	68.5$_{\pm 1.7}$	69.1$_{\pm 1.2}$	49.1$_{\pm 1.4}$	54.5$_{\pm 0.9}$	39.7$_{\pm 0.8}$	25.6$_{\pm 0.3}$
Our techniques								
RD	74.4$_{\pm 0.6}$	55.6$_{\pm 3.1}$	55.9$_{\pm 0.7}$	59.7$_{\pm 3.3}$	34.5$_{\pm 0.3}$	31.5$_{\pm 1.4}$	19.6$_{\pm 0.8}$	11.2$_{\pm 1.0}$
LGV-SWA	85.8$_{\pm 0.7}$	68.0$_{\pm 3.4}$	67.0$_{\pm 0.4}$	65.1$_{\pm 1.8}$	48.4$_{\pm 0.7}$	47.0$_{\pm 1.6}$	34.8$_{\pm 0.5}$	15.8$_{\pm 1.1}$
LGV-SWA+RD	92.0$_{\pm 0.5}$	77.9$_{\pm 3.0}$	76.2$_{\pm 1.4}$	75.2$_{\pm 2.8}$	58.1$_{\pm 0.3}$	55.6$_{\pm 1.9}$	42.7$_{\pm 0.6}$	20.2$_{\pm 0.5}$
LGV (ours)	96.3$_{\pm 0.2}$	90.1$_{\pm 0.9}$	88.7$_{\pm 0.5}$	87.2$_{\pm 1.8}$	79.6$_{\pm 1.2}$	78.0$_{\pm 1.6}$	71.8$_{\pm 0.5}$	42.8$_{\pm 0.4}$
LGV combined with other techniques								
MI	96.0$_{\pm 0.1}$	88.3$_{\pm 1.7}$	85.8$_{\pm 0.7}$	84.3$_{\pm 2.6}$	72.6$_{\pm 0.8}$	71.8$_{\pm 1.9}$	62.7$_{\pm 0.7}$	31.1$_{\pm 0.3}$
GN	95.8$_{\pm 0.5}$	89.3$_{\pm 1.6}$	87.6$_{\pm 0.6}$	85.8$_{\pm 1.8}$	77.7$_{\pm 1.0}$	77.5$_{\pm 0.6}$	71.0$_{\pm 0.6}$	41.5$_{\pm 1.5}$
GN+MI	95.3$_{\pm 0.4}$	86.1$_{\pm 2.2}$	84.1$_{\pm 0.6}$	82.6$_{\pm 2.4}$	71.0$_{\pm 1.2}$	71.1$_{\pm 1.1}$	62.0$_{\pm 0.8}$	30.2$_{\pm 0.5}$
DI	95.3$_{\pm 0.3}$	89.5$_{\pm 0.9}$	89.5$_{\pm 0.5}$	87.3$_{\pm 0.9}$	83.9$_{\pm 0.9}$	83.7$_{\pm 0.2}$	82.2$_{\pm 0.9}$	59.0$_{\pm 0.8}$
DI+MI	95.2$_{\pm 0.4}$	88.6$_{\pm 0.7}$	88.0$_{\pm 0.7}$	85.7$_{\pm 1.5}$	81.2$_{\pm 0.7}$	81.6$_{\pm 0.7}$	79.3$_{\pm 1.6}$	50.8$_{\pm 0.7}$
SGM	85.8$_{\pm 0.5}$	74.1$_{\pm 2.5}$	73.4$_{\pm 0.7}$	71.6$_{\pm 2.1}$	59.5$_{\pm 0.8}$	68.5$_{\pm 1.4}$	62.2$_{\pm 2.1}$	34.4$_{\pm 1.9}$
SGM+MI	85.0$_{\pm 0.7}$	73.3$_{\pm 2.3}$	72.5$_{\pm 0.9}$	70.1$_{\pm 1.9}$	57.5$_{\pm 0.3}$	67.6$_{\pm 1.2}$	60.7$_{\pm 1.9}$	33.0$_{\pm 1.5}$
SGM+DI	85.0$_{\pm 1.1}$	75.2$_{\pm 1.4}$	75.5$_{\pm 0.7}$	72.5$_{\pm 1.7}$	65.2$_{\pm 0.8}$	74.2$_{\pm 1.6}$	71.6$_{\pm 1.6}$	46.0$_{\pm 1.7}$
SGM+DI+MI	84.4$_{\pm 0.6}$	74.0$_{\pm 1.4}$	74.9$_{\pm 0.8}$	71.7$_{\pm 1.2}$	63.8$_{\pm 0.7}$	73.3$_{\pm 1.4}$	70.3$_{\pm 1.4}$	44.6$_{\pm 1.4}$

C.4 Flatness — Random Directions in the Weight Space

In addition to results in Section 4.1, we report in Figure 8, the loss of adversarial examples crafted along 10 random directions in the weight space, for both norms and evaluated on the eight targets. We confirm on the L_2 attack and on the inter-architecture case that the increased flatness of LGV-SWA in the weight space comes with an increased transferability of LGV-SWA adversarial examples, compared to the initial DNN.

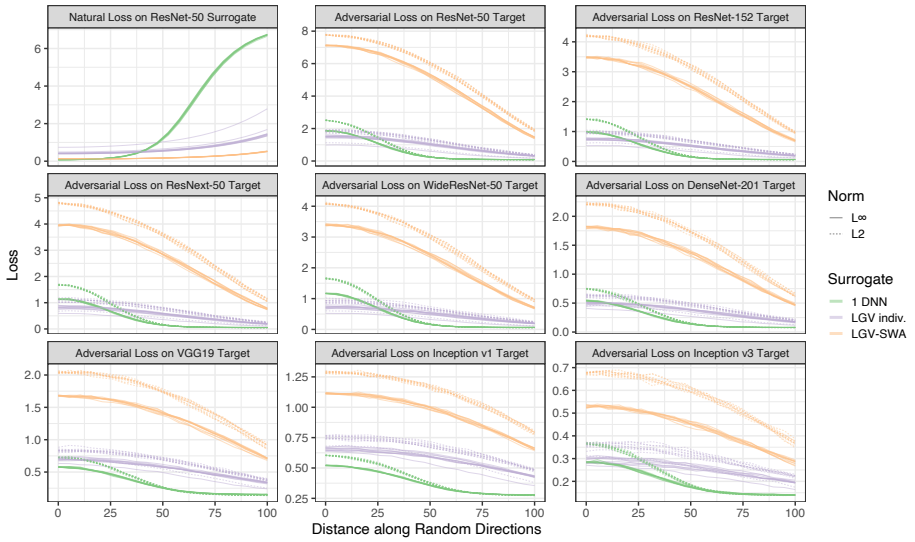

Fig. 8: Surrogate natural loss (*first subplot*) and adversarial target loss (*other subplots*) with respect to the 2-norm distance along 10 random directions in the weight space from the initial model (*green*), LGV-SWA (*orange*) and randomly drawn individual LGV weights (*purple*). For adversarial target losses, plain lines are L_∞ and dashed ones are L_2. Ordinate scale not shared.

C.5 Flatness — Interpolation in Weight Space between LGV-SWA and the Initial Model

We confirm the observations in Section 4.1 about flatness on another specific direction. None of the 10 studied random directions increases transferability on their own[8]. However, we know that at least one behave differently, since

[8] This monotonic decrease in random directions does not contradict our findings in Appendix C.2. Here, all the I-FGSM attack iterations are applied on a single surrogate, whereas previously each iteration was performed on a new *iid* sample.

transferability increases from the initial DNN to LGV-SWA (Section 3). As [13], we study the path in the weight space connecting both with $\alpha \in \mathbb{R}$:

$$w(\alpha) = \alpha w_0 + (1 - \alpha) w_{\text{SWA}}$$

We observe the same correlation between the flatness of the natural surrogate loss (orange) and the target adversarial loss (blue and red) in Figure 9. Around the LGV-SWA solution, the natural loss is flatter than around the initial DNN where it explodes at α close to 1.2. The same conclusions hold for all target architectures. Interestingly, LGV-SWA is not always the best single surrogate. The best surrogate in the studied segment is achieved for values of α between 0.154 and 0.538 for target architectures that belong to the ResNet family. The natural loss looks also flat in this region, so this does not contradict our observation. In conclusion, LGV produces weights on a flatter region of the loss landscape than where it starts from.

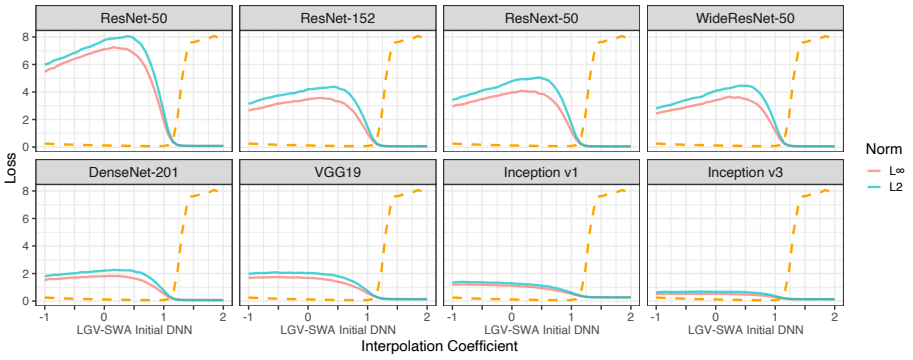

Fig. 9: Adversarial target loss (*plain*) and surrogate natural loss (*orange dashed*) with respect to the interpolation coefficient α between the LGV-SWA solution and the initial model. The subplot title is the target architecture. The first subplot is intra-architecture transferability.

C.6 Flatness in Feature Space

We show that flat surrogates in the weight space produce flatter adversarial examples in the feature space. We report here visualisations of the plane in feature space defined in Section 4.1 for all eight targets and with several combinations of surrogates. We recall that each plane contains three points: the original example, and two adversarial examples crafted on the two surrogates of interest. The first two steps of the Gram–Schmidt process defines an orthonormal basis (u', v').

As shown in Section 4.1, the initial DNN is sharper than individual LGV models in the weight space, and LGV-SWA is flatter than both. We show here that the order of flatness is the same in the weight space and in feature space. Figure 11 shows that LGV-SWA produces flatter adversarial examples. A randomly sampled individual LGV weights surrogate leads to flatter adversarial examples than the initial DNN (Figure 13), but sharper than the LGV-SWA (Figure 14) and the LGV (Figure 15) surrogates.

Appendix B.2 shows that LGV-SWA is a good approximation to the ensemble of LGV weights. LGV transfers better than LGV-SWA (Tables 1 and 8). We observe in Figure 12 that the adversarial examples of the former are flatter in average than the ones of the latter. The optimization of the I-FGSM attack overfits the single set of weights approximation, leading to sharper minima. We also observe that the form of the contours around the LGV-SWA surrogate can be explained by a shift between target and surrogate.

Appendix C.2 exhibits that noise applied to the weights of a DNN increases transferability. Figure 16 establishes that this noise also slightly flatten the adversarial examples in feature space.

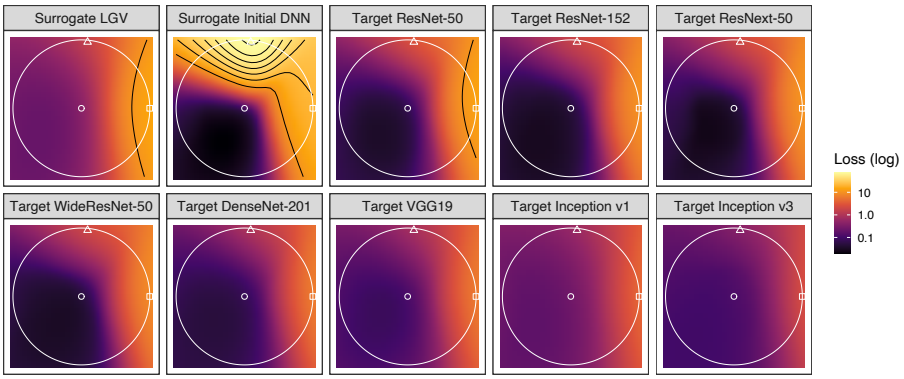

Fig. 10: **LGV** surrogate (*first up*), the **initial DNN** surrogate (*second up*) and targets (*others*) losses in the plane containing the original example (*circle*), an adversarial example against LGV (*square*) and one against the initial DNN (*triangle*), in the (u', v') coordinate system. Colours are in log-scale, contours in natural scale. The white circle represents the intersection of the 2-norm ball with the plane.

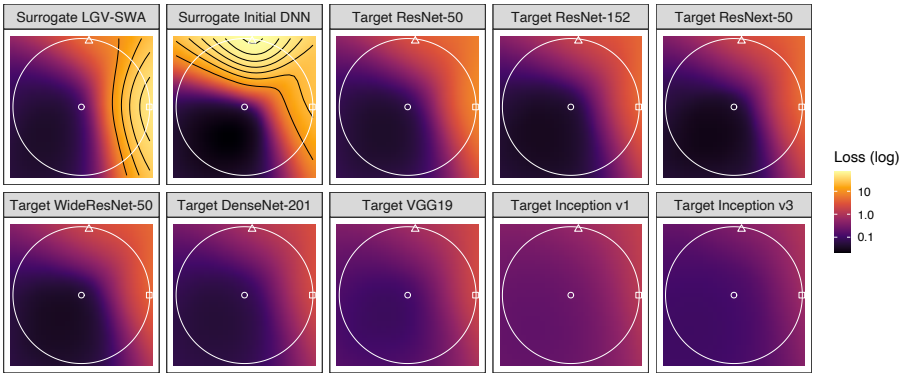

Fig. 11: **LGV-SWA** surrogate (*first up*), the **initial DNN** surrogate (*second up*) and targets (*others*) losses in the plane containing the original example (*circle*), an adversarial example against LGV-SWA (*square*) and one against the initial DNN (*triangle*), in the (u', v') coordinate system. Colours are in log-scale, contours in natural scale. The white circle represents the intersection of the 2-norm ball with the plane.

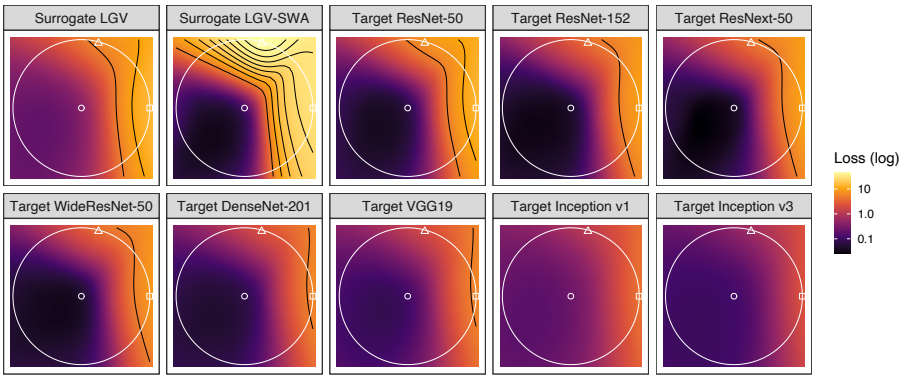

Fig. 12: **LGV** surrogate (*first up*), the **LGV-SWA** surrogate (*second up*) and targets (*others*) losses in the plane containing the original example (*circle*), an adversarial example against LGV (*square*) and one against LGV-SWA (*triangle*), in the (u', v') coordinate system. Colours are in log-scale, contours in natural scale. The white circle represents the intersection of the 2-norm ball with the plane.

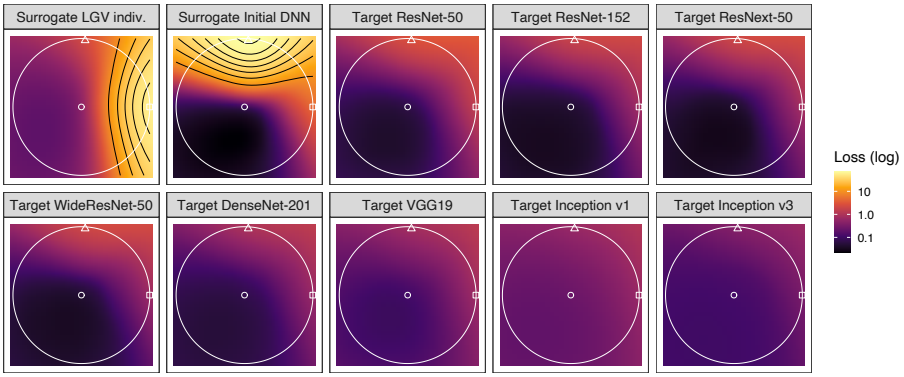

Fig. 13: A randomly sampled **individual LGV weights** surrogate (*first up*), the **initial DNN** surrogate (*second up*) and targets (*others*) losses in the plane containing the original example (*circle*), an adversarial example against the individual LGV weights (*square*) and one against the initial DNN (*triangle*), in the (u', v') coordinate system. Colours are in log-scale, contours in natural scale. The white circle represents the intersection of the 2-norm ball with the plane.

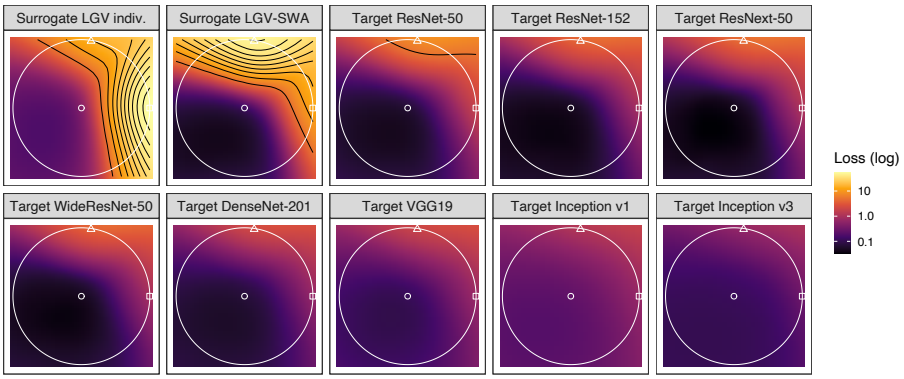

Fig. 14: A randomly sampled **individual LGV weights** surrogate (*first up*), **LGV-SWA** surrogate (*second up*) and targets (*others*) losses in the plane containing the original example (*circle*), an adversarial example against the individual LGV weights (*square*) and one against LGV-SWA (*triangle*), in the (u', v') coordinate system. Colours are in log-scale, contours in natural scale. The white circle represents the intersection of the 2-norm ball with the plane.

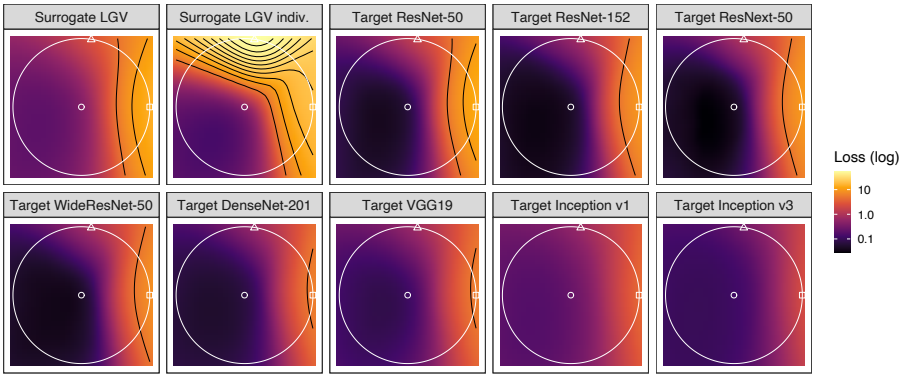

Fig. 15: **LGV** surrogate (*first up*), a randomly sampled **individual LGV weights** surrogate (*second up*) and targets (*others*) losses in the plane containing the original example (*circle*), an adversarial example against LGV (*square*) and one against the individual LGV weights (*triangle*), in the (u', v') coordinate system. Colours are in log-scale, contours in natural scale. The white circle represents the intersection of the 2-norm ball with the plane.

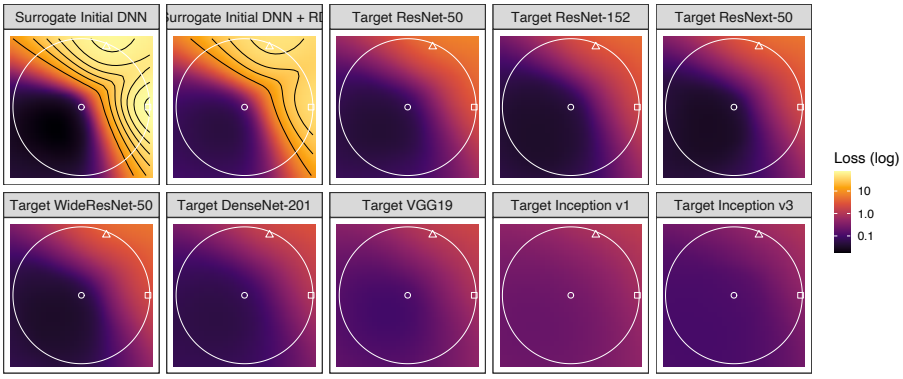

Fig. 16: The **initial DNN** surrogate (*first up*), the **initial DNN + random directions** surrogate (*second up*) and targets (*others*) losses in the plane containing the original example (*circle*), an adversarial example against the initial DNN (*square*) and one against the initial DNN + random directions (*triangle*), in the (u', v') coordinate system. Colours are in log-scale, contours in natural scale. The white circle represents the intersection of the 2-norm ball with the plane.

C.7 Individual LGV Weights

The success of LGV cannot be explained by the intrinsic properties of each of its model taken on its own. Figure 17 shows that no single model sampled by LGV improves consistently upon the baseline of the initial model it originates from. On the contrary, the initial DNN is generally a better surrogate.

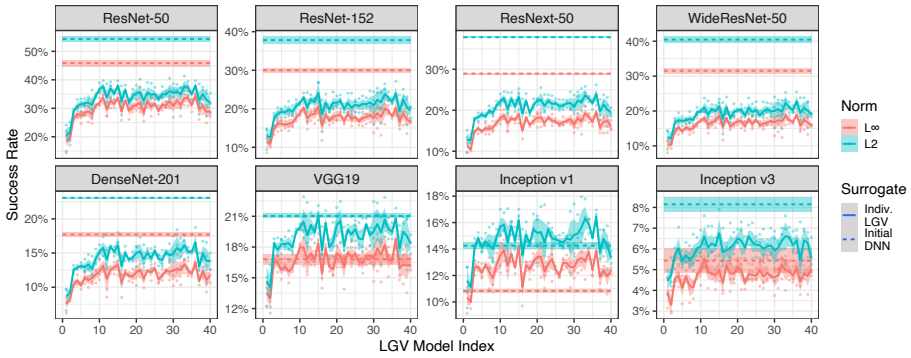

Fig. 17: Transfer success rate of each individual LGV weights indexed by the sampling order (*plain*) and the initial DNN baseline (*dashed*). The subplot title is the target architecture. The first subplot is intra-architecture transferability. Ordinate scale not shared.

C.8 Random Directions around LGV-SWA ("LGV-SWA + RD")

The "LGV-SWA + RD" surrogate is defined by:

$$\{w_{\text{SWA}} + e'_k \mid e'_k \sim \mathcal{N}(\mathbf{0}, \sigma' I_p),\, k \in [\![1, K]\!]\}\,, \tag{10}$$

where σ' is selected by cross validation. Figure 18 reports the success rate for various values of σ'. Similarly to "RD" in Appendix C.2, we tune this hyper-parameter on 10 random directions, and generate the final "LGV-SWA + RD" surrogate on 50 directions.

In accordance with our findings about the respective flatness of the DNNs and LGV-SWA, the optimal standard deviation for "LGV-SWA + RD" (1×10^{-2}) is larger than the one for "RD" (5×10^{-3}). A flatter solution implies that we can sample further along random directions before exiting the vicinity of low loss.

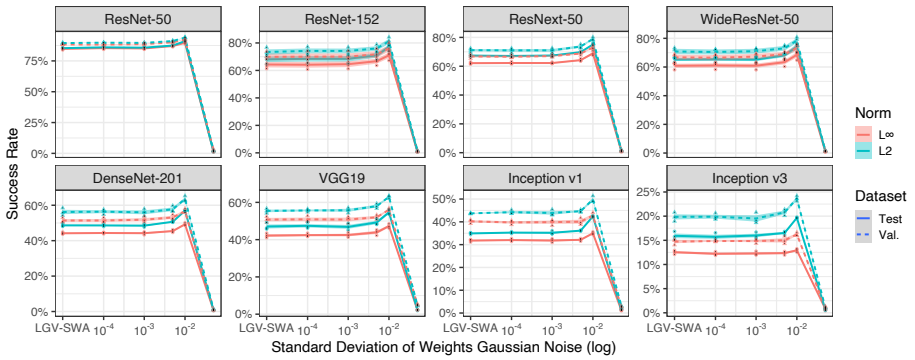

Fig. 18: Transfer success rate of I-FGSM with respect to the standard deviation σ' of the Gaussian white noise added to the weight of LGV-SWA. Ten random directions are sampled in the weight space. The subplot title is the target architecture. The first subplot is intra-architecture transferability.

C.9 Random Directions in LGV Subspace

We show that the LGV deviation subspace is *densely* related to transferability, in the sense that it is a dense subspace of good surrogates. We form a new surrogate called "LGV-SWA + RD in $\mathcal{S}$" by sampling random directions in the LGV deviations subspace $\mathcal{S}$,

$$\{w_{\text{SWA}} + \mathbf{P}z_k \mid z_k \sim \mathcal{N}(\mathbf{0}, I_K),\ k \in [\![1, K]\!]\} \subset \mathcal{S}, \tag{11}$$

where $\mathbf{P} = (w_1 - w_{\text{SWA}}, \dots, w_K - w_{\text{SWA}})^\intercal$ is the projection matrix of LGV weights deviations from their mean. Table 9 reports the success rates of this surrogate along with other techniques. We observe that the transferability of random directions in the subspace is close to the original LGV surrogate (average difference of 1.45 percentage point, with values between -0.6 and 5.65), especially for ResNet-like targets. The negative difference correspond to the intra-architecture transferability, where "LGV-SWA + RD in $\mathcal{S}$" outperforms LGV (significantly for L∞ and non-significantly for L2). Sampling random directions in the full weigh space ("LGV-SWA + RD") instead of the subspace hinders transferability of 14.7 percentage points in average (4.7—24.73). Therefore, the subspace $\mathcal{S}$ is densely and intrinsically 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$_{\pm 0.1}$	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$_{\pm 0.6}$

C.10 Decomposition of the LGV Projection Matrix

In addition to Figure 5 in Section 4.2, we report in Figure 19 the success rate of surrogates projected into an increasing number of eigenvectors of the LGV weights deviations matrix $\mathbf{P}$, evaluated on all eight targets. The plain lines are smoothed means (local polynomial regression). We observe that the relationship between the weight space variance ratio and the transferability gets progressively less linear as the target architecture is different from the surrogate one (ResNet-50 here). Inception family targets are pretty close to the case of equal contribution of each dimension to transferability (dashed), independently of its variance. From an information theory perspective view of PCA, this would mean that the information contained in the weight space is directly relevant for intra-architecture transferability, and is not discriminant for dissimilar targets[9]. DenseNet-201 and VGG19 are intermediary cases. We show that the increase in transferability from the subspace $\mathcal{S}$ depends fundamentally on the functional similarity between the target and the surrogate architectures.

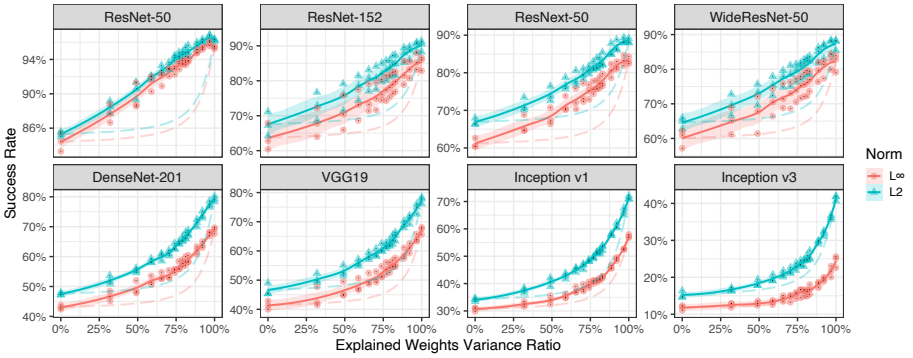

Fig. 19: Transfer success rate of the LGV surrogate projected on an increasing number of dimensions with the corresponding ratio of explained variance in the weight space. The plain line is the smooth mean, and the area corresponds to one standard deviation. The dashed line is the hypothetical average case of equal contributions of all subspace dimensions. Ordinate scale not shared.

[9] However, we recall that these directions remain more relevant than random directions in the full weight space, even for these target architectures.

C.11 Shift of LGV Subspace to Other Solutions

Tables 10 and 11 reports detailed results about the shifts of LGV deviations to other shift vectors. Results are discussed in Section 4.2. In the following paragraph, we cover specifically the scaling of LGV deviations for the shift to a regularly trained DNN.

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

		Target			
Norm	Surrogate	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$_{\pm 3.2}$

Table 11: Transfer success rate of LGV deviations shifted to other independent solutions, for non-ResNet targets.

		Target			
Norm	Surrogate	DN201	VGG19	IncV1	IncV3
L∞	LGV-SWA + (LGV' - LGV-SWA')	62.2$_{\pm 0.4}$	57.4$_{\pm 1.5}$	45.4$_{\pm 0.6}$	18.7$_{\pm 0.5}$
L∞	LGV-SWA + RD	50.0$_{\pm 1.0}$	47.5$_{\pm 1.9}$	34.9$_{\pm 0.4}$	13.4$_{\pm 0.7}$
L∞	LGV (ours)	69.7$_{\pm 1.0}$	67.5$_{\pm 1.1}$	58.6$_{\pm 0.8}$	25.4$_{\pm 1.5}$
L∞	1 DNN + γ (LGV' - LGV-SWA')	32.6$_{\pm 0.2}$	30.0$_{\pm 0.9}$	18.4$_{\pm 0.1}$	9.6$_{\pm 0.3}$
L∞	1 DNN + RD	23.1$_{\pm 0.8}$	22.8$_{\pm 0.4}$	14.1$_{\pm 0.1}$	6.8$_{\pm 0.6}$
L2	LGV-SWA + (LGV' - LGV-SWA')	69.8$_{\pm 0.6}$	65.7$_{\pm 0.7}$	55.0$_{\pm 1.0}$	27.4$_{\pm 0.5}$
L2	LGV-SWA + RD	58.1$_{\pm 0.3}$	55.6$_{\pm 1.9}$	42.7$_{\pm 0.6}$	20.2$_{\pm 0.5}$
L2	LGV (ours)	79.6$_{\pm 1.1}$	78.0$_{\pm 1.5}$	71.8$_{\pm 0.6}$	42.9$_{\pm 0.9}$
L2	1 DNN + γ (LGV' - LGV-SWA')	47.4$_{\pm 0.9}$	42.2$_{\pm 0.2}$	29.2$_{\pm 0.3}$	17.2$_{\pm 0.2}$
L2	1 DNN + RD	34.7$_{\pm 0.3}$	31.5$_{\pm 1.3}$	19.8$_{\pm 0.7}$	11.4$_{\pm 1.1}$

Scale of LGV deviations shifted to another DNN To shift LGV deviations to another independently obtained DNN, we need to consider that this new shift

vector is sharper than LGV-SWA. A sharper shift vector means that deviations around it needs to be smaller to stay in the desirable vicinity. We adapt LGV deviations, scaling them by a scalar called γ. The surrogate obtained by shifting independently obtained LGV' deviations to the initial model w_0 is:

$$\{w_0 + \gamma(w'_k - w'_{\text{SWA}}) \mid k \in [\![1, K]\!]\}\,. \tag{12}$$

We choose the γ hyperparameter by cross-validation (Figure 20). For computational efficiency, we randomly draw without replacement a subset of 10 LGV' deviations for each random seed.

The original scale of LGV deviations is clearly not appropriate for a DNN. The optimal γ value is 0.5 for all eight targets. The difference between the original scale ($\gamma = 1$) and the optimal one ($\gamma = 0.5$) is as high as 32.8 percentage points on average (6.52–59.47). Therefore, considering the flatness around the shift vector is of first importance to construct a good surrogate from weight deviations.

The optimal scale is consistent with the previously found optimal length along random directions. The optimal Gaussian standard deviation for a DNN is also half of the one optimal for LGV-SWA. Flatness is consistent in that aspect between LGV and random subspaces. These observations also corroborate our observations that LGV-SWA is flatter than the initial DNN.

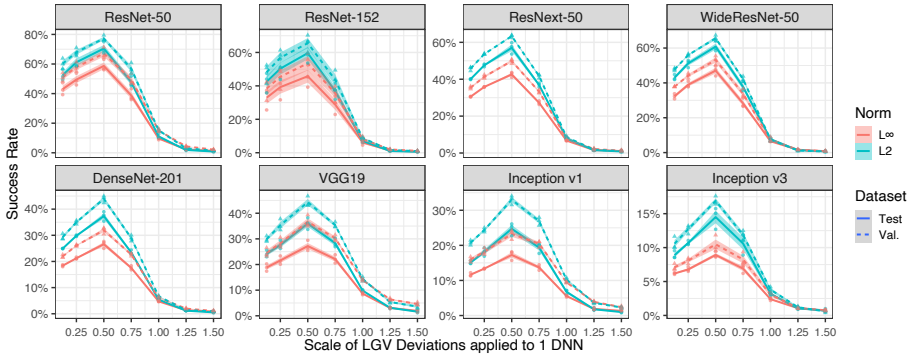

Fig. 20: Transfer success rate with respect to the γ hyperparameter, the scale of the LGV' deviations applied to an independent DNN ("1 DNN + γ (LGV' - LGV-SWA')").

D Hyperparameters

This section reports the success rates of the I-FGSM transfer attack for the LGV and I-FGSM attack hyperparameters: the LGV learning rate, the number of LGV epochs, the number of collected LGV weights per epoch, and the number of I-FGSM attack iterations. We select all hyperparameters by cross-validation. A random subset of 2000 examples from the ImageNet train set is used as validation set to craft adversarial examples. The selected hyperparameter value is unique and does not depend on the target to respect the black-box threat model where the architecture is unknown.

Each figure includes the eight studied targets (*subfigure title*), both L_∞ (*red*) and L_2 (*blue*) attacks, and the adversarial examples from the validation set (*dashed*) for hyperparameter selection and from the test set (*plain*) for independent evaluation.

D.1 Sensibility to the Learning Rate

We study the sensitivity of LGV on the constant learning rate value. LGV provides reliable transferability improvements for a wide range of learning rate, between 0.01 and 0.1 (Figure 21). The effectiveness of LGV degrades quickly as the learning rate goes larger than 1×10^{-1} or smaller than 5×10^{-3}. We suppose that small learning rates produce surrogates with gradients that overfit the initial model.

We describe the type of high learning rates suitable for LGV[10]. We can identify several kinds of high learning: (a) the highest possible learning rate that does not make the model leave the current local minimum; (b) the highest possible learning rate that makes the model jump between different local minima but does not cause deterministic chaos; (c) the highest possible learning rate that causes deterministic chaos but does not lead to numerical divergence. "High" in our case refers to the definition (b). With a learning rate of 0.05, LGV exits the initial local minimum, as indicated by the spike of the training loss during the first LGV epochs from 0.95 of the initial DNN to 3.1. This creates a drop of 5.31 percentage points in natural test accuracy between the initial DNN and the ensemble of 40 LGV models (Figure 21). Our learning rate allows SGD to explore a larger vicinity in the weight space. This leaves (mostly) definition (a) out. Numerical divergence appears for a learning rate of 50, which is three orders of magnitude above the optimal one. Transferability drops suddenly when deterministic chaos appears (from 95% to 9% along with the natural test accuracy from 67% to 33% when changing the learning rate from 0.1 to 1). Deterministic chaos is more dangerous to LGV than exploring without leaving the local minimum. Very low LGV success rates might be an indication of convergence to local a maximum due to an excessively high learning rate. These observations exclude definition (c), leaving definition (b) coherent with our results.

[10] We would like to thank the reviewers for raising this interesting discussion.

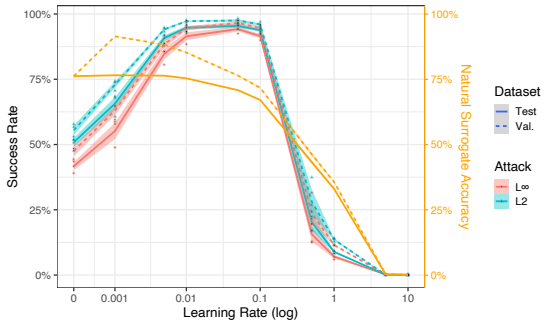

Fig. 21: Transfer success rate against the ResNet-50 target (*red, blue*) and natural test accuracy (*orange*) of the LGV surrogate trained with a wide range of constant learning rate, in pseudo-log scale. The null learning rate refers to the initial DNN.

We select a learning rate equal to 0.05, half of the learning rate at the beginning of training, based on a validation set, both attack norms, and the eight target models. These observations are valid for the three other target architectures of the ResNet family (Figure 22). However, the best learning rate against Inception v1 and v3 targets is 0.1 for both norms. This tolerance to a higher learning rate is coherent with our observation in Section 4.2 that transferability to these targets is less sensitive to the locally meaningful directions in the subspace spanned by LGV weights.

D.2 Number of LGV epochs

In the paper, LGV performs 10 additional epochs on the training set, which reach convergence (Figure 23). The computational cost of LGV is low, as it represents less than 7.7% of the training of the initial DNN. If the attacker has limited computational capability, five epochs are enough to obtain close results for most targets.

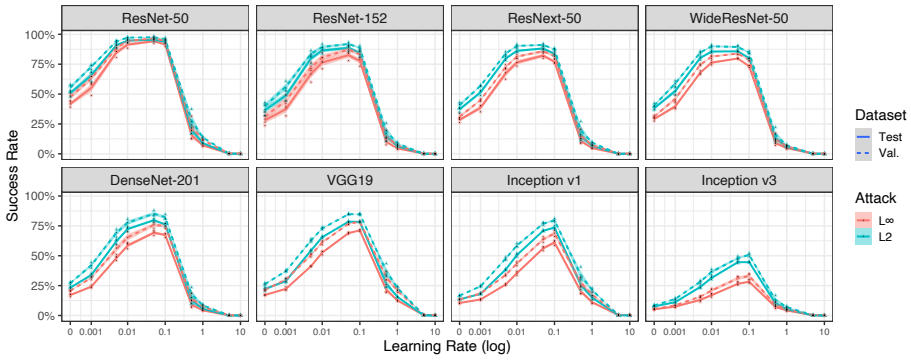

Fig. 22: Transfer success rate with respect to the LGV learning rate, for the eight targets.

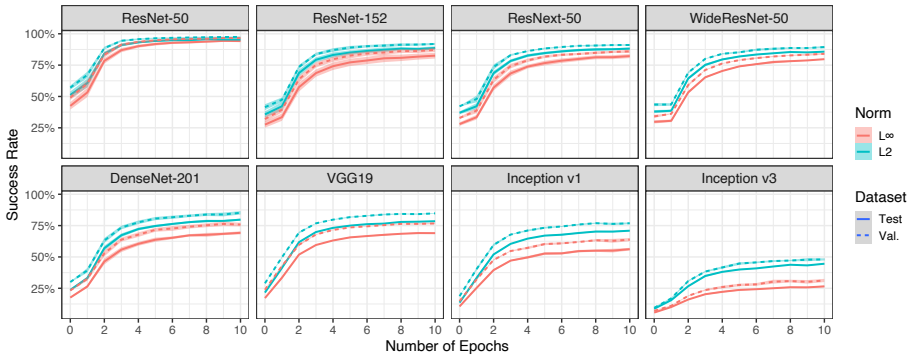

Fig. 23: Transfer success rate with respect to the number of LGV epochs.

D.3 Number of LGV weights per epoch

In the paper, LGV save four weights per epoch. As developed in Appendix B.2 a heavily restricted threat model where the memory is limited to a single model, can leverage the LGV-SWA surrogate. A threat model with intermediary limitation memory-wise could sample two LGV weights per epoch with minor success rate loss.

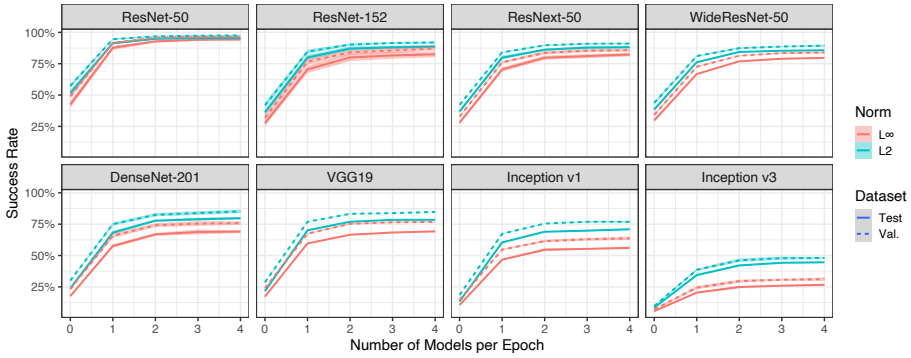

Fig. 24: Transfer success rate with respect to the number of LGV weights saved per epoch.

D.4 Number of attack iterations

The number of I-FGSM iterations is set to 50 based on the validation success rate of both the initial DNN ("1 DNN") and the LGV surrogate. The attack on the initial DNN converges to its maximum around 50 iterations for all targets. The same is true for the LGV surrogate against the ResNet family targets, but not against the Inception v1 and v3 architectures, where the success rate is already decreasing. For fairness, we choose 50 iterations in favour of the 1 DNN surrogate.

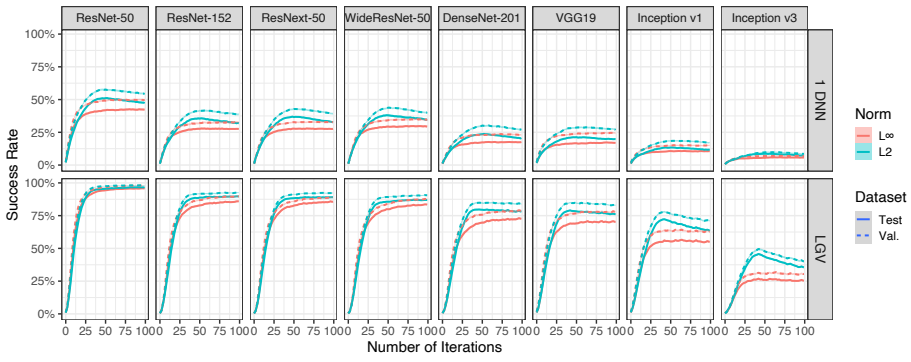

Fig. 25: Transfer success rate with respect to the number of iterations of the I-FGSM attack.