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

1 Introduction

Deep Neural Networks (DNNs) can effectively solve a board variety of computer vision tasks [4] but they are vulnerable to adversarial examples, i.e., misclassified examples that result from slight alterations to an original, well-classified example [2,24]. This phenomenon leads to real-world security flaws in various computer vision applications, including road sign classification [6], face recognition [23] and person detection [30].

Algorithms to produce adversarial examples – the *adversarial attacks* – typically work in white-box settings, that is, they assume full access to the target DNN and its weights. In practice, however, an attacker has limited knowledge of the target model. In these black-box settings, the attacker executes the adversarial attack on a *surrogate model* to produce adversarial examples that should *transfer to* (i.e., are also misclassified by) the target DNN.

Transferability is challenging to achieve consistently, though, and the factors behind transferability (or lack thereof) remain an active field of study [3,5,17,26,27,29]. This is because adversarial attacks seek the examples that

maximize the loss function of the surrogate model [8,15], whereas the target model has a different loss function. Methods to improve transferability typically rely on building diversity during optimisation [17,27,29]. While these approaches typically report significantly higher success rates than a classical surrogate, the relationships between the properties of the surrogate and transferability remain obscure. Understanding these relationships would enable the efficient construction of attacks (which would directly target the properties of interest) that effectively improve transferability.

In this paper, we propose Transferability from Geometric Vicinity (LGV), an efficient technique to increase the transferability of black-box adversarial attacks. LGV starts from a pretrained surrogate model and collects multiple weight samples from a few additional training epochs with a constant and high learning rate. Through extensive experiments, we show that LGV outperforms competing techniques by 3.1 to 59.9 percentage points of transfer rate.

We relate this improved transferability to two properties of the weights that LGV samples. First, LGV samples weights on a wider surface of the loss landscape in the weight space, leading to wider adversarial examples in the feature space. Our observations support our hypothesis that misalignment between surrogate and target alters transferability, which LGV avoids by sampling from wider optima. Second, the span of LGV weights forms a dense subspace whose geometry is intrinsically connected to transferability, even when the subspace is shifted to other local optima.

DNN geometry has been intensively studied from the lens of natural generalization [16,10,14,13,7,28]. However, the literature on the importance of geometry to improve transferability is scarcer [26,3] and has not yielded actionable insights that can drive the design of new transferability methods (more in Appendix A).

Our main contribution is, therefore, to shed new light on the importance of the surrogate loss geometry to explain the transferability of adversarial examples, and the development of the LGV method that improves over state-of-the-art transferability techniques.

2 Experimental Settings

Our study uses standard experimental settings to evaluate transfer-based black-box attacks. The surrogates are trained ResNet-50 models from [1]. The targets are eight trained models from PyTorch [21] with a variety of architectures – including ResNet-50. Therefore, we cover both the intra-architecture and inter-architecture cases. We craft adversarial examples from a random subset of 2000 ImageNet test images that all eight targets classify correctly. We compare LGV with four test-time transformations and their combinations, all applied on top of I-FGSM. We do not consider query-based black-box attacks because the threat model of transfer attacks does not grant oracle access to the target. To select the hyperparameters of the attacks, we do cross-validation on an independent subset of well-classified training examples. We provide results for L_∞ norm bounded perturbations (results for L_2 are in Appendix C.3). We report the average and

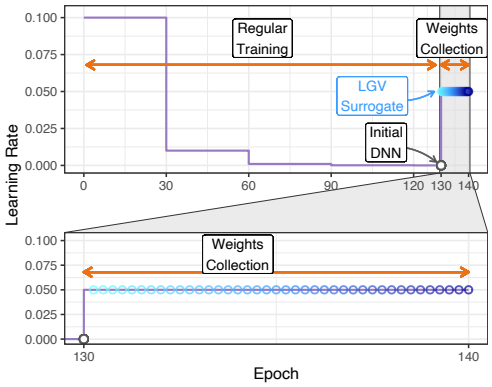


Fig. 1: Representation of the proposed approach.

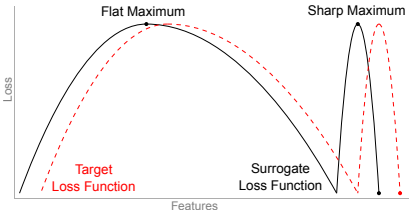


Fig. 2: Conceptual sketch of flat and sharp adversarial examples. Adapted from [14].

standard deviation of the attack success rate, i.e. the misclassification rate of untargeted adversarial examples, over 3 independent runs. Each run involves independent sets of examples, different surrogate models, and different random seeds. All code and models are available on GitHub[3]. More details are available in Appendix C.1.

Notation In the following, we denote (x, y) an example in $\mathcal{X} \times \mathcal{Y}$ with $\mathcal{X} \subset \mathbb{R}^d$, w a vector of p DNN weights in $\mathbb{R}^p$, and $\mathcal{L}(x; y, w)$ the loss function at input x of a DNN parametrised by w. The weights of a regularly trained DNN are noted w_0. Our LGV approach samples K weights $w_1, \cdots, w_K$. We name *LGV-SWA* the model parametrised by the empirical average of weights collected by LGV, i.e. $w_{\text{SWA}} = \frac{1}{K} \sum_{k=1}^{K} w_k$. The dot product between two vectors u, v is noted $\langle u, v \rangle$.

3 LGV: Transferability from Large Geometric Vicinity

Preliminaries. We aim to show the importance of the geometry of the surrogate loss in improving transferability. As a first step to motivate our approach, we experimentally demonstrate that adding random directions in the weight space to a regularly trained DNN increases its transferability, whereas random directions in the feature space applied on gradients do not. We build a surrogate called *RD* (see Table 1) by adding Gaussian white noise to a DNN with weight w_0:

$$\{w_0 + e_k \mid e_k \sim \mathcal{N}(\mathbf{0}, \sigma I_p),\ k \in [\![1, K]\!]\}\,. \tag{1}$$

This boils down to structuring the covariance matrix of the Gaussian noise added to input gradients from local variations in the weight space (at the first order approximation, see Appendix B.1). These preliminary experiments and their

[3] `https://github.com/Framartin/lgv-geometric-transferability`

results are detailed in Appendix C.2. These findings reveal that exploiting local variations in the weight space is a promising avenue to increase transferability. However, this success is sensitive to the length of the applied random vectors, and only a narrow range of σ values increase the success rate.

Based on these insights, we develop LGV (Transferability from Geometric Vicinity), our approach to efficiently build a surrogate from the vicinity of a regularly trained DNN. Despite its simplicity, it beats the combinations of four state of the art competitive techniques. The effectiveness of LGV confirms that the weight space of the surrogate is of first importance to increase transferability.

3.1 Algorithm

Our LGV approach performs in two steps: weight collection (Algorithm 1) and iterative attack (Algorithm 2).

First, LGV performs a few additional training epochs from a regularly trained model with weights w_0. LGV collects weights in a single run along the SGD trajectory at regular interval (4 per epoch). The *high constant learning rate* is key for LGV to sample in a sufficiently large vicinity. On the ResNet-50 surrogate we use in our experiments, we run SGD with half the learning rate at the start of the regular training (Figure 1). It allows SGD to escape the basin of attraction of the initial local minimum. Appendix D.1 includes an in-depth discussion on the type of high learning rates used by LGV. Compared to adding white noise to the weights, running SGD with a high constant learning rate changes the shape of the Gaussian covariance matrix to a non-trivial one [19]. As Table 1 shows, LGV improves over random directions (RD).

Second, LGV iteratively attacks the collected models (Algorithm 2). At each iteration k, the attack computes the gradient of one collected model with weights w_k randomly sampled without replacement. If the number of iterations is greater than the number of collected models, we cycle on the models. Because the attack computes the gradient of a single model at each iteration, this step has a negligible computational overhead compared to attacking a single model.

LGV offers multiple benefits. It is efficient (requires 5 to 10 additional training epochs from a pretrained model – see Appendix D.2), and it requires only minor modifications to training algorithms and adversarial attacks. In case memory is limited, we can approximate the collected set of LGV weights by their empirical average (see Appendix B.2). The most important hyperparameter is the learning rate. In Appendix D.1, we show that LGV provides reliable transferability improvements for a wide range of learning rate.

3.2 Comparison with the State of the Art

We evaluate the transferability of LGV and compare it with four state-of-the-art techniques.

MI [5] adds momentum to the attack gradients to stabilize them and escape from local maxima with poor transferability. Ghost Networks (**GN**) [17] use

Algorithm 1 LGV Weights Collection

Input: n_{epochs} number of epochs, K number of weights, η learning rate, γ momentum, w_0 pretrained weights, $\mathcal{D}$ training dataset
Output: $(w_1, \dots, w_K)$ LGV weights

$w \leftarrow w_0$ ▷ Start from a regularly trained DNN
for $i \leftarrow 1$ **to** K **do**
    $w \leftarrow \text{SGD}(w, \eta, \gamma, \mathcal{D}, \frac{n_{\text{epochs}}}{K})$ ▷ Perform $\frac{n_{\text{epochs}}}{K}$ of an epoch of SGD with η learning rate and γ momentum on $\mathcal{D}$
    $w_i \leftarrow w$
end for

Algorithm 2 I-FGSM Attack on LGV

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

Shuffle $(w_1, \dots, w_K)$ ▷ 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

dropout or skip connection erosion to efficiently generate diverse surrogate ensembles. **DI** [29] applies transformations to inputs to increase input diversity at each attack iteration. Skip Gradient Method (**SGM**) [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 discuss these techniques more deeply in Appendix A.

Table 1 reports the success rates of the ∞-norm attack (2-norm in Appendix C.3). We see that LGV alone improves over all (combinations of) other techniques (simple underline). Compared to individual techniques, LGV raises success rate by 10.1 to 59.9 percentage points, with an average of 35.6. When the techniques are combined, LGV still outperforms them by 1.8 to 55.4 percentage points, and 26.6 on average.

We also see that combining LGV with test-time techniques does not always improve the results and can even drastically decrease success rate. Still, LGV combined with input diversity (DI) and momentum (MI) generally outperforms LGV alone (by up to 20.5%) and ranks the best or close to the best. Indeed, both techniques tackle properties of transferability not covered by LGV: DI captures some input invariances learned by different architectures, and MI smooths the attack optimization updates in a moving average way.

The incompatibility of GN and SGM with LGV leads us to believe that their feature perturbations are cheap and bad proxies for local weight geometry. Eroding randomly skip connection, applying dropout on all layers, or backpropagating more linearly, may (poorly) approximate sampling in the weight space vicinity. LGV does this sampling explicitly.

Overall, our observations lessen both the importance of skip connections to explain transferability claimed by [27], and what was believed to hurt most

Table 1: Success rates of baselines, state-of-the-art and LGV under the L∞ 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	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$_{\pm 1.0}$
MI	53.0$_{\pm 2.2}$	36.3$_{\pm 1.5}$	34.7$_{\pm 0.4}$	38.1$_{\pm 2.0}$	22.0$_{\pm 0.1}$	21.1$_{\pm 0.3}$	13.9$_{\pm 0.4}$	7.3$_{\pm 0.8}$
GN	63.9$_{\pm 2.4}$	43.8$_{\pm 2.4}$	43.3$_{\pm 1.3}$	47.4$_{\pm 0.9}$	24.8$_{\pm 0.3}$	24.1$_{\pm 1.0}$	14.6$_{\pm 0.3}$	6.8$_{\pm 1.2}$
GN+MI	68.4$_{\pm 2.3}$	49.3$_{\pm 2.5}$	47.9$_{\pm 1.2}$	52.1$_{\pm 1.7}$	28.4$_{\pm 0.8}$	28.0$_{\pm 0.7}$	17.5$_{\pm 0.5}$	8.7$_{\pm 0.5}$
DI	75.0$_{\pm 0.2}$	56.4$_{\pm 1.9}$	59.6$_{\pm 1.5}$	61.6$_{\pm 2.4}$	41.6$_{\pm 1.1}$	39.7$_{\pm 0.9}$	27.7$_{\pm 1.0}$	15.2$_{\pm 1.0}$
DI+MI	81.2$_{\pm 0.3}$	63.8$_{\pm 1.9}$	67.6$_{\pm 0.9}$	68.9$_{\pm 1.5}$	49.3$_{\pm 0.7}$	46.7$_{\pm 0.4}$	33.0$_{\pm 1.0}$	19.4$_{\pm 0.9}$
SGM	64.4$_{\pm 0.8}$	49.1$_{\pm 3.1}$	48.9$_{\pm 0.6}$	51.7$_{\pm 2.8}$	30.7$_{\pm 0.9}$	33.6$_{\pm 1.3}$	22.5$_{\pm 1.5}$	10.7$_{\pm 0.9}$
SGM+MI	66.0$_{\pm 0.6}$	51.3$_{\pm 3.5}$	50.9$_{\pm 0.9}$	54.3$_{\pm 2.3}$	32.5$_{\pm 1.3}$	35.8$_{\pm 0.7}$	24.1$_{\pm 1.0}$	12.1$_{\pm 1.2}$
SGM+DI	76.8$_{\pm 0.5}$	62.3$_{\pm 2.7}$	63.6$_{\pm 1.7}$	65.3$_{\pm 1.4}$	45.5$_{\pm 0.9}$	49.9$_{\pm 0.8}$	36.0$_{\pm 0.7}$	19.2$_{\pm 1.7}$
SGM+DI+MI	80.9$_{\pm 0.7}$	66.9$_{\pm 2.5}$	68.7$_{\pm 1.2}$	70.0$_{\pm 1.7}$	50.9$_{\pm 0.6}$	56.0$_{\pm 1.4}$	42.1$_{\pm 1.4}$	23.6$_{\pm 1.6}$
Our techniques								
RD	60.6$_{\pm 1.5}$	40.5$_{\pm 3.0}$	39.9$_{\pm 0.2}$	44.4$_{\pm 3.2}$	22.9$_{\pm 0.8}$	22.7$_{\pm 0.5}$	13.9$_{\pm 0.2}$	6.6$_{\pm 0.7}$
LGV-SWA	84.9$_{\pm 1.2}$	63.9$_{\pm 3.7}$	62.1$_{\pm 0.4}$	61.1$_{\pm 2.9}$	44.2$_{\pm 0.4}$	42.4$_{\pm 1.3}$	31.5$_{\pm 0.8}$	12.2$_{\pm 0.8}$
LGV-SWA+RD	90.2$_{\pm 0.5}$	71.7$_{\pm 3.4}$	69.9$_{\pm 1.2}$	69.1$_{\pm 3.3}$	49.9$_{\pm 1.0}$	47.4$_{\pm 2.0}$	34.9$_{\pm 0.3}$	13.5$_{\pm 0.9}$
LGV (ours)	95.4$_{\pm 0.1}$	85.5$_{\pm 2.3}$	83.7$_{\pm 1.2}$	82.1$_{\pm 2.4}$	69.3$_{\pm 1.0}$	67.8$_{\pm 1.2}$	58.1$_{\pm 0.8}$	25.3$_{\pm 1.9}$
LGV combined with other techniques								
MI	97.1$_{\pm 0.3}$	88.7$_{\pm 2.3}$	87.0$_{\pm 1.0}$	86.6$_{\pm 2.1}$	73.2$_{\pm 1.4}$	71.6$_{\pm 1.4}$	60.7$_{\pm 0.6}$	27.4$_{\pm 0.8}$
GN	94.2$_{\pm 0.2}$	83.0$_{\pm 2.2}$	80.8$_{\pm 0.7}$	79.5$_{\pm 2.4}$	66.9$_{\pm 0.7}$	66.6$_{\pm 0.7}$	56.2$_{\pm 0.5}$	24.4$_{\pm 1.4}$
GN+MI	96.4$_{\pm 0.1}$	87.2$_{\pm 2.0}$	85.3$_{\pm 0.8}$	84.4$_{\pm 2.3}$	70.4$_{\pm 1.0}$	71.2$_{\pm 0.8}$	59.2$_{\pm 0.5}$	26.5$_{\pm 0.4}$
DI	93.8$_{\pm 0.1}$	84.4$_{\pm 1.6}$	84.1$_{\pm 0.6}$	81.8$_{\pm 1.6}$	74.9$_{\pm 0.2}$	76.2$_{\pm 0.7}$	71.5$_{\pm 1.3}$	38.9$_{\pm 1.1}$
DI+MI	96.9$_{\pm 0.0}$	89.6$_{\pm 1.7}$	89.6$_{\pm 0.4}$	88.4$_{\pm 1.1}$	82.3$_{\pm 0.9}$	82.2$_{\pm 0.9}$	78.6$_{\pm 0.8}$	45.4$_{\pm 0.5}$
SGM	86.9$_{\pm 0.7}$	74.8$_{\pm 2.6}$	73.5$_{\pm 1.2}$	72.8$_{\pm 2.4}$	60.6$_{\pm 0.9}$	69.0$_{\pm 1.8}$	61.5$_{\pm 1.7}$	31.7$_{\pm 1.8}$
SGM+MI	89.1$_{\pm 0.5}$	77.1$_{\pm 2.8}$	76.7$_{\pm 1.1}$	75.6$_{\pm 2.1}$	62.7$_{\pm 1.1}$	72.3$_{\pm 1.0}$	64.7$_{\pm 2.2}$	34.2$_{\pm 1.7}$
SGM+DI	84.3$_{\pm 0.6}$	72.5$_{\pm 2.4}$	72.8$_{\pm 0.7}$	70.7$_{\pm 1.8}$	62.1$_{\pm 0.9}$	71.8$_{\pm 1.4}$	67.0$_{\pm 1.8}$	37.7$_{\pm 1.8}$
SGM+DI+MI	87.7$_{\pm 0.6}$	76.4$_{\pm 2.5}$	77.2$_{\pm 0.8}$	75.6$_{\pm 1.1}$	66.4$_{\pm 1.0}$	76.6$_{\pm 0.7}$	72.1$_{\pm 1.4}$	42.9$_{\pm 1.7}$

transferability, i.e., the optimization algorithm [5] and lack of input diversity [29]. Our results demonstrate that the diversity of surrogate models (one model per iteration) is at most importance to avoid adversarial examples overfitting to their surrogate model. LGV does so more effectively than [17].

We show that LGV consistently increases transfer-based attacks success. However, it is not trivial why sampling surrogate weights in the vicinity of a local minimum helps adversarial examples to be successful against a model from another local minimum. In the following, we analyse the LGV success with a geometrical perspective.

4 Investigating LGV Properties: On the Importance of the Loss Geometry

In the following, we relate the increased transferability of LGV to two geometrical properties of the weight space. First, we show that LGV collects weights on flatter regions of the loss landscape than where it started (the initial, pre-trained surrogate). These flatter surrogates produce wider adversarial examples in feature space, and improve transferability in case of misalignment between the surrogate loss (optimized function) and the target loss (objective function). Second, the span of LGV weights forms a dense subspace whose geometry is intrinsically connected to transferability, even when the subspace is shifted to other independent solutions. The geometry plays a different role depending on the functional similarity between the target and the surrogate architectures.

4.1 Loss Flatness: the Surrogate-Target Misalignment Hypothesis

We first explain why LGV is a good surrogate through the *surrogate-target misalignment hypothesis*. We show that LGV samples from flatter regions in the weight space and, as a result, produces adversarial examples flatter in the feature space. This leads to surrogates that are more robust to misalignment between the surrogate and target prediction functions.

Sharp and flat minima have been discussed extensively in machine learning (see Appendix A). A sharp minimum is one where the variations of the objective function in a neighbourhood are important, whereas a flat minimum shows low variations [11]. Multiple studies [13,14] correlate (natural) generalization with the width of the solution in the weight space: if the train loss is shifted w.r.t. the test loss in the weight space, wide optima are desirable to keep the difference between train and test losses small.

We conjecture that a similar misalignment occurs between the surrogate model and the target model *in the feature space*. See Figure 2 for an illustration of the phenomenon. Under this hypothesis, adversarial examples at wider maxima of the surrogate loss would transfer better than sharp ones. The assumption that surrogate and target models are shifted with respect to each other seems particularly reasonable when both are the same function parametrised differently (intra-architecture transferability), or are functionally similar (same architecture

family). We do not expect all types of loss flatness to increase transferability, since entirely vanished gradients would be the flatter loss surface possible and annihilate gradient-based attacks.

Table 2: Sharpness metrics in the weight space, i.e., the largest eigenvalue and the rank of the Hessian, computed on three types of surrogate and 10,000 training examples.

	Hessian	
Model	Max EV	Trace
1 DNN	558 ± 57	16258 ± 725
LGV indiv.	168 ± 127	4295 ± 517
LGV-SWA	30 ± 1	1837 ± 70

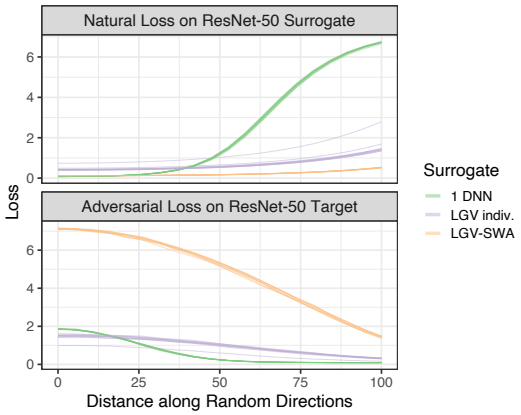


Fig. 3: L_∞ attack crafted on surrogate with natural loss (*up*), evaluated on target (*down*) with respect to the 2-norm distance along 10 random directions in the weight space from the LGV-SWA solution (*orange*), random LGV weights (*purple*), and the initial DNN (*green*).

We provide two empirical evidences for this hypothesis. First, LGV flattens weights compared to the initial DNN. Second, LGV similarly flattens adversarial examples in the feature space.

Flatness in the Weight Space We establish that LGV weights and their mean (LGV-SWA) are in a flatter region of the loss than the initial DNN. The reason we consider LGV-SWA is that this model lies at the center of the loss surface explored by LGV and attacking this model yields a good first-order approximation of attacking the ensemble of LGV weights (cf. Appendix B.2). First, we compute Hessian-based sharpness metrics. Second, we study the variations of the loss in the weight space along random directions from the solutions.

First, Table 2 reports two sharpness metrics in the weight space: the largest eigenvalue of the Hessian which estimates the sharpness of the sharpest direction, and the trace of the Hessian which estimates the sharpness of all directions. Both metrics conclude that the initial DNN is significantly sharper than the LGV and LGV-SWA weights.

Second, like [13], we sample a random direction vector d on the unit sphere, $d = \frac{e}{\|e\|_2}$ with $e \sim \mathcal{N}(\mathbf{0}, I_p)$ and we study the following rays,

$$w_0(\alpha, d) = w_0 + \alpha d, \quad w_k(\alpha, d) = w_k + \alpha d, \quad w_{\text{SWA}}(\alpha, d) = w_{\text{SWA}} + \alpha d, \quad (2)$$

with $\alpha \in \mathbb{R}^+$. That is, we follow the same direction d for the three studied solutions. Figure 3 reports the intra-architecture results for 10 random directions (see Appendix C.4 for other settings). The natural loss in the weight space is wider at the individual LGV weights and at LGV-SWA than it is at the initial model weights (upper plot). When adding the random vector αd, the natural loss of LGV-SWA barely increases, while that of the initial model w_0 reaches high values: 0.40 vs. 6.67 for $\|\alpha \cdot d\|_2$ from 0 to 100. The individual LGV models are in between, with an 1.12 increase on average. As Figure 3 also reveals, the increased flatness of LGV-SWA in the weight space comes with an increased transferability. We investigate this phenomenon deeper in what follows.

Flatness in the Feature Space Knowing that LGV (approximated via LGV-SWA) yields loss flatness in the weight space, we now connect this observation to the width of basins of attractions in the feature space when we craft adversarial examples. That is, we aim to show that flat surrogates in the weight space produce flatter adversarial examples in the feature space.

To study flatness of adversarial examples in the feature space, we consider the plane containing 3 points: the original example x, a LGV adversarial example $x^{\text{adv}}_{\text{LGV}}$, and an adversarial example crafted against the initial DNN $x^{\text{adv}}_{\text{DNN}}$. We build an orthonormal basis $(u', v') \coloneqq (\frac{u}{\|u\|}, \frac{v}{\|v\|})$ using the first two steps of the Gram–Schmidt process,

$$(u, v) = \left(x^{\text{adv}}_{\text{LGV}} - x,\ (x^{\text{adv}}_{\text{DNN}} - x) - \frac{\langle x^{\text{adv}}_{\text{DNN}} - x, u \rangle}{\langle u, u \rangle} u \right). \quad (3)$$

We focus our analysis on the 2-norm attack. It constrains adversarial perturbations inside the L_2-ball centred on x of radius ε. This has the convenient property that the intersection of this ball with our previously defined plane (containing x) is a disk of radius ε.

Figure 4 shows the loss of the ensemble of LGV weights and the loss of the initial DNN in the (u', v') coordinate system. We report the average losses over 500 disks, each one centred on a randomly picked test example. It appears that LGV has a much smoother loss surface than its initial model. LGV adversarial examples are in a wide region of the LGV ensemble's loss. The maxima of the initial DNN loss is highly sharp and much more attractive for gradient ascent than the ones found by LGV – the reason why adversarial examples crafted from the initial DNN overfit.

Flatness and Transferability Figure 4 also shows the losses of two target models in the (u', v') coordinate system. The LGV loss appears particularly well aligned with the one of the ResNet-50 target (intra-architecture transferability).

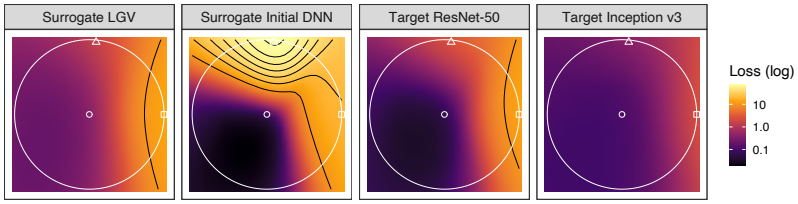


Fig. 4: Surrogate (*left*) and target (*right*) losses in the plane containing the original example (*circle*), an adversarial example against LGV (*square*) and one against the initial DNN (*triangle*), in the (u', v') coordinate system. Colours are in log-scale, contours in natural scale. The white circle represents the intersection of the 2-norm ball with the plane.

We observe a *shift between the contour of both models, with the same functional form*. These observations are valid for other targets and on planes defined by adversarial examples of other surrogates (see Appendix C.6). All these observations corroborate our surrogate-target misalignment hypothesis.

In Appendix C.5, we provide results of another experiment that corroborates our findings. We interpolate the weights between LGV-SWA and the initial model, i.e. moving along a non-random direction, and confirm that (i) the surrogate loss is flatter at LGV-SWA than at the initial model weights, (ii) that the adversarial loss of target models gets higher as we move from the initial model to LGV-SWA.

Section 4.1 – Conclusion. LGV weights lie in flatter regions of the loss landscape than the initial DNN weights. Flatness in the weight space correlates with flatness in the feature space: LGV adversarial examples are wider maxima than sharp adversarial examples crafted against the initial DNN. These conclusions support our surrogate-target misalignment hypothesis: if surrogate and target losses are shifted with respect to each other, a wide optimum is more robust to this shift than a sharp optimum.

4.2 On the Importance of LGV Weight Subspace Geometry

Although we have demonstrated the link between the better transferability that LGV-SWA (and in extenso, the LGV ensemble) achieves and the flatness of this surrogate's loss, additional experiments have revealed that the LGV models – taken individually – achieve lower transferability, although they also have a flatter loss than the initial model (see Appendix C.7 for details). This indicates that other factors are in play to explain LGV transferability.

In what follows, we show the importance of the geometry of the subspace formed by LGV models in increasing transferability. More precisely, deviations of LGV weights from their average spans a weight subspace which is (i) densely

related to transferability (i.e., *it is useful*), (ii) composed of directions whose relative importance depends on the functional similarity between surrogate and target (i.e., *its geometry is relevant*), (iii) remains useful when shifted to other solutions (i.e., *its geometry captures generic properties*). Similarly to [12], the K-dimensional subspace of interest is defined as,

$$\mathcal{S} = \{w \,|\, w = w_{\text{SWA}} + \mathbf{P}z\}\,, \tag{4}$$

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

A Subspace Useful for Transferability First, we show that the subspace has importance for transferability. Similarly to our previous RD surrogate, we build a new surrogate "LGV-SWA + RD" by sampling random directions in the full weight space around LGV-SWA. It is defined as:

$$\{w_{\text{SWA}} + e'_k \mid e'_k \sim \mathcal{N}(\mathbf{0},\, \sigma' I_p),\, k \in [\![1, K]\!]\}\,, \tag{5}$$

where the standard deviation σ' is selected by cross-validation in Appendix C.8.

Table 1 reports the transferability of this surrogate for the L_∞ attack (see Appendix C.3 for L_2). We observe that random deviations drawn in the entire weight space do improve the transferability of LGV-SWA (increase of 1.32 to 10.18 percentage points, with an average of 6.90). However, the LGV surrogate systematically outperforms "LGV-SWA + RD". The differences range from 4.33 to 29.15 percentage points, and average to 16.10. Therefore, the subspace $\mathcal{S}$ has specific geometric properties related to transferability that make this ensemble outperforms the ensemble formed by random directions around LGV-SWA.

In Appendix C.9, we also show that the subspace is densely connected to transferability by evaluating the transferability of surrogates built from $\mathcal{S}$ by sampling $z \sim \mathcal{N}(\mathbf{0},\, I_K)$.

Decomposition of the LGV Projection Matrix Second, we analyse the contribution of subspace basis vectors to transferability through a decomposition of their projection matrix. Doing so, we build alternative LGV surrogates with an increasingly reduced dimensionality, and we assess the impact of this reduction on transferability.

We decompose the matrix of LGV weights deviations $\mathbf{P}$ into orthogonal directions, using principal component analysis (PCA) since the PCA coordinate transformation diagonalises this matrix. Following [12], we apply PCA based on exact full SVD[4] to obtain a new orthonormal basis of the LGV weight subspace. We exploit the orthogonality of the components to change the basis of each w_k

[4] As [12] we use the PCA implementation of sklearn[22], but here we select the full SVD solver instead of randomized SVD to keep all the singular vectors.

with the PCA linear transformation and project onto the first C principal components. We then apply the inverse map, with w_{SWA} as shift vector, to obtain a new weight vector $w_{k,C}^{\text{proj}}$. We repeat the process with different value of C, which enables us to control the amount of explained weights variance and to build LGV ensembles with a reduced dimensionality.

The eigenvalues of the LGV weights deviation matrix equal the variance of the weights along the corresponding eigenvectors. We use the ratio of explained weights variance to measure the relative loss of information that would result from removing a given direction. From an information theory perspective, if a direction in the weight space is informative of transferability, we expect the success rate to decrease with the loss of information due to dimensionality reduction. Note that the surrogate projected on the PCA zero space (i.e. $C = 0$) is LGV-SWA, whereas $C = K$ means we consider the full surrogate ensemble.

Figure 5 shows, for each dimensionality reduced LGV surrogates, the explained variance ratio of its lower dimensional weight subspace and the success rate that this ensemble achieves on the ResNet-50 and Inception v3 targets. To observe the trends, we add the hypothetical cases of proportionality to the variance (solid line) and equal contributions of all dimensions (dashed line).

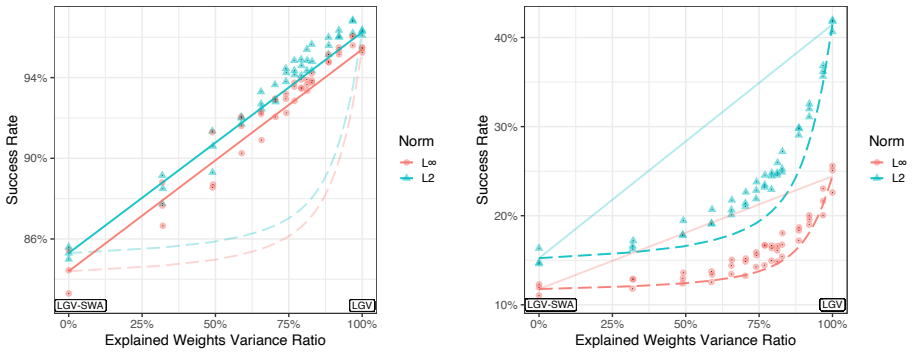
For both targets, explained variance correlates positively with transferability. This means that our approach improves transferability more, as it samples along directions (from SWA) with higher variance. Especially in the intra-architecture case (Figure 5a), there is an almost-linear correlation between the importance of a direction in the weight space and its contribution to transferability. This conclusion can be loosely extended to the targets that belong to the same architecture family as the surrogate, i.e. ResNet-like models (Appendix C.10).

In some inter-architecture cases, we do not observe this linear trend, although the correlation remains positive. In Figure 5b, we see that the real variance ratio/transfer rate curve is close to the hypothetical case where each direction would equally improve transferability on the Inception v3 target. This means that, in this inter-architecture case, each direction contributes almost-equally to transferability regardless of their contribution to the subspace variance. In supplementary materials, we show other inter-architecture cases (e.g., DenseNet-201 and VGG19) that are intermediate between linear correlation and almost-equal dimensional contributions (Appendix C.10).

Taking together the above results, we explain the better transferability of LGV with the variance of the subspace it forms. However, this correlation is stronger as the surrogate and target architectures are more functionally similar.

Shift of LGV Subspace to Other Local Minima Third, we demonstrate that the benefits of the LGV subspace geometry are shared across solutions in the weight space. This indicates that there are generic geometry properties that relate to transferability.

We apply LGV to another independently trained DNN w'_0. We collect K new weights w'_k, which we average to obtain w'_{SWA}. We construct a new surrogate by adding the new deviations to w_{SWA},



(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.

$$\{w_{\text{SWA}} + (w'_k - w'_{\text{SWA}}) \mid k \in [\![1, K]\!]\}\,, \tag{6}$$

and we call this new shifted surrogate "LGV-SWA + (LGV' - LGV-SWA')".

Shifting a LGV subspace to another flat solution (i.e., another LGV-SWA) yields a significantly better surrogate than sampling random directions from this solution. The difference between "LGV-SWA + (LGV' - LGV-SWA')" and "LGV-SWA + RD" varies from 3.27 to 12.32 percentage points, with a mean of 8.61 (see Appendix C.11 for detailed results). The fact that the subspace still improves transferability (compared to a random subspace) when applied to another vicinity reveals that subspace geometry has generic properties related to transferability.

Yet, we also find a degradation of success rate between this translated surrogate and our original LGV surrogate (-7.49 percentage points on average, with values between -1.02 and -16.80). It indicates that, though the geometric properties are shared across vicinities, the subspace is optimal (w.r.t. transferability) when applied onto its original solution.

The subspace is not solely relevant for solutions found by LGV: LGV deviations are also relevant when applied to regularly trained DNNs. For that, we build a new surrogate "1 DNN + γ (LGV' - LGV-SWA')" centred on the DNN w_0,

$$\{w_0 + \gamma(w'_k - w'_{\text{SWA}}) \mid k \in [\![1, K]\!]\}\,, \tag{7}$$

where the LGV deviations are scaled by a factor $\gamma \in \mathbb{R}$. Scaling is essential here because DNNs are sharper than LGV-SWA. Unscaled LGV deviations exit the

vicinity of low loss, which drops the success rate by 32.8 percentage points on average compared to the optimal γ value of 0.5 (see Appendix C.11 for detailed results). When properly scaled and applied to an independently and regularly trained DNN, LGV deviations improve upon random directions by 10.0 percentage points in average (2.87—13.88).

With all these results, we exhibit generic properties of the LGV subspace. It benefits solutions independently obtained. Applying LGV deviations on a solution of a different nature may require to scale them according to the new local flatness.

Section 4.2 – Conclusion. Taking together all our results, we conclude that the improved transferability of LGV comes from the geometry of the subspace formed by LGV weights in a flatter region of the loss. The LGV deviations spans a weight subspace whose geometry is densely and generically relevant for transferability. This subspace is key, as a single flat LGV model is not enough to succeed. This entire subspace enables to benefit from the flatness of this region, overcoming potential misalignment between the loss functions of the surrogate and that of the target model. That is, it increases the probability that adversarial examples maximizing the surrogate loss will also (near-)maximize the target loss – and thus successfully transfer.

5 Conclusion and Future Work

We show that random directions in the weight space sampled at each attack iteration increase transferability, unlike random directions in feature space. Based on this insight, we propose LGV, our approach to build a surrogate by collecting weights along the SGD trajectory with a high constant learning rate, starting from a regularly trained DNN. LGV alone beats all combinations of four state-of-the-art techniques. We analyse LGV extensively to conclude that (i) flatness in the weight space produces flatter adversarial examples which are more robust to surrogate-target misalignment; (ii) LGV weights spans a dense subspace whose geometry is intrinsically connected to transferability. Overall, we open new directions to understand and improve transferability from the geometry of the loss in the weight space. Future work may, based on the insights of [32] on natural generalization, study transferability with the perspective of volume in the weight space that leads to similar predictive function.

Acknowledgements

This work is supported by the Luxembourg National Research Funds (FNR) through CORE project C18/IS/12669767/STELLAR/LeTraon.

References

1. Ashukha, A., Lyzhov, A., Molchanov, D., Vetrov, D.: Pitfalls of In-Domain Uncertainty Estimation and Ensembling in Deep Learning (2 2020), http://arxiv.org/abs/2002.06470
2. Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., Roli, F.: Evasion attacks against machine learning at test time. In: Lecture Notes in Computer Science. vol. 8190 LNAI, pp. 387–402 (8 2013). https://doi.org/10.1007/978-3-642-40994-3_25
3. Charles, Z., Rosenberg, H., Papailiopoulos, D.: A geometric perspective on the transferability of adversarial directions. In: AISTATS 2019 (11 2020), http://arxiv.org/abs/1811.03531
4. Dargan, S., Kumar, M., Ayyagari, M.R., Kumar, G.: A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning. Archives of Computational Methods in Engineering **27**(4), 1071–1092 (9 2019)
5. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting Adversarial Attacks with Momentum. In: CVPR. pp. 9185–9193 (10 2018). https://doi.org/10.1109/CVPR.2018.00957
6. Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., Song, D.: Robust Physical-World Attacks on Deep Learning Models (7 2017). https://doi.org/10.48550/arxiv.1707.08945
7. Foret, P., Kleiner Google Research, A., Mobahi Google Research, H., Neyshabur Blueshift, B.: Sharpness-Aware Minimization for Efficiently Improving Generalization (10 2020), https://arxiv.org/abs/2010.01412v3
8. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples (12 2014)
9. Gubri, M., Cordy, M., Papadakis, M., Traon, Y.L.: Efficient and Transferable Adversarial Examples from Bayesian Neural Networks. UAI 2022 (2022), http://arxiv.org/abs/2011.05074
10. Gur-Ari, G., Roberts, D.A., Dyer, E.: Gradient Descent Happens in a Tiny Subspace (12 2018), http://arxiv.org/abs/1812.04754
11. Hochreiter, S., Schmidhuber, J.: Flat Minima. Neural Computation **9**(1), 1–42 (1 1997). https://doi.org/10.1162/NECO.1997.9.1.1
12. Izmailov, P., Maddox, W.J., Kirichenko, P., Garipov, T., Vetrov, D., Wilson, A.G.: Subspace Inference for Bayesian Deep Learning. UAI 2019 (7 2019), http://arxiv.org/abs/1907.07504
13. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., Wilson, A.G.: Averaging weights leads to wider optima and better generalization. In: 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018. vol. 2, pp. 876–885. Association For Uncertainty in Artificial Intelligence (AUAI) (3 2018), http://arxiv.org/abs/1803.05407
14. Keskar, N.S., Nocedal, J., Tang, P.T.P., Mudigere, D., Smelyanskiy, M.: On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. ICLR 2017 (9 2016), https://arxiv.org/abs/1609.04836v2
15. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. In: 5th International Conference on Learning Representations, ICLR 2017 - Workshop Track Proceedings (7 2017), http://arxiv.org/abs/1607.02533
16. Li, C., Farkhoor, H., Liu, R., Yosinski, J.: Measuring the Intrinsic Dimension of Objective Landscapes. 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings (4 2018), https://arxiv.org/abs/1804.08838v1

17. Li, Y., Bai, S., Zhou, Y., Xie, C., Zhang, Z., Yuille, A.: Learning Transferable Adversarial Examples via Ghost Networks. AAAI **34**(07), 11458–11465 (12 2018). https://doi.org/10.1609/aaai.v34i07.6810, http://arxiv.org/abs/1812.03413
18. Maddox, W.J., Garipov, T., Izmailov, Vetrov, D., Wilson, A.G.: A simple baseline for Bayesian uncertainty in deep learning. In: NeurIPS. vol. 32. arXiv (2 2019), http://arxiv.org/abs/1902.02476
19. Mandt, S., Hof Fman, M.D., Blei, D.M.: Stochastic Gradient Descent as Approximate Bayesian Inference. Journal of Machine Learning Research **18**, 1–35 (4 2017), https://arxiv.org/abs/1704.04289v2
20. Papernot, N., McDaniel, P., Goodfellow, I.: Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples (5 2016), http://arxiv.org/abs/1605.07277
21. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: NIPS, pp. 8024–8035 (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
22. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in {P}ython. Journal of Machine Learning Research **12**, 2825–2830 (2011)
23. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: A General Framework for Adversarial Examples with Objectives. ACM Transactions on Privacy and Security **22**(3), 30 (12 2017). https://doi.org/10.1145/3317611, http://dx.doi.org/10.1145/3317611
24. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks (12 2013), http://arxiv.org/abs/1312.6199
25. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. In: 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings (5 2018), http://arxiv.org/abs/1705.07204
26. Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: The Space of Transferable Adversarial Examples (4 2017), http://arxiv.org/abs/1704.03453
27. Wu, D., Wang, Y., Xia, S.T., Bailey, J., Ma, X.: Skip Connections Matter: On the Transferability of Adversarial Examples Generated with ResNets. In: ICLR (2 2020), http://arxiv.org/abs/2002.05990
28. Wu, D., Xia, S.T., Wang, Y.: Adversarial weight perturbation helps robust generalization. In: Advances in Neural Information Processing Systems. Neural information processing systems foundation (4 2020), https://arxiv.org/abs/2004.05884v2
29. Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., Yuille, A.L.: Improving transferability of adversarial examples with input diversity. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. vol. 2019-June, pp. 2725–2734 (3 2019). https://doi.org/10.1109/CVPR.2019.00284
30. Xu, K., Zhang, G., Liu, S., Fan, Q., Sun, M., Chen, H., Chen, P.Y., Wang, Y., Lin, X.: Evading Real-Time Person Detectors by Adversarial T-shirt (10 2019), http://arxiv.org/abs/1910.11099

31. Yao, Z., Gholami, A., Keutzer, K., Mahoney, M.W.: PyHessian: Neural Networks Through the Lens of the Hessian. Big Data 2020 pp. 581–590 (12 2019). https://doi.org/10.1109/BigData50022.2020.9378171
32. Zhang, S., Reid, I., Pérez, G.V., Louis, A.: Why flatness does and does not correlate with generalization for deep neural networks (3 2021), http://arxiv.org/abs/2103.06219