Efficient and Transferable Adversarial Examples from Bayesian Neural Networks (Supplementary Material)

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

[1]University of Luxembourg, Luxembourg, LU
[2]University of California, Berkeley, CA, USA

In the supplementary materials for the paper, the following are provided:

- The detailed experimental setup section
- The description and experimental setup of Bayesian and Ensembling training techniques
- Additional results, including:
  - The natural accuracy of target Neural Networks;
  - The proportions of vanishing gradients of cSGLD surrogate compared to DNN surrogate;
  - The intra-architecture transfer success rates on cSGLD and Deep Ensemble of 1, 2, 5 and 15 DNNs surrogates;
  - The inter-architecture transfer success rates of single architecture surrogates;
  - The intra-architecture transfer success rate of six Bayesian and Ensemble training methods attacked by L2 I-FGSM;
  - The intra-architecture transfer success rate combined with test-time transformations on CIFAR-10;
  - The transfer rate of cSGLD with respect to the number of cycles and samples per cycle;
- An illustration of the cSGLD cyclical learning rate schedule;
- A diagram of the relationships between gradient-based attacks;
- The algorithm applied to perform approximate Bayesian model averaging efficiently;
- Details on hyperparameters, including:
  - The transfer success rate of iterative attacks with respect to the number of iterations;
  - The tuning of the hyperparameter of the Skip Gradient Method technique to extend it to PreResNet110;
  - The hyperparameters used to train and attack models.

A EXPERIMENTAL SETUP

Datasets. We consider ImageNet (ILSVRC2012; Russakovsky et al. 2015), CIFAR-10 [Krizhevsky, 2009] and MNIST. In all cases, we train the surrogate and target models on the entire training set. For each CIFAR-10 and MNIST target model, we select all the examples from the test set that are correctly predicted by it. In the case of ImageNet, we use a random subset of 5000 correctly predicted test images.

Architectures. We cover a diverse set of architectures in terms of heterogeneity (similar and different families of architecture), computation cost, and release date. For ImageNet, we select five architectures with $3 \times 224 \times 244$ input size. Three classical architectures: ResNet-50 He et al. [2016a][1], ResNeXt-50 32x4d Xie et al. [2017] and Densenet-121 Xie et al. [2017]; and two mobile architectures: MNASNet 1.0 Tan et al. [2018] and EfficientNet-B0 Tan and Le [2019]. Following the work of Ashukha et al. [2020], we consider the following five architectures for CIFAR-10: PreResNet110 He et al. [2016b],

[1]Ashukha et al. [2020] study ResNet-50 only on ImageNet. We used their shared trained models as surrogate DNNs.

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).

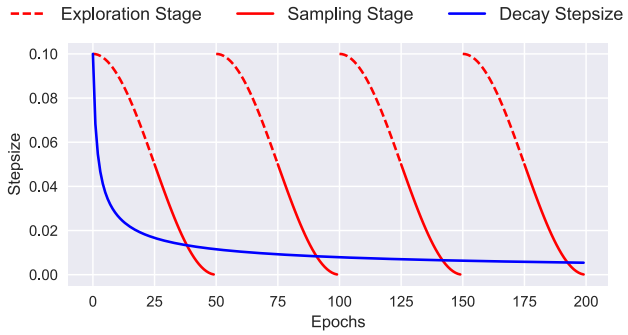


Figure 1: Illustration of the cSGLD cyclical learning rate schedule (red) and the traditional decreasing learning rate schedule (blue). Each cSGLD cycle is composed of an exploration phase (burn-in period of MCMC algorithms — red doted) and of a sampling phase (red plain). Figure taken from Zhang et al. (2020).

PreResNet164, VGG16BN, VGG19BN Simonyan and Zisserman [2015], and WideResNet28x10 Zagoruyko and Komodakis [2016]. We study three architectures on MNIST: "FC" a fully connected neural network with two hidden layers 1200-1200, "Small FC" with a single fully connected hidden layer of size 512, and "CNN" a convolutional neural network composed of two convolutional layers with 32 filters each followed by two fully connected hidden layers 200-200.

Target models. The target models are deterministic DNNs. For ImageNet, we use the pre-trained models provided by PyTorch Paszke et al. [2019] and the pre-trained EfficientNet-B0 provided by PyTorch Image Models (*timm*). In the case of CIFAR-10, they are trained using Adam optimizer for 300 epochs with step-wise learning rate decay that divides it by 10 every 75 epochs (MNIST: 50 epochs in total, learning rate divided by 10 every 20 epochs). The benign accuracy of all target models exceeds 73% (ImageNet), 83% (CIFAR-10) and 98% (MNIST); see Table 1 below exact values.

Table 1: Top-1 natural test accuracy of target DNNs.

Dataset	**Target DNN**	**Benign Test Accuracy**
CIFAR-10	PreResNet110	93.26 %
	PreResNet164	93.03 %
	VGG16bn	83.68 %
	VGG19bn	83.62 %
	WideResNet28x10	92.13 %
ImageNet	ResNet50	76.15 %
	ResNeXt50 32x4d	77.62 %
	Densenet121	74.65 %
	MNASNet 1.0	73.51 %
	EfficientNet-B0	77.70 %
MNIST	CNN	99.33 %
	FC	98.65 %
	Small FC	98.41 %

Surrogate models (Deep Ensemble). For CIFAR-10 and MNIST, the DNNs used to form surrogate ensembles are trained using the same process as the target models. Therefore, the comparison between deterministic DNNs and cSGLD is fair, since one can expect the deterministic DNNs surrogate to be "close" to the target. As for ImageNet, we retrieve an ensemble of 15 ResNet-50 models trained independently by Ashukha et al. [2020] using SGD with momentum during 130 epochs. For the RQ2 experiments, we train similarly one model for every 4 other ImageNet architectures.

Surrogate models (cSGLD). Following the work of Ashukha et al. [2020] and Zhang et al. [2020], we train models with cSGLD on CIFAR-10 for 6 learning rate cycles (which, as our RQ4 experiments reveal, is where the transfer rate starts plateauing). cSGLD performs 5 cycles on ImageNet, and 10 on MNIST. The learning rate is set with cosine annealing schedule for fast convergence. Each cycle lasts 45 on ImageNet, 50 epochs on CIFAR-10 and 10 on MNIST. The last epochs

Algorithm 1 Variant of I-FGSM attack to perform approximate Bayesian Model Averaging efficiently on numerous models from several architectures

Input: original example (x, y), S_A ordered sets of model parameters $(\theta_s^1)_{s=1}^S, \ldots, (\theta_s^{S_A})_{s=1}^S$ sampled from the corresponding posterior distribution $\theta_s^i \sim p(\theta_s|\mathcal{D})$, number of iterations n_{iter}, perturbation p-norm ε, step-size α
Output: adversarial example x_{adv}
Shuffle each ordered set of model samples $(\theta_s^1)_{s=1}^S, \ldots, (\theta_s^{S_A})_{s=1}^S$
$x_{\text{adv}} \leftarrow x$
for $i = 1$ **to** n_{iter} **do**
  $x_{\text{adv}} \leftarrow x_{\text{adv}} + \frac{\alpha}{S_A} \sum_{a=1}^{S_A} \nabla \mathcal{L}(x_{\text{adv}};\, y, \theta^a_{i \bmod S})$
  $x_{\text{adv}} \leftarrow \text{project}(x_{\text{adv}}, B_\varepsilon[x])$
  $x_{\text{adv}} \leftarrow \text{clip}(x_{\text{adv}})$
end for

of every cycle form the sampling phase: noise is added and one sample is drawn at the end of each epoch. On CIFAR-10, we obtain 5 samples per cycle (resp. 3 on ImageNet and 4 MNIST), so 30 samples in total (resp. 15 and 20). An illustration of a cSGLD cyclical learning rate schedule is in supplementary materials. To train ResNet-50 models on ImageNet, we re-use the original cSGLD hyperparameters.

Surrogate models (other training methods). Additionally, to Deep Ensemble cSGLD and following Ashukha et al. [2020], we consider 2 Bayesian Deep Learning techniques (SWAG and VI) and 2 Ensemble ones (SSE and FGE). We train every technique on CIFAR-10 and cSGLD and SWAG on ImageNet. We retrieve trained Deep Ensemble, SSE, FGE and VI ImageNet models from Ashukha et al. [2020]. Technique descriptions and experimental setup of surrogates trained with SWAG, VI, FGE, or SSE are detailed below in the Bayesian and Ensemble Training Techniques section.

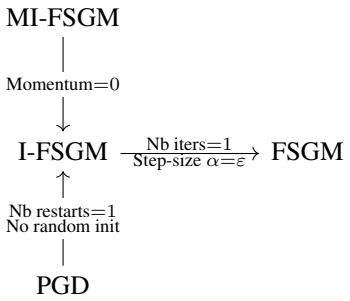


Figure 2: Relationships between gradient-based attacks.

Adversarial attacks. We applied our variant of 4 gradient-based attacks as described in the approach section. The attacker's goal is misclassification (untargeted adversarial examples). We perform both 2-norm and ∞-norm bounded adversarial attacks, and report means and standard deviations computed on 3 random seeds. In accordance to values commonly used in the literature Croce and Hein [2020], the maximum perturbation norm ε is set respectively to 0.5 and $\frac{4}{255}$ on CIFAR-10, and respectively to 3 and $\frac{4}{255}$ on ImageNet. MNIST ones are respectively 3 and 0.1. The step-size α is set to $\frac{\varepsilon}{10}$. We choose to perform 50 iterations such that the transferability rates plateaus for all iterative attacks (I-FGSM, MI-FGSM and PGD) on both norms and both datasets (see Figures 4 and 5 below). PGD runs with 5 random restarts. FGSM aside, *every iteration computes the gradient of 1 model per architecture*. Therefore, the attack computation cost and volatile memory are not multiplied by the size of the surrogate, except for FGSM which computes its unique gradient against all available models. cSGLD samples are attacked in random order. The MI-FGSM decay factor is set to 0.9.

Test-time transformations. In the dedicated section, we consider three test-time transformations applied during attack designed for transferability (see related work section): Ghost Networks Li et al. [2018], Input Diversity Xie et al. [2019] and Skip Gradient Method Wu et al. [2020]. We implemented the first two in PyTorch with their original hyperparameters. To extend Input Diversity to the smaller input sizes of CIFAR-10, we keep the same maximum resize ratio of 0.9. We reuse the original implementation of the third one on ResNet50, and extend it to PreResNet110 (we set its hyperparameter via grid-search, see Figure 8 below).

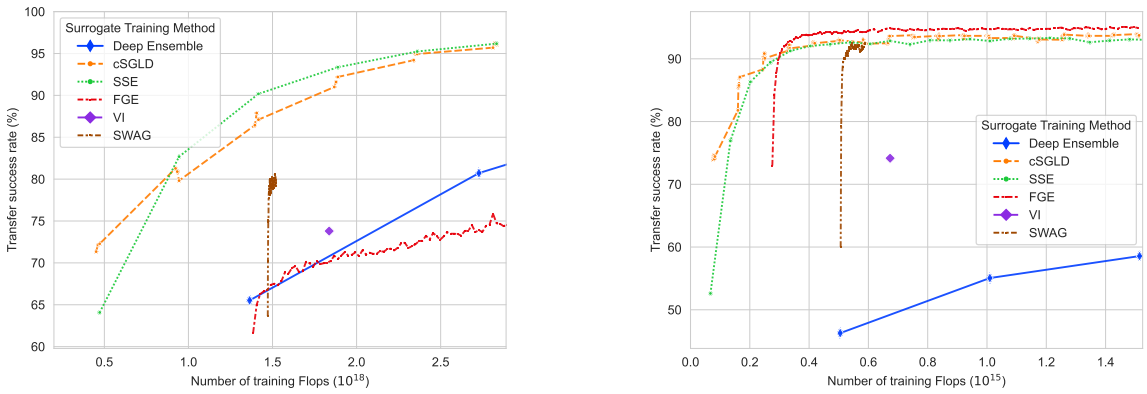
Implementation. The source code of the experiments are publicly available on GitHub[2]. Our attack is built on top of the Python ART library Nicolae et al. [2018]. cSGLD, VI, SSE, and FGE models were trained thanks to the implementation of Ashukha et al. [2020] available on GitHub[3]. All models were trained with PyTorch Paszke et al. [2019]. We use EfficientNet-B0 from timm[4]. We train SWAG on ImageNet with the original implementation Maddox et al. [2019]. We use the following software versions: Python 3.8.8, Pytorch 1.7.1 (1.9.0 for Flops measurement), torchvision 0.8.2, Adversarial Robustness Toolbox 1.6.0, and timm 0.3.2.

Flops. We measure the training computational complexity in Flops using the PyTorch profiler. The computation overhead of one epoch with cSGLD compared to one with SGD/Adam is negligible. The main difference is the addition of noise to the weights during the sampling phase. On CIFAR-10, the overhead of 1 cSGLD epoch of PreResNet110 with added noise compared to one of a DNN trained with Adam (SGD) is 0.0187% Flops (respectively 0.0146% for ResNet50 on ImageNet).

Infrastructure. Experiments were run on Tesla V100-DGXS-32GB GPUs. The server has the following specifications: 256GB RDIMM DDR4, CUDA version 10.1, Linux (Ubuntu) operating system.

B BAYESIAN AND ENSEMBLE TRAINING TECHNIQUES

Following the work of Ashukha et al. [2020], we consider the following training techniques: Deep Ensemble Lakshminarayanan et al. [2016], cSGLD Zhang et al. [2020], SWAG Maddox et al. [2019], VI, SSE Huang et al. [2017], and FGE Garipov et al. [2018]. For computational limitations, we evaluate them on a single attack run (one random seed) of 5000 images.



(a) ImageNet

(b) CIFAR-10

Figure 3: Intra-architecture L∞ I-FGSM success rate with respect to the training computational complexity of six Bayesian and Ensemble methods. Every curve starts with one model, and each successive point is obtained by forming an ensemble with one more model.

Deep Ensemble. Deep Ensemble Lakshminarayanan et al. [2016] simply trains several DNNs independently with random initialization and random subsampling (mini-batch on shuffled data in practice). All DNNs have the same standard hyperparameters for training. For classification, predictions of individual DNNs are averaged. We train 15 PreResNet110, 4 PreResNet164, 4 VGG16bn, 4 VGG19bn, and 4 WideResNet28x10 DNNs on CIFAR-10. We retrieve 15 ResNet50 DNNs trained by Ashukha et al. [2020] on ImageNet, and trained on our own 1 DNN for each of the remaining studied architectures (ResNeXt50 32x4d, DenseNet121, MNASNet 1.0, and EfficientNet-B0).

cSGLD. We refer the reader to the approach section for a detailed description of cyclical Stochastic Gradient Langevin Dynamics. Figure 1 illustrates both the cyclical cosine annealing learning rate schedule and the separation of each cycle into an exploration phase (called the burn-in period of MCMC algorithm) and a sampling phase.

SWAG. Stochastic Weight Averaging-Gaussian (SWAG) Maddox et al. [2019] is a Bayesian Deep Learning method that fits a Gaussian onto SGD iterates to approximate the posterior distribution over weights. Its first moment is the SWA solution,

[2] https://github.com/Framartin/transferable-bnn-adv-ex
[3] https://github.com/bayesgroup/pytorch-ensembles
[4] https://github.com/rwightman/pytorch-image-models

and its second moment a diagonal plus low-rank covariance matrix. Both are estimated from SGD iterates with constant learning rate (0.001 on ImageNet and 0.01 on CIFAR-10). On ImageNet, SWAG performs 10 additional epochs to collect SGD iterates from one of the Deep Ensemble DNNs. On CIFAR-10, a regular pre-training phase of 160 epochs precedes 140 epochs to collect checkpoints. Once fitted, models are sampled from the Gaussian distribution. For every sample, batch normalization statistics are updated in a forward pass over the entire CIFAR-10 train set and over a random subset of 10% on ImageNet. Apart from the fixed initial cost, the marginal computational cost to obtain a sample is very low. We sample a maximum of 50 models because iterative attacks perform 50 iterations of one model per iteration, and further samples would be discarded. Thus, the line corresponding to SWAG in Figure 3 is shorter than the ones of other methods. The rank of the estimated covariance matrix is 20. Batch-size is 128 on CIFAR-10, and 256 on ImageNet.

VI. Variational Inference (VI) approximates the true posterior distribution with a variational approximation, here a fully-factorized Gaussian distribution, and maximizes a corresponding lower bound. A Gaussian prior is chosen. Once trained, the variational approximation is used as the posterior. There is no additional sampling phase to perform Bayesian model averaging. Therefore, we cannot tune the number of samples and a single VI point is plotted in Figure 3. We follow the solutions of Ashukha et al. [2020] to avoid underfitting: pre-training and annealing of β. The first moment of the Gaussian variational approximation is initially set to a DNN pre-trained similarly to Deep Ensemble (300 epochs on CIFAR-10 with initial learning rate of 10^{-4}, and 130 epochs on ImageNet starting at 10^{-3}). The log of its second moment is initially set to -5 on CIFAR-10 and -6 on ImageNet, and further optimized for 100 epochs (45 on ImageNet) with Adam and a learning rate of 10^{-4}. β is set to 10^{-5} on CIFAR-10 and 10^{-4} on ImageNet. Batch-size is 128 on CIFAR-10, and 256 on ImageNet. On MNIST, we train VI using the code and the hyperparameters of Carbone et al. [2020].

SSE. Snapshot ensembles technique Huang et al. [2017] is the foundation of cSGLD. The learning rate is cyclical with a cosine annealing schedule. Contrary to cSGLD, SSE saves a single snapshot per cycle and does not add gradient noise. The cycles are 40 epochs long on CIFAR-10, 45 on ImageNet. The maximum learning rate is 0.2, batch size is 64 on CIFAR-10, respectively 0.1 and 256 on ImageNet.

FGE. Fast Geometric Ensembling Garipov et al. [2018] is a method developed after the empirical observation of Mode Connectivity on CIFAR-10 and CIFAR-100: it's possible to find a path in the parameters space that connects two independently trained DNNs such that the models along the path have low loss and high test accuracy. In practice, it uses a cyclical triangular learning rate and collects one model during each cycle. It is quite similar to SSE, except for the learning rate schedule, the much shorter cycles (4 epochs on CIFAR-10, 2 epochs on ImageNet), and a pre-training phase. Pre-training lasts for 160 epochs on CIFAR-10. On ImageNet, FGE is initialized from one Deep Ensemble checkpoint. The learning rate varies between 5×10^{-5} and 5×10^{-3} on CIFAR-10 and 10^{-6} and 10^{-4} on ImageNet. Batch-size is 128 on CIFAR-10, and 256 on ImageNet.

HMC. Hamiltonian Monte Carlo (HMC) is considered a golden standard to train BNN. We trained the small FC architecture on MNIST, using the code and the hyperparameters of Carbone et al. [2020]. Unfortunately, HMC does not scale to larger DNNs, even on MNIST.

C VANISHING GRADIENTS

Table 2: Proportion of vanished gradients of each 15 individual models and of the ensemble of 15 models (in %). Gradients disappear before and after averaging in similar proportion (except in one case for VI where there is more gradient vanishing after averaging). A gradient vanishes if its L2 norm is lower than 10^{-8}, the numerical tolerance of the Adversarial Robustness Toolbox library. Gradients are on 10 000 original test examples. Means and standard deviations of 15 models are reported when not ensembled.

Dataset	**Architecture**	**Surrogate**	**Vanished individual model gradients**	**Vanished ensemble gradients (averaging)**
ImageNet	ResNet50	cSGLD (ours)	0.06 ±0.06	0.00
		VI	0.15 ±0.02	0.05
		DNN	0.11 ±0.03	0.00
CIFAR-10	PreResNet110	cSGLD (ours)	3.04 ±0.72	2.52
		VI	2.79 ±0.11	2.08
		DNN	59.15 ±0.73	63.96
MNIST	CNN	cSGLD (ours)	30.94 ±2.00	31.67
		DNN	91.53 ±2.36	94.20
	FC	cSGLD (ours)	11.16 ±0.51	11.31
		VI	84.73 ±1.95	91.72
		DNN	90.60 ±1.71	92.14
	Small FC	cSGLD (ours)	4.63 ±0.48	4.71
		VI	60.61 ±4.61	82.00
		HMC	85.61 ±0.02	85.62
		DNN	77.56 ±2.84	79.88

D INTRA-ARCHITECTURE TRANSFERABILITY

Table 3: Intra-architecture transfer success rates of four attacks on PreResNet110 (CIFAR-10) and ResNet50 (ImageNet), in %. Bold is best. Higher is better.

Dataset	Attack	Surrogate	L2 Attack	L∞ Attack	Nb training epochs	Nb backward passes
ImageNet	I-FGSM	cSGLD	94.41 ±0.46	90.77 ±0.09	225	50
		1 DNN	64.95 ±0.54	57.79 ±0.17	130	50
		2 DNNs	80.39 ±0.83	74.25 ±0.71	260	50
		5 DNNs	94.53 ±0.43	92.81 ±0.45	650	50
		15 DNNs	**98.51 ±0.11**	**98.28 ±0.16**	1950	50
	MI-FGSM	cSGLD	93.42 ±0.73	93.61 ±0.41	225	50
		1 DNN	61.11 ±0.35	63.70 ±0.21	130	50
		2 DNNs	77.93 ±0.44	79.27 ±0.76	260	50
		5 DNNs	94.41 ±0.47	95.32 ±0.25	650	50
		15 DNNs	**98.89 ±0.13**	**99.19 ±0.13**	1950	50
	PGD (5 restarts)	cSGLD	91.81 ±0.38	88.76 ±0.24	225	250
		1 DNN	57.47 ±0.52	53.79 ±0.45	130	250
		2 DNNs	74.04 ±0.47	70.90 ±0.41	260	250
		5 DNNs	91.99 ±0.41	91.27 ±0.59	650	250
		15 DNNs	**97.83 ±0.20**	**97.65 ±0.21**	1950	250
	FGSM	cSGLD	58.91 ±0.11	67.17 ±0.26	225	15
		1 DNN	37.37 ±0.19	44.55 ±0.72	130	1
		2 DNNs	46.73 ±0.34	53.91 ±0.60	260	2
		5 DNNs	58.17 ±0.18	65.53 ±0.10	650	5
		15 DNNs	**68.48 ±0.52**	**76.57 ±0.62**	1950	15
CIFAR-10	I-FGSM	cSGLD	**92.38 ±0.23**	92.74 ±0.33	300	50
		1 DNN	43.17 ±0.97	77.59 ±0.01	300	50
		2 DNNs	52.08 ±1.03	84.75 ±0.20	600	50
		5 DNNs	58.74 ±0.98	94.81 ±0.17	1500	50
		15 DNNs	62.08 ±0.92	**97.83 ±0.03**	4500	50
	MI-FGSM	cSGLD	92.29 ±0.25	94.20 ±0.14	300	50
		1 DNN	72.34 ±0.23	80.43 ±0.04	300	50
		2 DNNs	84.10 ±0.33	90.70 ±0.07	600	50
		5 DNNs	91.66 ±0.26	97.04 ±0.07	1500	50
		15 DNNs	**93.87 ±0.30**	**98.30 ±0.11**	4500	50
	PGD (5 restarts)	cSGLD	**91.65 ±0.33**	92.10 ±0.25	300	250
		1 DNN	51.08 ±0.10	77.58 ±0.38	300	250
		2 DNNs	60.60 ±0.06	83.67 ±0.27	600	250
		5 DNNs	67.55 ±0.21	94.19 ±0.07	1500	250
		15 DNNs	70.42 ±0.23	**97.37 ±0.06**	4500	250
	FGSM	cSGLD	**43.13 ±0.00**	58.85 ±0.01	300	30
		1 DNN	20.92 ±0.00	38.89 ±0.01	300	1
		2 DNNs	23.75 ±0.00	45.83 ±0.01	600	2
		5 DNNs	25.60 ±0.00	54.62 ±0.01	1500	5
		15 DNNs	26.71 ±0.00	**61.81 ±0.00**	4500	15

Table 4: Intra-architecture transfer success rates of four attacks on the FC architecture (MNIST), in %. Bold is best. Higher is better.

Dataset	Attack	Surrogate	L2 Attack	L∞ Attack	Nb training epochs	Nb backward passes
MNIST	I-FGSM	cSGLD	**97.65% ±0.02**	**41.49% ±0.02**	50	50
		1 DNN	17.17% ±0.00	34.53% ±0.00	50	50
		2 DNNs	18.52% ±0.01	36.44% ±0.01	100	50
		5 DNNs	26.21% ±0.10	43.12% ±0.16	250	50
		15 DNNs	26.46% ±0.19	45.22% ±0.27	750	50
	MI-FGSM	cSGLD	**97.62% ±0.05**	**42.07% ±0.09**	50	50
		1 DNN	80.72% ±0.00	34.52% ±0.00	50	50
		2 DNNs	82.63% ±0.05	39.83% ±0.06	100	50
		5 DNNs	91.83% ±0.12	44.74% ±0.23	250	50
		15 DNNs	92.08% ±0.09	46.99% ±0.37	750	50
	PGD (5 restarts)	cSGLD	**97.78% ±0.04**	**41.64% ±0.18**	50	250
		1 DNN	31.99% ±0.08	34.80% ±0.07	50	250
		2 DNNs	33.61% ±0.07	37.26% ±0.17	100	250
		5 DNNs	43.27% ±0.37	43.61% ±0.29	250	250
		15 DNNs	44.56% ±0.29	45.50% ±0.29	750	250
	FGSM	cSGLD	**75.09% ±0.00**	**34.90% ±0.00**	50	20
		1 DNN	8.62% ±0.00	22.52% ±0.00	50	1
		2 DNNs	7.42% ±0.00	25.76% ±0.00	100	2
		5 DNNs	7.95% ±0.00	29.52% ±0.00	250	5
		15 DNNs	7.52% ±0.00	31.08% ±0.00	750	15

E INTER-ARCHITECTURE TRANSFERABILITY

Table 5: Inter-architecture transfer success rates of I-FGSM of single architecture surrogate on ImageNet (in %). All combinations of surrogate and targeted architectures are evaluated. Diagonals are intra-architecture. 1 DNN and cSGLD have similar computation budget (135 epochs). Bold is best. Higher is better.

			Target Architecture				
Norm	**Surrogate Architecture**	**Surrogate**	ResNet50	ResNeXt50	DenseNet121	MNASNet	EfficientNetB0
L2	ResNet50	cSGLD	**84.93** ±0.59	**74.70** ±0.91	**71.32** ±0.63	**60.09** ±0.60	**39.70** ±0.29
		1 DNN	56.98 ±0.62	41.13 ±0.97	29.81 ±0.33	27.90 ±0.43	16.39 ±0.46
	ResNeXt50	cSGLD	**79.25** ±0.24	**77.34** ±0.39	**68.53** ±0.19	**62.16** ±0.19	**43.51** ±0.62
		1 DNN	37.48 ±0.52	36.35 ±0.22	23.77 ±0.41	23.69 ±0.21	14.32 ±0.24
	DenseNet121	cSGLD	**63.23** ±1.16	**59.89** ±1.12	**73.28** ±0.45	**60.84** ±0.33	**40.27** ±0.44
		1 DNN	32.61 ±0.29	32.06 ±0.61	39.18 ±0.47	32.01 ±0.44	17.72 ±0.49
	MNASNet	cSGLD	**7.81** ±0.19	**5.97** ±0.37	**9.81** ±0.31	30.41 ±1.45	**15.46** ±0.44
		1 DNN	7.04 ±0.51	5.29 ±0.36	8.41 ±0.20	**32.65** ±0.22	13.13 ±0.06
	EfficientNetB0	cSGLD	**18.93** ±2.17	**14.16** ±1.69	**19.89** ±1.21	**65.97** ±3.60	**49.41** ±3.64
		1 DNN	15.15 ±0.30	13.33 ±0.33	16.12 ±0.71	58.73 ±0.25	48.85 ±0.56
L∞	ResNet50	cSGLD	**78.67** ±1.19	**65.21** ±1.48	**61.54** ±0.83	**51.75** ±1.39	**31.11** ±1.13
		1 DNN	48.03 ±0.94	32.17 ±0.43	23.37 ±0.34	22.60 ±0.40	12.59 ±0.21
	ResNeXt50	cSGLD	**71.67** ±1.00	**69.33** ±0.85	**59.18** ±1.14	**54.75** ±1.33	**35.13** ±0.71
		1 DNN	31.19 ±0.42	28.68 ±0.76	19.12 ±0.07	19.53 ±0.51	11.20 ±0.33
	DenseNet121	cSGLD	**54.13** ±1.70	**50.66** ±1.62	**65.80** ±0.66	**53.43** ±1.30	**32.49** ±0.36
		1 DNN	25.49 ±0.81	23.73 ±0.59	30.78 ±0.21	26.05 ±0.66	13.41 ±0.20
	MNASNet	cSGLD	**6.77** ±0.29	4.72 ±0.27	**8.26** ±0.36	25.27 ±1.83	**12.21** ±0.84
		1 DNN	6.52 ±0.23	**5.06** ±0.12	7.83 ±0.13	**29.19** ±0.05	11.13 ±0.16
	EfficientNetB0	cSGLD	**17.81** ±1.58	**13.91** ±1.45	**19.71** ±1.29	**63.67** ±3.16	46.91 ±3.44
		1 DNN	15.83 ±0.32	13.51 ±0.52	16.78 ±0.38	60.14 ±0.37	**50.16** ±0.64

Table 6: Inter-architecture transfer success rates of I-FGSM of single architecture surrogate on CIFAR-10 (in %). All combinations of surrogate and targeted architectures are evaluated. Diagonals are intra-architecture. Symbols $\star$ indicate 1 DNN having higher transferability than cSGLD. 1 DNN and cSGLD have similar computation budget (300 epochs). Bold is best. Higher is better.

Norm	Surrogate Architecture	Surrogate	Target Architecture				
			PreResNet110	PreResNet164	VGG16bn	VGG19bn	WideResNet
L2	PreResNet110	cSGLD	**88.96** ±0.02	**88.57** ±0.00	26.18 ±0.02	24.38 ±0.00	**63.35** ±0.01
		1 DNN	34.42 ±0.00	34.39 ±0.01	12.66 ±0.01	12.54 ±0.00	26.29 ±0.00
		4 DNNs	50.50 ±0.00	50.49 ±0.00	**27.45** ±0.01	**27.30** ±0.00	46.10 ±0.00
	PreResNet164	cSGLD	**88.28** ±0.01	**87.52** ±0.01	25.83 ±0.01	23.64 ±0.01	**62.79** ±0.01
		1 DNN	33.89 ±0.00	34.36 ±0.01	11.93 ±0.00	12.07 ±0.01	25.95 ±0.01
		4 DNNs	50.36 ±0.01	50.45 ±0.00	**26.79** ±0.01	**27.13** ±0.00	45.94 ±0.00
	VGG16bn	cSGLD	**69.22** ±0.06	**69.03** ±0.03	43.70 ±0.04	38.54 ±0.02	**55.62** ±0.07
		1 DNN	27.22 ±0.04	27.23 ±0.05	29.28 ±0.08	28.73 ±0.02	22.22 ±0.00
		4 DNNs	55.14 ±0.06	54.96 ±0.04	**73.65** ±0.00	**71.24** ±0.04	44.89 ±0.09
	VGG19bn	cSGLD	**69.82** ±0.05	**68.27** ±0.07	44.59 ±0.10	39.76 ±0.13	**54.40** ±0.08
		1 DNN	18.09 ±0.10	18.09 ±0.06	$\star$44.63 ±0.03	$\star$46.76 ±0.03	14.38 ±0.03
		4 DNNs	34.30 ±0.06	33.77 ±0.01	**66.20** ±0.03	**68.87** ±0.05	27.44 ±0.02
	WideResNet	cSGLD	**82.25** ±0.03	**85.06** ±0.02	**26.34** ±0.08	**23.81** ±0.03	**69.31** ±0.07
		1 DNN	22.14 ±0.01	23.00 ±0.00	9.43 ±0.00	9.54 ±0.00	26.85 ±0.00
		4 DNNs	41.07 ±0.00	41.75 ±0.04	22.91 ±0.04	22.65 ±0.03	43.00 ±0.01
L∞	PreResNet110	cSGLD	88.70 ±0.00	88.48 ±0.01	26.32 ±0.00	24.27 ±0.01	62.95 ±0.01
		1 DNN	72.73 ±0.00	74.57 ±0.00	22.26 ±0.00	20.98 ±0.00	47.59 ±0.01
		4 DNNs	**91.98** ±0.00	**92.25** ±0.00	**38.24** ±0.00	**35.56** ±0.00	**72.64** ±0.01
	PreResNet164	cSGLD	87.99 ±0.01	87.74 ±0.00	26.33 ±0.00	23.67 ±0.01	61.83 ±0.02
		1 DNN	68.97 ±0.01	71.76 ±0.00	20.29 ±0.00	18.86 ±0.00	45.07 ±0.00
		4 DNNs	**90.67** ±0.00	**92.22** ±0.00	**37.62** ±0.00	**35.23** ±0.00	**73.18** ±0.00
	VGG16bn	cSGLD	**66.97** ±0.13	**67.48** ±0.11	42.91 ±0.05	37.91 ±0.02	**50.52** ±0.01
		1 DNN	35.57 ±0.02	35.89 ±0.03	38.35 ±0.00	35.82 ±0.00	26.77 ±0.02
		4 DNNs	52.59 ±0.00	53.12 ±0.00	**70.89** ±0.00	**68.53** ±0.00	41.34 ±0.00
	VGG19bn	cSGLD	**67.11** ±0.00	**66.55** ±0.02	43.50 ±0.01	38.72 ±0.02	**49.69** ±0.02
		1 DNN	20.50 ±0.02	20.97 ±0.00	$\star$45.90 ±0.02	$\star$48.60 ±0.02	16.37 ±0.01
		4 DNNs	32.43 ±0.06	32.25 ±0.04	**63.11** ±0.07	**65.64** ±0.06	25.34 ±0.02
	WideResNet	cSGLD	**81.99** ±0.01	**85.63** ±0.01	27.04 ±0.02	23.46 ±0.01	68.43 ±0.01
		1 DNN	49.24 ±0.16	52.84 ±0.03	20.23 ±0.04	18.53 ±0.02	60.84 ±0.09
		4 DNNs	77.45 ±0.01	79.55 ±0.13	**36.33** ±0.13	**33.60** ±0.22	**83.24** ±0.00

Table 7: Inter-architecture transfer success rates of I-FGSM of single architecture surrogate on MNIST (in %). All combinations of surrogate and targeted architectures are evaluated. Diagonals are intra-architecture. cSGLD has always higher transferability than 1 DNN. Symbols $\star$ indicate Bayesian methods (SVI or HMC) having lower transferability than 1 DNN. 1 DNN and cSGLD have similar computation budget (50 epochs). Bold is best. Higher is better.

Norm	Surrogate Architecture	Surrogate Method	Target Architecture		
			Small FC	FC	CNN
L2	Small FC	cSGLD	**99.17** ±0.01	**97.15** ±0.05	**46.04** ±0.15
		HMC	⋆2.66 ±0.01	⋆2.04 ±0.01	⋆0.37 ±0.01
		SVI	⋆5.67 ±0.09	⋆4.04 ±0.09	⋆0.62 ±0.02
		1 DNN	44.19 ±0.00	43.98 ±0.00	19.35 ±0.00
		5 DNNs	48.01 ±0.01	47.78 ±0.04	24.76 ±0.02
		10 DNNs	52.36 ±0.09	51.97 ±0.11	26.52 ±0.05
		15 DNNs	53.13 ±0.09	52.84 ±0.08	27.05 ±0.12
	FC	cSGLD	**98.61** ±0.00	**97.36** ±0.03	**49.27** ±0.17
		SVI	17.16 ±0.17	15.47 ±0.17	⋆4.85 ±0.06
		1 DNN	15.37 ±0.00	15.32 ±0.00	10.40 ±0.00
		5 DNNs	23.13 ±0.06	23.07 ±0.08	16.03 ±0.06
		10 DNNs	24.55 ±0.14	24.46 ±0.13	16.96 ±0.21
		15 DNNs	23.46 ±0.13	23.44 ±0.12	16.44 ±0.21
	CNN	cSGLD	**46.86** ±0.27	**47.06** ±0.32	**92.57** ±0.14
		1 DNN	10.73 ±0.00	10.43 ±0.00	14.80 ±0.00
		5 DNNs	22.20 ±0.09	22.22 ±0.05	28.69 ±0.03
		10 DNNs	19.18 ±0.23	19.27 ±0.34	23.84 ±0.40
		15 DNNs	19.71 ±0.22	19.83 ±0.22	24.33 ±0.26
L∞	Small FC	cSGLD	61.75 ±0.25	37.66 ±0.25	**1.25** ±0.01
		HMC	⋆1.24 ±0.01	⋆0.91 ±0.03	⋆0.10 ±0.01
		SVI	⋆1.76 ±0.02	⋆1.25 ±0.01	⋆0.16 ±0.03
		1 DNN	58.77 ±0.00	32.15 ±0.00	0.95 ±0.00
		5 DNNs	66.81 ±0.02	37.40 ±0.04	1.04 ±0.01
		10 DNNs	67.88 ±0.18	38.22 ±0.02	1.02 ±0.02
		15 DNNs	**68.07** ±0.13	**38.35** ±0.08	1.04 ±0.03
	FC	cSGLD	**60.06** ±0.01	41.04 ±0.02	**1.33** ±0.01
		SVI	⋆4.29 ±0.02	⋆3.18 ±0.05	⋆0.30 ±0.01
		1 DNN	40.15 ±0.00	34.01 ±0.00	1.11 ±0.00
		5 DNNs	51.62 ±0.05	42.66 ±0.17	1.25 ±0.02
		10 DNNs	54.05 ±0.52	44.44 ±0.15	1.26 ±0.02
		15 DNNs	55.03 ±0.45	**44.78** ±0.27	1.27 ±0.01
	CNN	cSGLD	3.07 ±0.08	2.89 ±0.04	5.42 ±0.03
		1 DNN	2.40 ±0.00	2.30 ±0.00	3.83 ±0.00
		5 DNNs	3.50 ±0.03	3.09 ±0.06	6.05 ±0.04
		10 DNNs	3.79 ±0.04	**3.39** ±0.01	6.37 ±0.03
		15 DNNs	**3.81** ±0.09	3.37 ±0.04	**6.55** ±0.05

F TEST-TIME TRANSFERABILITY TECHNIQUES

Table 8: Transfer success rates of (M)I-FGSM attack improved by our approach combined with test-time transformations on CIFAR-10 (in %). Columns are targets. PreResNet110 columns are intra-architecture transferability, others are inter-architecture. Bold is best. Symbols ⋆ are DNN-based techniques better than our vanilla cSGLD surrogate, and † are techniques that do not improve the corresponding vanilla surrogate. The success rate for every cSGLD-based technique is better than its counterpart with 1 DNN.

		Target Architecture				
Norm	**Surrogate**	PreResNet110	PreResNet164	VGG16bn	VGG19bn	WideResNet
L2	1 DNN	34.42 ±0.00	34.39 ±0.01	12.67 ±0.00	12.54 ±0.00	26.29 ±0.01
	+ Input Diversity	59.63 ±0.80	59.79 ±0.75	24.37 ±0.16	23.25 ±0.12	46.09 ±0.47
	+ Skip Gradient Method	57.00 ±0.00	57.66 ±0.04	20.87 ±0.03	20.10 ±0.09	41.80 ±0.04
	+ Ghost Networks	79.22 ±0.30	80.38 ±0.16	⋆32.03 ±0.25	⋆28.63 ±0.17	56.65 ±0.24
	+ Momentum	67.12 ±0.07	67.80 ±0.00	20.49 ±0.02	19.15 ±0.01	44.11 ±0.04
	+ Input Diversity	81.44 ±0.32	82.69 ±0.29	27.64 ±0.03	25.82 ±0.42	57.29 ±0.12
	+ Skip Gradient Method	73.52 ±0.00	75.23 ±0.01	24.52 ±0.00	22.76 ±0.00	49.73 ±0.00
	+ Ghost Networks	77.44 ±0.28	79.13 ±0.12	⋆28.98 ±0.57	25.74 ±0.18	54.06 ±0.04
	cSGLD	90.67 ±0.39	89.74 ±0.31	28.05 ±0.33	26.12 ±0.14	67.27 ±0.89
	+ Input Diversity	92.45 ±0.14	91.80 ±0.14	33.69 ±0.28	31.35 ±0.28	72.41 ±0.76
	+ Skip Gradient Method	92.46 ±0.17	92.10 ±0.28	31.96 ±0.53	29.84 ±0.34	71.04 ±1.23
	+ Ghost Networks	**92.73 ±0.21**	**92.20 ±0.07**	**36.17 ±0.39**	**33.08 ±0.32**	**74.77 ±0.10**
	+ Momentum	†90.35 ±0.37	89.77 ±0.28	†26.89 ±0.37	†25.02 ±0.29	†65.98 ±0.52
	+ Input Diversity	92.31 ±0.33	91.58 ±0.23	31.92 ±0.49	29.72 ±0.46	70.94 ±0.31
	+ Skip Gradient Method	92.33 ±0.34	91.94 ±0.41	31.95 ±0.29	29.85 ±0.28	70.96 ±0.65
	+ Ghost Networks	92.42 ±0.16	91.93 ±0.25	33.02 ±0.60	29.77 ±0.14	72.28 ±0.53
L∞	1 DNN	72.73 ±0.00	74.58 ±0.01	22.26 ±0.00	20.98 ±0.00	47.59 ±0.01
	+ Input Diversity	81.29 ±0.18	82.77 ±0.12	28.10 ±0.22	26.17 ±0.25	57.04 ±0.10
	+ Skip Gradient Method	77.92 ±0.00	79.50 ±0.01	27.43 ±0.00	25.31 ±0.01	53.39 ±0.00
	+ Ghost Networks	74.92 ±0.08	77.23 ±0.26	⋆29.61 ±0.19	26.31 ±0.30	52.93 ±0.05
	+ Momentum	76.12 ±0.01	78.05 ±0.00	23.77 ±0.02	22.33 ±0.01	50.49 ±0.01
	+ Input Diversity	84.66 ±0.19	86.38 ±0.12	⋆31.47 ±0.05	⋆28.89 ±0.31	61.60 ±0.16
	+ Skip Gradient Method	79.72 ±0.02	80.80 ±0.02	28.75 ±0.01	26.12 ±0.00	55.74 ±0.00
	+ Ghost Networks	80.34 ±0.34	82.59 ±0.42	⋆34.17 ±0.48	⋆29.37 ±0.18	60.62 ±0.40
	cSGLD	90.98 ±0.40	90.26 ±0.35	29.26 ±0.53	26.97 ±0.43	67.18 ±1.03
	+ Input Diversity	92.46 ±0.14	91.62 ±0.16	33.81 ±0.25	30.84 ±0.34	71.15 ±0.92
	+ Skip Gradient Method	93.38 ±0.50	92.84 ±0.25	35.68 ±0.61	32.43 ±0.52	73.55 ±1.08
	+ Ghost Networks	91.66 ±0.40	91.32 ±0.19	34.77 ±0.09	31.01 ±0.27	71.60 ±0.40
	+ Momentum	92.84 ±0.18	92.18 ±0.28	32.03 ±0.49	28.53 ±0.38	71.56 ±0.25
	+ Input Diversity	94.05 ±0.31	93.53 ±0.21	37.31 ±0.38	33.23 ±0.23	75.40 ±0.25
	+ Skip Gradient Method	**94.64 ±0.26**	**94.29 ±0.31**	**38.08 ±0.27**	**34.28 ±0.17**	**76.62 ±0.50**
	+ Ghost Networks	93.76 ±0.14	93.75 ±0.13	38.01 ±0.44	33.15 ±0.36	76.23 ±0.29

G ATTACK AND TRAINING HYPERPARAMETERS

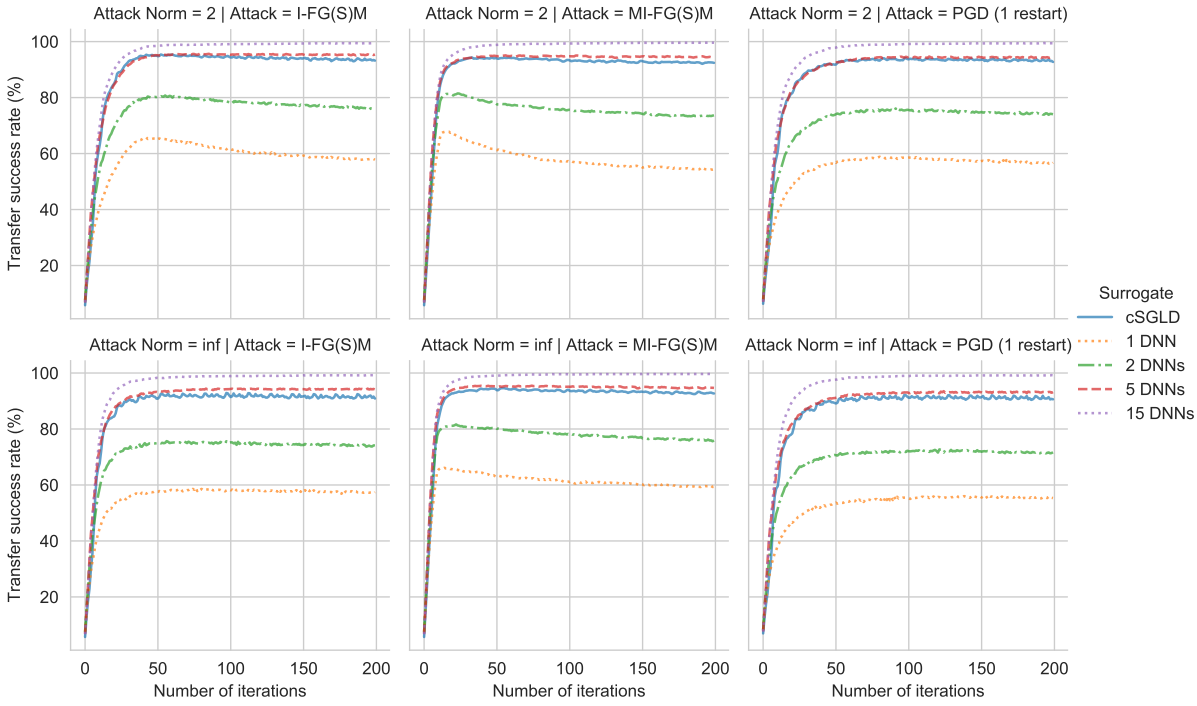


Figure 4: Transfer success rates on ImageNet of three iterative gradient-based attacks on the same architecture (ResNet-50) with respect to the number of iterations.

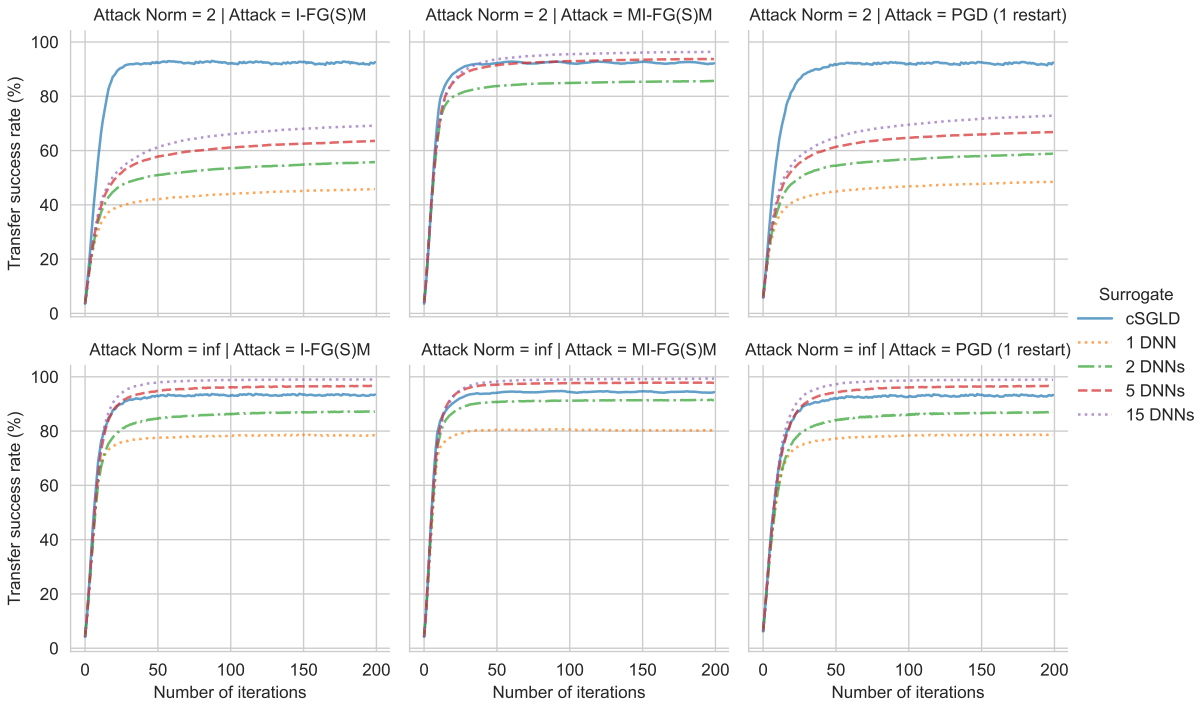


Figure 5: Transfer success rates on CIFAR-10 of three iterative gradient-based attacks on the same architecture (PreResNet110) with respect to the number of iterations.

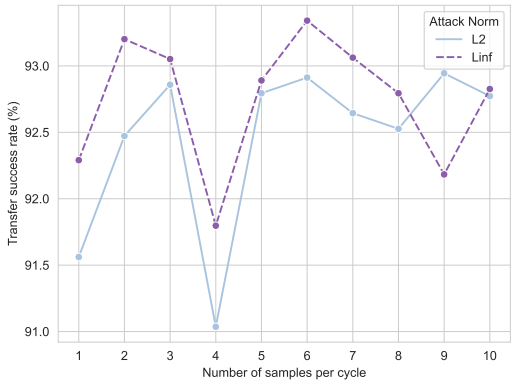


Figure 6: Intra-architecture transfer success rate of I-FGSM with respect to the number of cSGLD samples per cycle. We train one PreResNet110 cSGLD on CIFAR-10 for every number of cycles, from 1 to 10 samples per cycle. Each additional sample per cycle increases the training cost by 1 epoch per cycle (starting at 48 epochs per cycle). A fixed number of 5 cSGLD cycles is used.

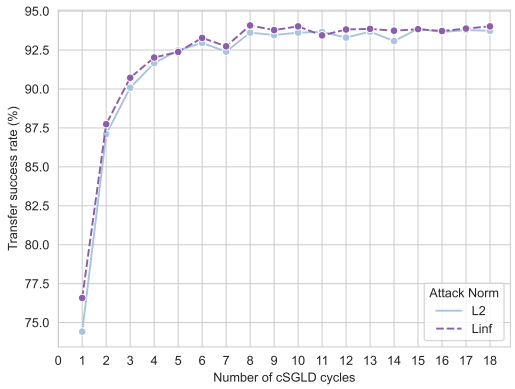


Figure 7: Intra-architecture transfer success rate of I-FGSM with respect to the number of cSGLD cycles on CIFAR-10 (PreResNet110).

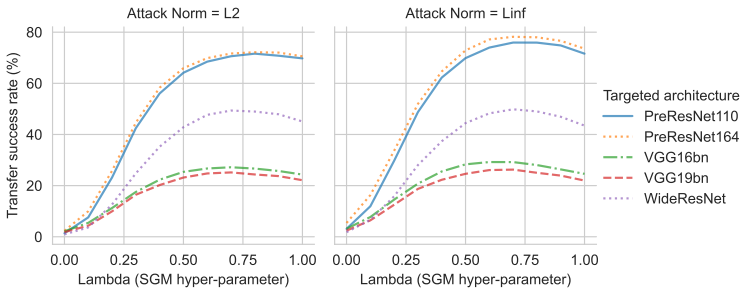


Figure 8: Transfer success rates of the test-time transferability technique Skip Gradient Method with varying values of its hyperparameter γ between 0 and 1 with 0.1 steps. The surrogate is a PreResNet110 DNN trained on CIFAR-10 and evaluated on 1 independently trained DNN for every targeted architecture. The plain line represents the intra-architecture transferability, and the dotted ones the inter-architecture transferability. Adversarial examples are crafted from a validation set randomly sampled from the train set. $\gamma = 0.7$ is selected in the rest of the paper for PreResNet110.

Table 9: Hyperparameters used to train cSGLD or Deep Ensemble. The $\star$ symbols refer to the inter-architecture and test-time techniques sections, and $\star\star$ to the Bayesian and Ensemble training methods section. We do not include target DNNs on ImageNet, since they are pretrained models from PyTorch and timm.

		CIFAR-10			ImageNet	
Method	**Hyperparameter**	**cSGLD**	**DNN Surrogate**	**DNN Target**	**cSGLD**	**DNN Surrogate**
All	Number epochs	50 per cycle	300	300	45 per cycle	130 (135 for $\star$)
	Initial learning rate	0.5	0.01	0.01	0.1	0.1
	Learning rate schedule	Cosine Annealing	Step size decay ($\times$0.1 each 75 epochs)	Step size decay ($\times$0.1 each 75 epochs)	Cosine Annealing	Step size decay ($\times$0.1 each 30 epochs)
	Optimizer	cSGLD	Adam	Adam	cSGLD	SGD
	Momentum	0	0.9	0.9	0.9	0.9
	Weight decay	5e-4 (3e-4 for PreResNet)	1e-4	1e-4	1e-4	1e-4
	Batch-size	64	128	128	256 for ResNet50, 64 for others	256 for ResNet50, 64 for others
cSGLD	Sampling interval	1 sample per epoch	-	-	1 sample per epoch	-
	Nb cycles	6 (18 for $\star\star$)	-	-	5 (3 for $\star$, 6 for $\star\star$)	-
	Nb samples per cycle	5	-	-	3	-
	Nb epochs with noise	5	-	-	3	-

Table 10: Hyperparameters of attacks and test-time transferability techniques.

Attack / Technique	**Hyperparameter**	**ImageNet**	**CIFAR-10**	**MNIST**
All attacks	Perturbation 2-norm ε	3	0.5	3
	Perturbation ∞-norm ε	$\frac{4}{255}$	$\frac{4}{255}$	0.1
Iterative Attacks	Step-size α	$\frac{\varepsilon}{10}$	$\frac{\varepsilon}{10}$	$\frac{\varepsilon}{10}$
	Number iterations	50	50	50
MI-FGSM	Momentum term	0.9	0.9	0.9
PGD	Number random restarts	5	5	5
Ghost Network	Skip connection erosion random range	[1-0.22, 1+0.22]	[1-0.22, 1+0.22]	-
Input Diversity	Minimum resize ratio	90 %	90 %	-
	Probability transformation	0.5	0.5	-
Skip Gradient Method	Residual Gradient Decay γ	0.2 (ResNet50)	0.7 (PreResNet110)	-