Efficient and Transferable Adversarial Examples from Bayesian Neural Networks

Martin Gubri Maxime Cordy Mike Papadakis Yves Le Traon

University of Luxembourg

Background

- **Adversarial example** worst-case distributional shift at test-time (no alteration of target model).
- **Transferability** an adversarial example against a model is likely to be also adversarial against another model. Can be leverage to attack an unknown model.
- **Test-time transferability techniques** a wide range of techniques applied during the attack to increase the transferability. Applied either on the model (ghost networks), the input (input diversity), or the gradient (translation invariant).

Transferability is a Bayesian inference problem

Ensemble approach

Ensemble → enrich the space of hypothesis (**diversity**) → ↗transferability.
[2] proposes an ensemble of 1 deterministic NN per architecture.

Bayesian approach (ours)

Motivation Transferability is fundamentally related to the notion of **knowledge**: the (deterministic) parameters of the target model are unknown, and can therefore be treated as random variables. The posterior represents the belief about the value of each parameter.
Bayesian NN consider epistemic uncertainty → ↗transferability.

Claim *The best transferable adversarial example minimizes the true Bayesian posterior predictive distribution* $p(y|x, \mathcal{D}) = \mathbb{E}_{p(\theta|\mathcal{D})} p(y|x, \theta)$.

Threat model In black-box setting without oracle access, the targeted classifier is unknown. Assuming that (i) its architecture is known and so is $\hat{p}(y|x, \bullet)$ (extension to unknown architecture in the paper), (ii) its training set $\mathcal{D}$ is known, (iii) its parameters θ_t are unknown, one can treat θ_t as a random variable s.t. $\theta_t \sim p(\theta|\mathcal{D})$.

This approach offers a flexible and rigorous framework to include prior knowledge on the target model (architecture, training data, etc.).

Method

- **cSGLD** We choose cSGLD [3], a Bayesian Deep Learning technique, to sample approximately from $p(\theta|\mathcal{D})$ during training. It is a cyclical SG-MCMC method (Figure 1) that adds noise to parameters during sampling phase. cSGLD is comparatively efficient for natural accuracy [1].
- **Training cost** We consider the computational cost of training the surrogate model. The computational cost of the attack is kept constant.
- **Attack** Apply standard gradient-based attack on 1 parameters sample per architecture per iteration → **no** computational overhead compared to [2].
- **Target** the target model is a deterministic NN. Same hyperparameters (except random seed) than Deep Ensemble surrogate.

References

[1] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of In-Domain Uncertainty Estimation and Ensembling in Deep Learning. feb 2020.

[2] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, nov 2017.

[3] Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning. *International Conference on Learning Representations (ICLR)*, feb 2020.

Approach

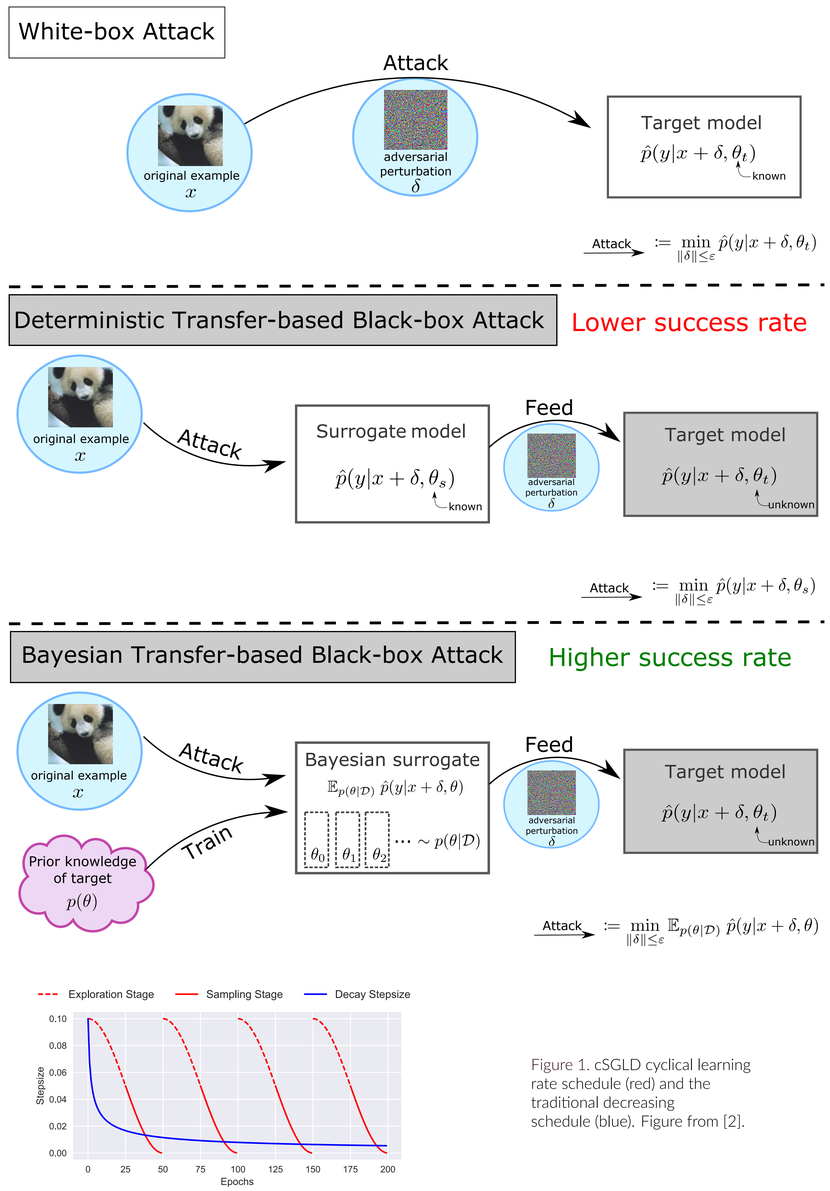

Figure 1. cSGLD cyclical learning rate schedule (red) and the traditional decreasing schedule (blue). Figure from [2].

Results

Intra-architecture transferability

Dataset	Attack	Norm	T-DEE	Comput. Ratio
CIFAR-10	I-FG(S)M	L2	>15	>15
		L∞	3.80	3.80
	MI-FG(S)M	L2	6.08	6.08
		L∞	2.98	2.98
	PGD	L2	>15	>15
		L∞	4.00	4.00
	FG(S)M	L2	>15	>15
		L∞	7.35	7.35
ImageNet	I-FG(S)M	L2	4.93	2.85
		L∞	4.18	2.42
	MI-FG(S)M	L2	4.83	2.79
		L∞	4.39	2.54
	PGD	L2	4.90	2.83
		L∞	4.25	2.46
	FG(S)M	L2	5.31	3.07
		L∞	6.16	3.56

Table 1. Number of DNNs (T-DEE) and computation budget (Comput. Ratio, measured as number of epochs) needed to achieve the same intra-architecture transferability than cSGLD using classical deep ensemble. Higher is better. ">15" indicates that 15 DNNs have a lower transfer rate than cSGLD.

Inter-architecture transferability

Attack	Surrogate	−ResNet50	−ResNeXt50	−DenseNet121	−MNASNet	−EfficientNetB0	Nb epochs
L2	1 cSGLD per arch.	**93.36 %**	**90.84 %**	**92.36 %**	**96.06 %**	**81.78 %**	4×135
	1 DNN per arch.	73.58 %	72.40 %	65.40 %	84.22 %	54.70 %	4×135
L∞	1 cSGLD per arch.	**92.42 %**	**89.66 %**	**91.00 %**	**96.02 %**	**79.82 %**	4×135
	1 DNN per arch.	70.14 %	69.12 %	61.76 %	82.28 %	50.62 %	4×135

Table 2. Success rates of I-FG(S)M attack on ImageNet hold-out architectures. Higher is better.

Test-time transferability techniques

Dataset	Test-time Transformation	Base model	L2 Attack	L∞ Attack	Nb epochs	Nb backwards
CIFAR-10	Ghost Networks	cSGLD	**95.29 %**	**94.54 %**	250	50
		1 DNN	83.67 %	80.41 %	250	50
	Input Diversity	cSGLD	**94.55 %**	**94.58 %**	250	50
		1 DNN	69.42 %	85.15 %	250	50
	Translation Invariant⋆	cSGLD	**22.42 %**	**21.03 %**	250	50
		1 DNN	17.28 %	18.87 %	250	50
	Baseline (None)	cSGLD	93.46 %	93.42 %	250	50
		1 DNN	49.02 %	77.96 %	250	50
ImageNet	Ghost Networks	cSGLD	**98.06 %**	**96.92 %**	225	100
		2 DNNs	96.08 %	93.46 %	260	100
	Input Diversity	cSGLD	**98.08 %**	**97.12 %**	225	100
		2 DNNs	95.60 %	94.36 %	260	100
	Translation Invariant⋆	cSGLD	**61.68 %**	**56.86 %**	225	100
		2 DNNs	45.46 %	43.42 %	260	100
	Baseline (None)	cSGLD	95.54 %	93.54 %	225	100
		2 DNNs	84.88 %	80.58 %	260	100

Table 3. Success rates of I-FG(S)M attack improved by our approach combined with test-time transformations.