Strengthening the Transferability of Adversarial Examples Using Advanced Looking Ahead and Self-CutMix

Donggon Jang* Sanghyeok Son* Dae-Shik Kim
Korea Advanced Institute of Science and Technology (KAIST)
{jdg900, ssh816, daeshik}@kaist.ac.kr

Abstract

Deep neural networks (DNNs) are vulnerable to adversarial examples generated by adding malicious noise imperceptible to a human. The adversarial examples successfully fool the models under the white-box setting, but the performance of attacks under the black-box setting degrades significantly, which is known as the low transferability problem. Various methods have been proposed to improve transferability, yet they are not effective against adversarial training and defense models. In this paper, we introduce two new methods termed Lookahead Iterative Fast Gradient Sign Method (LI-FGSM) and Self-CutMix (SCM) to address the above issues. LI-FGSM updates adversarial perturbations with the accumulated gradient obtained by looking ahead. A previous gradient-based attack is used for looking ahead during N steps to explore the optimal direction at each iteration. It allows the optimization process to escape the sub-optimal region and stabilize the update directions. SCM leverages the modified CutMix, which copies a patch from the original image and pastes it back at random positions of the same image, to preserve the internal information. SCM makes it possible to generate more transferable adversarial examples while alleviating the overfitting to the surrogate model employed. Our two methods are easily incorporated with the previous iterative gradient-based attacks. Extensive experiments on ImageNet show that our approach acquires state-of-the-art attack success rates not only against normally trained models but also against adversarial training and defense models.

1. Introduction

Since [19] shows the existence of adversarial examples that mislead deep neural networks, it evokes a great deal of concern especially in security-sensitive fields, *e.g*. autonomous driving, medical imaging, and face verification. Existing numerous attack methods [1, 5, 8, 12] have shown satisfactory attack success rates, assuming the white-box setting where the parameters and architectures of target models are known. Yet, in the real scenario, they can not be easily deployed as we usually do not have information about the target models. This is known as the black-box setting.

In the [5], they demonstrate that adversarial examples generated by white-box models can also attack the black-box models with high success rates, which is referred to as *transferability*. There are some approaches to enhance transferability for black-box attacks. [3, 10] try to improve the transferability by developing the gradient-based method with momentum. [4, 25] incorporate input transformation in terms of overfitting to surrogate models. However, they are not effective against adversarial training and defense models.

In this work, we propose two new methods, namely Lookahead Iterative Fast Gradient Sign Method (LI-FGSM) and Self-CutMix (SCM), to alleviate the above issues.

Motivated by standout results of NI-FGSM [10] that uses momentum to look ahead, we attempt to develop an advanced looking ahead method. We suppose that looking ahead only once may hinder the boost in performance. Therefore, inspired by lookahead optimizer [28], we design a loop inside MI-FGSM [3] to look ahead more times. Lookahead optimizer updates parameters by interpolating initial weights and last weights after looking ahead k steps at each iteration. It can improve convergence as well as reduce the variance. By employing a similar approach, we expect to stabilize the update directions and escape poor local maxima.

SI-NI-FGSM [10] considers that the transferability of adversarial examples can be equivalent to the generalization of deep neural networks from a perspective of training neural networks. Various input transformations [4, 10, 25] have been introduced to improve the transferability of adversarial examples, which can be treated as data augmentation for generalization. Recently, in the classification model training, CutMix [26], in which patches are cut and pasted among the training images, is widely used to improve the

*These authors contributed equally.

generalization of deep learning models. However, applying CutMix directly as the input transformation may introduce incorrect update direction for crafting adversaries because the images with different classes are mixed. Therefore, we propose a new input transformation method, Self-CutMix, that performs CutMix in the single image itself while preserving the input information of the image.

Extensive experiments on ImageNet [16] demonstrate that each method exhibits noticeable results even when used alone. On seven advanced defense methods, our method successfully attacks them, thus indicating that our method is suitable for attacking various defense models.

2. Background

Adversarial examples are generated by maximizing the loss of classifier w.r.t input images while the perturbation is constrained by the L_p norm. It can be formulated as follows.

$$\arg\max_{x^{adv}} J(x^{adv}, y^{true}) \quad s.t. \quad ||x^{adv} - x||_p \leq \epsilon \tag{1}$$

where J is the loss function, x and x^{adv} correspond to benign and adversarial examples respectively, y^{true} is a ground truth label, ϵ is the maximum perturbation, and $||\cdot||_p$ denotes the L_p norm.

2.1. Gradient-based Attacks

Fast Gradient Sign Method (FGSM) [5] adds gradients of loss w.r.t benign images after applying sign function for update based on the linear hypothesis.

$$x^{adv} = x + \epsilon \cdot \text{sign}(\bigtriangledown_x J(x, y^{true})) \tag{2}$$

Iterative-FGSM (I-FGSM) [8] improves FGSM by modifying to the iterative approach with a clipping function.

$$x_{t+1}^{adv} = x_t^{adv} + \text{Clip}_\epsilon\{\alpha \cdot \text{sign}(\bigtriangledown_x J(x_t^{adv}, y^{true}))\} \tag{3}$$

where α denotes the small step size.

Most attack algorithms including the above methods achieve noticeable performance, yet low transferability in the target models. MI-FGSM [3] utilizes momentum to mitigate the issue based on I-FGSM. It can deviate the poor local maxima by adapting the concept of SGD with momentum [15].

$$g_{t+1} = \mu \cdot g_t + \frac{\bigtriangledown_x J(x_t^{adv}, y^{true})}{|| \bigtriangledown_x J(x_t^{adv}, y^{true})||_1} \tag{4}$$

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1}) \tag{5}$$

where μ denotes the decay factor.

Further, NI-FGSM [10] leverages Nesterov's accelerated gradient [14] so that it can improve transferability. It looks ahead with the momentum in advance to calculate the gradients represented by

$$x_t^{nes} = x_t^{adv} + \alpha \cdot \mu \cdot g_t \tag{6}$$

Unlike the above methods that utilize momentum, variance tuning [21] focuses on reducing variance. It tunes the gradients of the current iteration with variance, which is calculated by subtracting gradients from averaged gradients w.r.t neighbors of adversarial example in the previous iteration.

$$v_t = \frac{1}{N}\sum_{i=1}^{N} \bigtriangledown_x J(x_{t-1}^i, y^{true}) - \bigtriangledown_x J(x_{t-1}^{adv}, y^{true}) \tag{7}$$

$$g_{t+1} = \mu \cdot g_t + \frac{\hat{g}_{t+1} + v_t}{||\hat{g}_{t+1} + v_t||_1} \tag{8}$$

where x_{t-1}^i is a sample neighboring x_{t-1}^{adv}, $\hat{g}_{t+1}$ is gradient at t iteration, and v_t is the variance.

Similar to variance tuning, EMI-FGSM [23] gets the average gradient from the neighbor samples along the gradient direction of the previous iteration.

$$\bar{x}_t^{adv}[i] = x_t^{adv} + c_i \cdot \bar{g}_{t-1} \tag{9}$$

$$\bar{g}_t = \frac{1}{N}\sum_{i=1}^{N} \nabla_{\bar{x}_t^{adv}[i]} J_f(\bar{x}_t^{adv}[i], y) \tag{10}$$

where N is the number of samples and c_i is the sampling coefficient between $[-\eta, \eta]$.

2.2. Input Transformations

Various input transformations are adopted to craft the more transferable adversarial examples.

DIM [25] applies the transformations, which are composed of random resizing and random padding, to the inputs. By increasing the input diversity at each iteration, DIM improves the transferability of adversarial examples. Further, DIM can be integrated with existing gradient-based attack methods.

TIM [4] uses a set of translated images to optimize adversarial perturbation. By convolving the gradients of the untranslated images with a Gaussian kernel, TIM calculates the derivatives of a loss function efficiently. TIM can also be integrated with existing gradient-based attack methods.

SIM [10] utilizes the deep learning classifier's scale-invariance property. The gradients of a loss function are calculated for a set of images scaled by $1/2^i$ on the input image. Likewise, it improves the attack transferability by being integrated with other existing gradient-based attacks.

Admix [22] applies the input transformation similar with Mixup [27]. For each input images, a set of images from other categories are randomly sampled and a minor portion for each sampled image is mixed with input image to craft a set of diverse images.

3. Methodology

3.1. Lookahead Iterative Fast Gradient Sign Method (LI-FGSM)

Unlike the existing approaches, Lookahead Iterative Fast Gradient Sign Method (LI-FGSM) obtains the gradients from the inner loop corresponding to looking ahead. Inputs are optimized by the off-the-shelf gradient-based attack method during N steps in the loop while accumulating the gradients. The accumulated ones named lookahead gradients can be a good proxy, which represents optimal direction. After escaping the loop, the normalized lookahead gradients are adapted to input examples for the updates. We also use the momentum to inherit the advantages of MI-FGSM as in Eq. (4) and Eq. (5). Therefore, LI-FGSM can run away from local maxima, which are suspected as the cause of overfitting, reduce the variance, and boost transferability. The algorithm is summarized in Algorithm 1 and Algorithm 2.

Any kind of iterative gradient-based attack methods, *e.g.* M(N)I-FGSM, can be leveraged for looking ahead. In this work, we utilize VNI-FGSM [21] that exhibits the highest performance among others in the experiments discussed in the supplementary material.

Algorithm 1 LI-FGSM: Inner Loop

Input: A classifier f with loss function J; an example x_t^{adv} at step t with ground truth label y^{true}
Input: The maximum perturbation ϵ and decay factor μ
Input: The number of looking ahead steps N, the number of mixed copies c
Output: A lookahead gradient l

```
1: $\alpha = \epsilon/T$; $x_0^{adv} = x_t^{adv}$; $l = 0$; $g_0 = 0$
2: for $n = 0$ to $N-1$ do
3:     Get $x_n^{nes}$ by Eq. (6)
4:     Get average gradients as $\hat{g}_{n+1} = \frac{1}{c+1} \cdot \hat{g}$
5:     Calculate variance $v_n$ by Eq. (7)
6:     Update $g_{n+1}$ by Eq. (8)
7:     Accumulate the gradients as $l = l + g_{n+1}$
8:     Update $x_{n+1}^{adv}$ by Eq. (5)
9: end for
10: return $l$
```

3.2. Self-CutMix (SCM)

Let $x \in R^{W \times H \times C}$ be an input image of width W, height H, and channel C. **B** = (r_x, r_y, W_p, H_p) denotes a bounding box where r_x and r_y are the coordinates of top-left corner and W_p and H_p are the copied (pasted) patch size. The goal of Self-CutMix is to generate a new sample by cropping a patch from the input image itself and then pasting it back into the original image. We introduce two variants of SCM.

Algorithm 2 LI-FGSM

Input: A classifier f with loss function J; a benign example x with ground truth label y^{true}
Input: The maximum perturbation ϵ, the number of iteration T, and decay factor μ
Output: An adversarial example x^{adv}

```
1: $\alpha = \epsilon/T$; $x_0^{adv} = x$; $g_0 = 0$
2: for $t = 0$ to $T-1$ do
3:     Get Lookahead gradients $\hat{g}$ by Algorithm 1
4:     Update $g_{t+1}$ by $g_{t+1} = \mu \cdot g_t + \frac{\hat{g}}{||\hat{g}||_1}$
5:     Update $x_{t+1}^{adv}$ by Eq. (5)
6: end for
7: return $x^{adv} = x_T^{adv}$
```

SCM-P We first sample bounding box coordinates **B1** = $(r_{x1}, r_{y1}, W_p, H_p)$. The region matched to **B1** is copied from the input image x. To paste the copied patch back to the original image, we sample another bounding box **B2** = $(r_{x2}, r_{y2}, W_p, H_p)$. The region matched to **B2** on input image x is removed and filled with the region **B1**. While the CutMix replaces the image region with a patch from another image belonging to a different class, the major difference is that SCM-P replaces the image region with a copied patch from the input image itself.

SCM-R SCM-R is a variant of SCM. The difference with SCM-P is that the entire image is cloned to preserve the global information of the input image. We sample the bounding box **B2** = $(r_{x2}, r_{y2}, W_p, H_p)$ on the input image. Then, we resize the copied image into size (W_p, H_p) and insert the resized image at the region matched to **B2** on the input image.

With the above input transformation methods, we propose the SCM attack method that can improve the transferability of the adversarial examples, which optimizes the perturbation over a set of mixed images $SCM(x)$.

$$\hat{g}_{t+1} = \bigtriangledown_x J(x_t^{adv}, y^{true}) + \sum_{i=1}^{c} \bigtriangledown_x J(s_i \cdot SCM(x_t^{adv}), y^{true}) \tag{11}$$

where the c denotes the number of the mixed images, $s_i \in (0, 1]$ denotes the scale factor of mixed image $SCM(x_t^{adv})$.

Since we implement the input transformation in the same image, we do not mix the labels. In both SCM-P and SCM-R, the patch size can be fixed or chosen randomly. In our experiments, we uniformly sample the bounding box coordinates according to

$$r_x \sim Unif(0, W - W_p), \quad W_p \sim Unif(P, W) \tag{12}$$

$$r_y \sim Unif(0, H - H_p), \quad H_p \sim Unif(P, H) \tag{13}$$

where the $Unif$ denotes the uniform distribution, P denotes the minimum patch size. Experiments according to varying patch sizes are reported in the supplementary material.

SCM can be integrated with other input transformations, *e.g*. DIM, TIM, and SIM. Furthermore, SCM increases the variance of the input distribution, while maintaining an identity of inputs and thus avoiding overfitting to white-box models. Consequently, it makes the adversarial examples more transferable.

4. Experiments

4.1. Experimental Setup

We consider untargeted L_∞ attacks on image classifiers. For the dataset, we evaluate our method by randomly choosing 1,000 clean images belonging to the 1,000 classes from the ImageNet validation set [16] provided by [10]. And those images are almost accurately classified by all the testing models as in [3, 10].

We choose four normally trained models, *i.e*. Inception-v3 [18], Inception-v4, Inception-Resnet-v2 [17], and Resnet-v2-101 [6] as source models. To demonstrate the effectiveness of our attack method, we adopt three adversarially trained models, *i.e*. Inc-v3$_{ens3}$, Inc-v3$_{ens4}$, and IncRes-v2$_{ens}$ [20]. Additionally, we consider seven advanced defense models which are robust against adversarial attacks on ImageNet [16], *i.e*. HGD [9], R&P [24], NIPS-r3[1], FD [11], Comdefend [7], RS [2], and NRP [13].

For comparison with the proposed method, we take four momentum-based iterative attack methods, *i.e*. MI-FGSM [3], NI-FGSM [10], VM(N)I-FGSM [21], and EMI-FGSM [23], and adopt four input transformation methods, *i.e*. DIM [25], TIM [4], SIM [10], and Admix [22] as our baselines. Implementation details are described in the supplementary material.

4.2. Attacking a Single Network

We evaluate the attack success rates of the proposed methods, SCM-P(R) and LI-FGSM, with other baselines under a single model. We craft the adversaries on the four normally trained models and validate the performance on the seven neural networks. We first compare the attack success rates of our proposed SCM-P(R) with other input transformations. As shown in Tab. 1, SCM-P(R) outperform the baselines by a large margin on all models under the black-box setting. Specifically, SCM-P(R) exhibit better transferability on adversarially trained models and achieves nearly 100% success rates on the white-box models. It is because SCM-P(R) increase the variance of the input distribution and alleviates overfitting to the source model, thereby enhancing transferability. We also compare LI-FGSM and other gradient-based attacks without any input transformation to demonstrate the effectiveness of our method. In Tab. 2, we observe that LI-FGSM surpasses other gradient-based attacks in terms of attack success rates. It indicates that advanced looking ahead may help to stabilize the update directions.

[1]https://github.com/anlthms/nips-2017/tree/master/mmd

4.3. Attacking Defense Models

To show the strong transferability of adversarial examples generated by our methods, we additionally evaluate the performance on seven advanced defense models. We attack the defense methods under a single model setting, not an ensemble model setting to clarify the effectiveness of the proposed methods. As shown in Tab. 3, LI-CT-FGSM shows outstanding performance, compared to other gradient-based attacks. In addition, we compare the performance while adding Admix, a state-of-the-art transformation, and the proposed SCM-P(R) to MI-CT-FGSM. MI-SCM-P-CT-FGSM achieves comparable performance and MI-SCM-R-CT-FGSM outperforms MI-Admix-CT-FGSM on seven defense methods. It shows that our methods can fool the defense methods successfully and will be an attractive option to attack the defense methods.

4.4. Discussion

Stabilize Update Directions The normalized cosine similarity of the gradients between adjacent iterations might be a proxy metric to measure the degree of stabilization. As shown in Fig. 1a, LI-FGSM has a higher value than others in the entire process. From the result, we suppose that stabilized update direction contributes to our outstanding attack performance. The most curious part is that VM(N)I-FGSM and EMI-FGSM exhibit higher attack success rates than M(N)I-FGSM, yet, they show lower similarity than others even they try to reduce the variance. We conjecture that neighborhood sampling can be considered input transformation that impose randomness. Accordingly, it may confuse the stabilization but can get good performance. Our method utilizing VNI-FGSM has the effects of stabilization and uniform noise. Consequently, it surpasses the others.

The Number of Steps Looking Ahead We experiment with varying the number of steps looking ahead. Only the empirical results for LI-VNI-FGSM, which shows the best performance, are included due to space constraints. As shown in Fig. 1b, the success rate grows rapidly as the number of steps increases and then converges at about 17 steps. We use it in all of the experiments in this work.

SCM vs CutMix To show why SCM helps maintain input information and craft more transferable adversarial examples, we compare the attack success rates of CutMix and SCM while varying the pasted patch size. As shown in Fig. 1c and Fig. 1d, the attack success rates of CutMix increase until the patch size is around a certain size and steeply decrease after that. It implies that the increasing patch size from other classes impairs input information, which leads to less transferable directions. However, the attack success rates of SCM consistently increase, which

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3 (ens3)	Inc-v3 (ens4)	IncRes-v2 (ens)
Inc-v3	DIM	98.72*	64.54	60.46	54.04	19.22	17.96	9.56
	TIM	**100.00***	48.02	41.28	40.20	24.10	21.34	13.48
	SIM	**100.00***	69.84	67.86	62.86	32.20	32.47	17.42
	Admix	**100.00***	82.00	80.07	73.13	39.97	38.70	20.13
	SCM-P	**100.00***	**87.28**	**85.04**	80.92	50.16	45.28	27.36
	SCM-R	99.85*	85.90	83.68	**81.50**	**62.58**	**58.94**	**36.86**
Inc-v4	DIM	72.00	97.70*	64.08	55.68	21.68	20.76	11.44
	TIM	57.28	99.66*	47.08	43.38	26.48	22.96	17.42
	SIM	80.56	**99.88***	74.12	67.94	47.42	44.74	28.62
	Admix	88.50	99.83*	84.83	78.47	54.20	50.57	32.60
	SCM-P	**91.12**	99.82*	**88.12**	**83.36**	61.52	58.68	40.22
	SCM-R	89.44	99.82*	86.96	83.04	**70.74**	**68.96**	**51.72**
IncRes-v2	DIM	70.32	64.26	93.14*	59.16	30.44	25.02	17.32
	TIM	62.66	55.48	97.42*	51.10	31.94	27.32	22.24
	SIM	84.68	80.48	**99.00***	75.86	56.28	49.48	42.28
	Admix	89.90	87.00	98.83*	83.63	63.57	56.83	49.40
	SCM-P	**90.94**	**88.28**	98.88*	**85.32**	68.90	62.10	53.18
	SCM-R	89.52	86.60	98.62*	85.10	**75.36**	**71.16**	**62.78**
Res-101	DIM	74.84	68.88	69.90	98.08*	35.88	32.58	19.78
	TIM	59.16	52.30	51.80	99.30*	36.34	31.44	23.10
	SIM	74.84	69.16	69.64	99.72*	42.94	38.92	26.38
	Admix	81.63	76.50	78.07	**99.80***	48.37	42.73	28.40
	SCM-P	**91.20**	**88.85**	**88.34**	99.76*	67.64	61.10	44.00
	SCM-R	89.22	84.96	87.02	99.78*	**74.54**	**71.04**	**54.66**

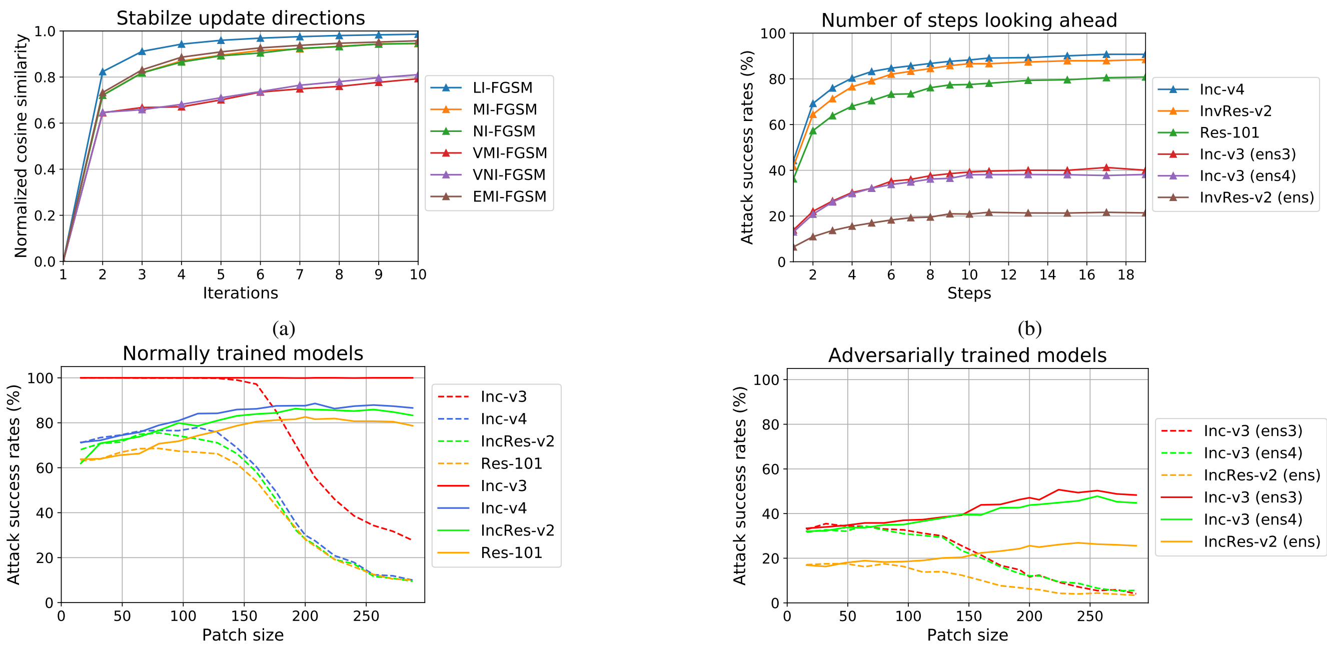
Table 1. Attack success rates (%) against seven models under a single model setting with DIM, TIM, SIM, Admix, and SCM. Inc-v3, Inc-v4, IncRes-v2, and Res-101 using MI-FGSM are adopted as the source model respectively. * indicates that the target model is the same as the source model.

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3 (ens3)	Inc-v3 (ens4)	IncRes-v2 (ens)
Inc-v3	MI-FGSM	**100.00***	45.48	42.48	36.08	13.90	12.50	6.14
	NI-FGSM	**100.00***	51.36	48.12	41.12	12.62	13.28	6.20
	VMI-FGSM	**100.00***	76.90	74.30	67.00	36.80	35.60	19.50
	VNI-FGSM	**100.00***	77.00	75.50	65.30	35.00	32.00	19.40
	EMI-FGSM	**100.00***	72.63	69.93	59.43	21.00	19.30	10.03
	LI-FGSM	**100.00***	**91.02**	**88.28**	**80.66**	**40.14**	**38.84**	**22.00**
Inc-v4	MI-FGSM	55.74	99.86*	46.50	41.44	16.94	14.84	7.98
	NI-FGSM	63.50	**100.00***	52.18	45.50	15.82	13.80	6.72
	VMI-FGSM	81.10	99.00*	75.40	66.20	41.18	39.72	25.86
	VNI-FGSM	82.88	99.98*	75.96	66.96	40.24	38.34	24.66
	EMI-FGSM	87.73	**100.00***	76.37	67.83	27.23	25.17	13.00
	LI-FGSM	**94.00**	**100.00***	**89.18**	**81.76**	**48.40**	**44.20**	**27.88**
IncRes-v2	MI-FGSM	59.24	50.96	98.08*	45.58	22.56	16.14	11.50
	NI-FGSM	62.88	53.92	99.08*	45.26	19.62	15.56	10.06
	VMI-FGSM	79.78	75.18	97.72*	69.70	49.62	42.34	36.88
	VNI-FGSM	81.04	76.38	98.36*	69.68	48.16	40.54	34.08
	EMI-FGSM	88.07	82.97	99.47*	73.53	37.20	30.80	22.17
	LI-FGSM	**94.06**	**91.44**	**99.84***	**86.06**	**61.52**	**50.52**	**42.66**
Res-101	MI-FGSM	58.38	51.12	49.12	99.20*	24.56	22.12	12.24
	NI-FGSM	64.48	58.90	57.00	99.36*	23.30	20.64	11.72
	VMI-FGSM	79.36	73.28	73.32	99.26*	50.16	44.44	34.58
	VNI-FGSM	79.20	75.06	73.02	99.58*	45.88	42.88	31.98
	EMI-FGSM	82.20	76.37	76.00	**100.00***	35.83	31.43	19.13
	LI-FGSM	**90.90**	**88.78**	**88.00**	99.42*	**59.58**	**51.86**	**38.82**

Table 2. Attack success rates (%) against seven models. The adversarial examples are generated by Inc-v3, Inc-v4, IncRes-v2, and Res-101 respectively using MI-FGSM, NI-FSGSM, VMI-FGSM, VNI-FGSM, EMI-FGSM, and LI-FGSM. * indicates that the target model is the same as the source model.

Model	Attack	HGD	R&P	NIPS-r3	FD	ComDefend	RS	NRP
Inc-v3	MI-CT-FGSM	57.27	46.63	54.33	73.43	70.73	33.00	39.90
	VMI-CT-FGSM	71.50	63.60	69.67	78.80	79.43	37.90	56.73
	EMI-CT-FGSM	67.47	58.47	67.37	82.33	82.53	37.93	49.97
	LI-CT-FGSM	**88.60**	**79.97**	**85.17**	**86.90**	**92.03**	**39.97**	**65.17**
	MI-Admix-CT-FGSM	66.00	54.23	63.50	77.50	78.10	34.83	47.50
	MI-SCM-P-CT-FGSM	65.20	54.43	63.13	76.90	76.07	34.33	45.10
	MI-SCM-R-CT-FGSM	**67.87**	**59.67**	**68.10**	**80.13**	**81.27**	**38.87**	**52.00**

Table 3. Attack success rates (%) against seven defense models. The adversarial examples are generated by Inc-v3 using MI-CT-FSGM, EMI-CT-FGSM, LI-CT-FGSM, MI-Admix-CT-FGSM, and MI-SCM-P(R)-FGSM, respectively. CT denotes the combination of DIM, TIM, and SIM.



(a) (b)

(c) (d)

Figure 1. **(a)**: The normalized cosine similarity according to the increase of iterations on six gradient-based attack methods. The source model is Inc-v3 and the cosine similarity is averaged over the first 10 samples of the dataset. **(b)**: Attack success rates (%) according to varying the number of steps looking ahead on six models except for Inc-v3, which is the source model. **(c), (d)**: Attack success rates (%) according to varying the patch sizes of CutMix (dashed lines) and SCM (solid lines) against four normally and three adversarially trained models, respectively. Inc-v3 is used as the source model and SCM refers to SCM-P.

shows that preserving the important features of the input and increasing variance of input distribution help to craft transferable adversaries. Therefore, we believe that applying the transformation that preserves the information of the input image is necessary for improving the transferability.

5. Conclusion and Future Work

In this paper, we introduce two methods for boosting the transferability in adversarial attacks, namely Lookahead Iterative Fast Gradient Sign Method (LI-FGSM) and Self-CutMix (SCM). LI-FGSM looks ahead during N steps to explore the optimal direction in advance of updates. Also, SCM utilizes the input transformation that copies the patch from the original image and pastes it back at the same one. They acquires state-of-the-art attack success rates not only against normally trained models but also against adversarial training and defense models. Although methods developed in this work could serve as a potential tool to fool machine learning models, we hope that our methods will be used as a practical algorithm to create robust defense models.

Acknowledgment This work was supported by the Engineering Research Center of Excellence (ERC) Program supported by National Research Foundation (NRF), Korean Ministry of Science & ICT (MSIT) (Grant No. NRF-2017R1A5A1014708).

References

[1] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017. 1

[2] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019. 4

[3] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 1, 2, 4

[4] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019. 1, 2, 4

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[7] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. Comdefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6084–6092, 2019. 4

[8] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016. 1, 2

[9] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1787, 2018. 4

[10] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations*, 2020. 1, 2, 4

[11] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 860–868. IEEE, 2019. 4

[12] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 1

[13] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 262–271, 2020. 4

[14] Y. E. NESTEROV. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983. 2

[15] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 12(1):145–151, 1999. 2

[16] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 2, 4

[17] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4278–4284. AAAI Press, 2017. 4

[18] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 4

[19] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1

[20] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017. 4

[21] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. *arXiv preprint arXiv:2103.15571*, 2021. 2, 3, 4

[22] Xiaosen Wang, Xuanran He, Jingdong Wang, and Kun He. Admix: Enhancing the transferability of adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16158–16167, 2021. 2, 4

[23] Xiaosen Wang, Jiadong Lin, Han Hu, Jingdong Wang, and Kun He. Boosting adversarial transferability through enhanced momentum. *CoRR*, abs/2103.10609, 2021. 2, 4

[24] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017. 4

[25] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019. 1, 2, 4

[26] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 1

[27] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

[28] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back.

In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 1