RESEARCH

Open Access

Maxwell's Demon in MLP-Mixer: towards transferable adversarial attacks

Haoran Lyu[1], Yajie Wang[1], Yu-an Tan[1], Huipeng Zhou[2], Yuhang Zhao[1] and Quanxin Zhang[2*]

Abstract

Models based on MLP-Mixer architecture are becoming popular, but they still suffer from adversarial examples. Although it has been shown that MLP-Mixer is more robust to adversarial attacks compared to convolutional neural networks (CNNs), there has been no research on adversarial attacks tailored to its architecture. In this paper, we fill this gap. We propose a dedicated attack framework called Maxwell's demon Attack (MA). Specifically, we break the channel-mixing and token-mixing mechanisms of the MLP-Mixer by perturbing inputs of each Mixer layer to achieve high transferability. We demonstrate that disrupting the MLP-Mixer's capture of the main information of images by masking its inputs can generate adversarial examples with cross-architectural transferability. Extensive evaluations show the effectiveness and superior performance of MA. Perturbations generated based on masked inputs obtain a higher success rate of black-box attacks than existing transfer attacks. Moreover, our approach can be easily combined with existing methods to improve the transferability both within MLP-Mixer based models and to models with different architectures. We achieve up to 55.9% attack performance improvement. Our work exploits the true generalization potential of the MLP-Mixer adversarial space and helps make it more robust for future deployments.

Keywords Adversarial attacks, Adversarial examples, Image classification

Introduction

Convolutional Neural Networks (CNNs) have become the de facto standard in the field of computer vision. Deep Neural Networks (DNNs) based on CNNs continue to improve classification performance in computer vision, such as Densenet (Huang et al. 2017), MobileNet (Sandler et al. 2018), EfficientNet (Tan and Le 2019), ReXNet (Han et al. 2021). However, with the development of attention-based transformers in the field of natural language processing, some new models applying this transformer structure have emerged, such as ViT (Dosovitskiy et al. 2020), T2T-ViT (Yuan et al. 2021) and DeiT (Touvron et al. 2021). The performance of these models has caught up with CNNs and is challenging the position of CNNs in the field of computer vision. With further research, researchers found that convolution and attention mechanisms are not unique to good performance, and only using MultiLayer Perceptrons (MLPs) can also achieve good performance, so MLP-Mixer (Tolstikhin et al. 2021) is proposed.

As we all know, DNNs have security risks and are vulnerable to adversarial examples. The adversary adds a well-designed and imperceptible perturbations to the clean input, leading DNNs to incorrect results. Due to the potential risks of DNNs, it is very important to understand whether the recently proposed ViTs and MLP-Mixer are vulnerable to adversarial attacks. The adversarial transferability of ViTs has been well studied (Naseer et al. 2021). In contrast, MLP-Mixer has not been carefully studied in the context of black-box adversarial, and there is no research on the transferability of adversarial attacks against MLP-Mixer. In this work, we focus specifically on transfer-based adversarial attacks

*Correspondence:
Quanxin Zhang
zhangqx@bit.edu.cn
[1] School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China
[2] School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China

and study how to improve the transferability of adversarial examples generated by MLP-Mixer.

Our analysis of MLP-Mixer is based on the following findings. MLP-Mixer differs in architecture from CNNs. Similar to ViTs, MLP-Mixer uses image patches as input, but does not use any convolution and attention mechanisms. Instead, the architecture of MLP-Mixer is entirely based on MLPs. MLP-Mixer contains two types of layers, one is mixing spatial location information, called token-mixing MLPs, and the other is mixing channel information, called channel-mixing MLPs. Information from different patches and channels can be fully mixed, enabling MLP-Mixer to capture the main information of the image. Disturbing the information mixing mechanism of MLP-Mixer can prevent the generated adversarial examples from overfitting the source models, which can improve the cross-architecture transferability of adversarial examples.

We propose a novel adversarial attack called Max-well's demon Attack (MA). The channel-mixing MLPs allow communication between different channels, and the token-mixing MLPs allow communication between different spatial locations. By randomly masking the inputs of each Mixer layer of MLP-Mixer, we break the channel-mixing and token-mixing mechanisms of MLP-Mixer, making it impossible for different locations and different channels to communicate normally, which makes MLP-Mixer unable to capture the main information of the picture. This achieves an effect similar to Dropout (Srivastava et al. 2014), prevents the generated adversarial examples from overfitting the MLP-Mixer, and improves the fooling rate of the adversarial examples attacking the target models. As shown in Fig. 1, our method can further force the target model to focus on the wrong regions in the adversarial examples compared to the original method.

Our proposed MA method is a detachable component that can be easily combined with existing methods. We conduct extensive experiments on models with multiple architectures using the ImageNet validation set. The adversarial examples generated by our method can improve the fool rate by 55.9%. Our work opens new perspectives for exploring the vulnerabilities of MLP-Mixer and explaining transfer attacks across architectures. It provides insights for enhancing the robustness of MLP-Mixer to fuel its future deployments.

In summary, our main contributions are as follows:

- We analyze the channel-mixing and token-mixing mechanism in MLP-Mixer and generate adversarial examples with high transferability by breaking them. The proposed MA is applicable to any model based on the MLP-Mixer architecture.

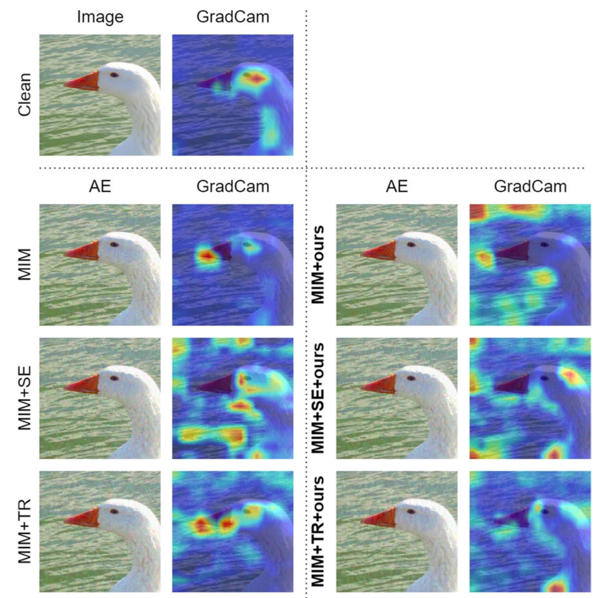

Fig. 1 Demonstration of the clean image, adversarial examples (AE) made by existing adversarial attack methods and our mothods on MLP-Mixer, and their GradCam images generated on VGG-16. The adversarial examples generated by the original method still make the target model focus on the object itself. Our method can force the target model to pay more attention to the regions far from the object in the adversarial examples

- Our approach can be combined with existing attacks and significantly improves the performance, bridging the gap that existing attacks cannot execute the cross-architecture attack. Our results demonstrate the feasibility of cross-architecture black-box attacks.
- We conducted transfer attack experiments using 2 different white-box MLP-Mixers against 7 blackbox MLPs, 10 black-box ViTs, and 39 CNNs. In extensive experimental evaluations, our methods all exhibit optimal performance.

Related work

Adversarial attack and transferability

Adversarial attacks are divided into white-box attacks and black-box attacks. The white-box attacks require access to all information about the target model, such as FGSM (Goodfellow et al. 2014) and PGD (Madry et al. 2017). The black-box attacks do not need to know the target model information, and the mainstream approach is the transfer-based attack. The transfer-based attacks require an alternative model that is similar to the target model, and white-box attacks on the alternative model to generate adversarial examples. The target model is attacked by virtue of the transferability of the adversarial examples, and thus the goal of transfer-based attacks is to improve the transferability of the adversarial examples.

The MIM (Dong et al. 2018) enhances the transferability of FGSM by adding a momentum term to the gradient. The DIM (Xie et al. 2019) improves the transferability of adversarial examples by creating different input modes. The TIM (Dong et al. 2019) improves transferability by using a predefined kernel convolution on the gradient. Our method can be easily combined with these existing methods to further improve the transferability of adversarial examples.

Robustness of new architectures

For the robustness of the new models, we mainly focus on the robustness of ViTs and MLP-Mixer. Benz et al. (2021) investigated the adversarial robustness of ViTs and MLP-Mixer. They found that MLP-Mixer is vulnerable to universal adversarial perturbations. But they did not explore the adversarial transferability of MLP-Mixer. To our knowledge, there is currently no work investigating the transferability of adversarial examples generated by MLP-Mixer. Naseer et al. (2021) introduced two strategies to enhance the transferability of adversarial examples generated by ViTs. One is to obtain the output of each ViT block to generate adversarial examples, called Self-Ensemble, and the other is to train a classifier head for each ViT block and use the output of each classifier head to generate adversarial examples, called Token Refinement. We try to introduce these two strategies to MLP-Mixer, but the effect is not significant. After our modification and the introduction of our proposed MA, the transferability of the adversarial examples generated by MLP-Mixer is substantially improved.

Methodology

Consider a clean image sample $x \in X$ and its ground-truth label $y \in Y$, a source model $\mathcal{F}(x) : X \to Y$ and a target model $\mathcal{M}$ which is under-attack. We focus on untargeted adversarial attack, the goal of the transfer-based attack is generating an adversarial example x_{adv}, using the information of source model $\mathcal{F}$, which can change the target model's prediction ($\mathcal{M}(x_{adv}) \neq y$). In order to make the adversarial example imperceptible to the human eye, it is necessary to limit the modification magnitude of the adversarial example relative to the clean sample, and we use the l_∞ for the restriction, i.e., $\|x_{adv} - x\|_\infty < \epsilon$. The optimization problem of generating the adversarial example is defined as follows:

$$\underset{x_{adv}-x}{\arg\max}\, J(\mathcal{F}(x_{adv}), y), s.t. \|x_{adv} - x\|_\infty < \epsilon \tag{1}$$

where $J(\cdot,\cdot)$ is the loss function (e.g. cross-entropy).

For the MLP-Mixer model $\mathcal{F}$ with n Mixer layers can be defined as:

$$\mathcal{F} = (l_1 \circ l_2 \circ l_3 \circ \ldots \circ l_n) \circ f_c \tag{2}$$

where l_i represents a single Mixer layer comprising of token-mixing layer and channel-mixing layer and f_c is the final classification head.

Our MA method is able to control the input of each Mixer layer. We multiply the input I of each Mixer layer of MLP-Mixer by a masking matrix M, which can be defined as follows:

$$\begin{aligned} &\mathcal{F} = (l_1(I_1 \odot M_1) \circ l_2(I_2 \odot M_2) \circ l_3(I_3 \odot M_3) \circ \ldots \circ l_n(I_n \odot M_n)) \circ f_c \\ &\text{where } M_i = \begin{cases} X \sim B(1,p) & P \\ 1 & 1-P \end{cases} \end{aligned} \tag{3}$$

$\odot$ represents element-wise product. In the case of probability P, we mask the input of each Mixer layer, and M is the matrix directly generated from the Bernoulli distribution.

As shown in Fig. 2, our MA method controls the input of each Mixer layer, and by masking part of the input, we destroy the channel-mixing and token-mixing mechanism of MLP-Mixer, thereby improving the transferability of adversarial examples against MLP-based models. Meanwhile, by masking the input of each Mixer layer of MLP-Mixer, our method achieves a Dropout-like effect. But unlike Dropout dropping neurons, our method is to drop the feature maps of each layer, which can prevent adversarial examples from overfitting the source model MLP-Mixer, thereby improving the transferability of adversarial examples to non-MLP models, such as CNNs and ViTs.

Our method benefits from the fact that each Mixer layer structure of MLP-Mixer is the same, so it only needs to generate a masking matrix of one size, thus saving computational overhead, which is not possible in most CNNs. Our method is a detachable component that can be easily combined with existing gradient-based methods, such as PGD, MIM, DIM, TIM.

Our method can also be combined with the Self-Ensemble (SE) method and Token Refinement (TR) method that attack ViTs. SE and TR methods are also components that can be combined with gradient-based methods. SE obtains the output of each block of ViTs,

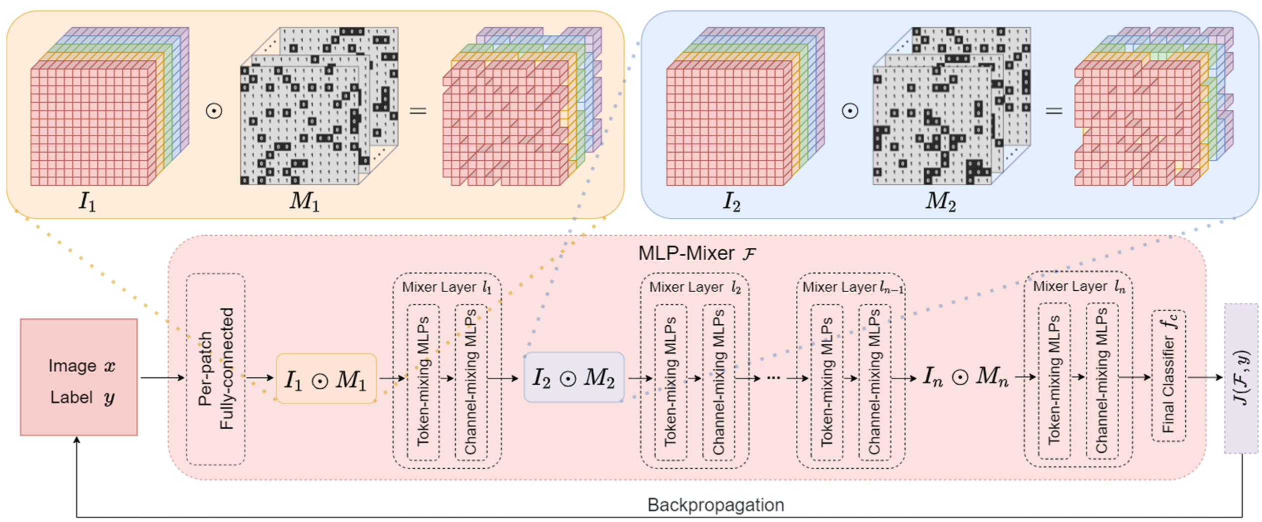

Fig. 2 Maxwell's demon Attack (MA). I_i represents the input of the Mixer layer l_i, M_i is the masking matrix generated for I_i based on the Bernoulli distribution, and $\odot$ represents the element-wise product. After I_i and M_i are multiplied, some elements in I_i are discarded. By masking the part input of each Mixer layer, MA breaks the token-mixing and channel-mixing of MLP-Mixer, preventing adversarial examples from overfitting MLP-Mixer, which can greatly improve the transferability of its generated adversarial examples

and then inputs it into the final classifier head respectively. After obtaining all the outputs of the classifier head, SE calculates the average of the outputs as the input of the loss function. We transplant SE into MLP-Mixer and combined them with our MA. For the SE combined with MA method, it can be defined as follows:

$$\mathcal{F}_k = \left(\prod_{i=1}^{k} l_i(I_i \odot M_i)\right) \circ f_c, \text{ where } k = 1, 2, \ldots, n \tag{4}$$

$$\mathcal{F} = \frac{1}{n}\sum_{k=1}^{n} F_k \tag{5}$$

Specifically, as shown in Algorithm 1, when combined with the SE method, the inputs of each layer of the Mixer Layer are masked and input to the final classifier head respectively, thus generating a Self-Ensemble of n MLP-Mixer networks with different depths. Finally, obtain all the outputs of the classifier head, calculate their average, and perform backpropagation to obtain the updated gradient of the adversarial example.

TR trains a classifier head for each block of ViTs, uses the output of each classifier head to calculate the loss value, and then averages all the loss values as the final loss value. We also transplant TR into MLP-Mixer and combined them with our MA. For the TR combined with MA method, it can be defined as follows:

$$\mathcal{F}_k = \left(\prod_{i=1}^{k} l_i(I_i \odot M_i)\right) \circ f_c^k, \text{ where } k = 1, 2, \ldots, n \tag{6}$$

$$J = \frac{1}{n}\sum_{k=1}^{n} J_k(\mathcal{F}_k(x_{adv}), y) \tag{7}$$

where f_c^k is the classifier head we trained for each Mixer layers. The algorithm combining MA with SE and TR is shown in Algorithm 1.

Algorithm 1 Maxwell's Demon attack

Input: A source MLP-Mixer model $\mathcal{F}$ composed of n Mixer layers l and the final classifier head f_c ; a loss function J; a clean image x with ground-truth label y;
Parameter: The perturbation budget of l_∞-normed ϵ; iterations T; step size α and decay facotr μ. The masking matrix M generated for each Mixer layer l defined by Eq. 3; the input I of each Mixer layer l
Output: An adversarial example x_{adv} with $\|x_{adv} - x\|_\infty \leq \epsilon$

$x^0_{adv} = x$; $g_0 = 0$
for $t = 0$ to $T - 1$ **do**
  $I_0 =$ Per-Patch Fully-Connected (x^t_{adv})
  $output = 0, grad = 0$
  for each Mixer layer l_i in $\mathcal{F}$, $i = 0$ to $i < n$ **do**
    $I_{i+1} = l_i(I_i \odot M_i)$
    if combined SE **then**
      $output+ = f_c(I_{i+1})$
    else if combined TR **then**
      $output = f_c^{i+1}(I_{i+1})$
      $grad+ = \frac{\nabla_x J(output,y)}{\|\nabla_x J(output,y)\|_1}$
    else
      $output = f_c(I_{i+1})$
    end if
  end for
  if combined SE **then**
    $g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(output/n,y)}{\|\nabla_x J(output/n,y)\|_1}$
  else if combined TR **then**
    $g_{t+1} = \mu \cdot g_t + grad/n$
  else
    $g_{t+1} = \mu \cdot g_t + \frac{\nabla_x J(output,y)}{\|\nabla_x J(output,y)\|_1}$
  end if
  $x^{t+1}_{adv} = clip_{x,\epsilon}(x^t_{adv} + \alpha \cdot sign(g_{t+1}))$
end for
return $x_{adv} = x^T_{adv}$

Experiments

In this section, experimental results of the proposed method are presented. The experimental settings are introduced in "Settings" section, the experimental results of attacking different architecture models are introduced in "Improve transferability to MLP-based models"–"Attack against defense approaches" sections, and the parameter selection and ablation experiments are introduced in "Effect of probability values" and "Ablation study" sections respectively.

Settings

We choose Mixer-B/16 and Mixer-L/16 in MLP-Mixer as source models. For the target model, we report experimental results on the following models, VGG (VGG-13, VGG-16, VGG-19) (Simonyan and Zisserman 2014), ResNet (Resnet-18, Resnet-34, Resnet-50, Resnet-101, Resnet-152) (He et al. 2016), DenseNet (Densenet-121, Densenet-161, Densenet-169, Densenet-201) (Huang et al. 2017), ReXNet (ReXNetV1-10, ReX-NetV1-13, ReXNetV1-15, ReXNetV1-20, ReXNetV1-30) (Han et al. 2021) and MobileNet-V2 based on the CNN architecture, ResNet152-denoise (Xie et al. 2019), ResNet50-FreeAT (Shafahi et al. 2019), ResNet50-FastAT (Wong et al. 2020) and EfficientNet (AdvEfficient-Net-b0, AdvEfficientNet-b1, AdvEfficientNet-b2) (Tan and Le 2019) after adversarial training, ViT-B/16 (Dosovitskiy et al. 2020), T2T-ViT (T2T-12, T2T-14, T2T-19) (Yuan et al. 2021) and DeiT-B (Touvron et al. 2021) based on the transformer architecture, and ResMLP-36 (Touvron et al. 2021) based on the MLP architecture. These models are provided by TIMM (Wightman 2019), the experimental results of more models can be found in the supplementary material. We randomly selected 1k samples from the ImageNet validation set, and these samples can be correctly classified by all the above models. We use the fooling rate to assess the transferability of adversarial examples, i.e. the percentage of adversarial examples whose predicted labels on the target model are inconsistent with ground-truth labels. We uniformly set the perturbation budget ϵ to 16, and the number of attack iterations T to 50.

Improve transferability to MLP-based models

In this section, we discuss the experimental results of adversarial transferability between MLP-Mixer and black-box MLP-based models. As shown in the first two columns of Table 1, we report the experimental results of white-box attack and black-box attack on the MLP-Mixer models. Since our method prevents adversarial examples from overfitting the source model, the fooling rate may drop slightly during white-box attacks. But when attacking models of the same architecture, our method can greatly improve the fooling rate. After DIM combined with our method, Mixer-L/16 generated adversarial examples that can improve the fooling rate by 55.9% on Mixer-B/16.

As shown in the third column of Table 1, we report the experimental results of the MLP-based ResMLP-36 as target model. For the basic adversarial attack methods PGD, MIM, DIM and TIM, combined with our method, the adversarial examples generated by Mixer-B/16 can improve the fooling rate on ResMLP-36 by about 20%. After DIM is combined with our method, Mixer-L/16 generated adversarial examples can improve the fooling rate by 38.0% on ResMLP-36. The SE and TR methods combined with our method can further improve the transferability of adversarial examples on ResMLP-36. Experimental results demonstrate that our method is able to break the channel-mixing and token-mixing mechanisms of MLP-Mixer by masking the input of each Mixer layer and improving the transferability of adversarial examples on the MLP-based models.

Table 1 The fooling rate (%) on 1k ImageNet val

Attack	Mixer-B/16	Mixer-L/16	ResMLP-36	ViT-B/16	Deit-B	VGG-16	ResNet-50	MoNet-V2
Source Model: Mixer-B/16								
PGD	100.0	35.2	18.5	9.6	7.6	5.4	2.6	6.7
PGD+MA	99.7 (– 0.3)	77.3 (+ 42.1)	33.7 (+ 15.2)	11.1 (+ 1.5)	13.8 (+ 6.2)	27.7 (+ 22.3)	14.6 (+ 12.0)	30.7 (+ 24.0)
PGD+SE	100.0	88.6	50.6	18.0	23.9	31.0	16.4	35.7
PGD+SE+MA	99.9 (– 0.1)	86.1 (– 2.5)	55.7 (+ 5.1)	27.8 (+ 9.8)	35.1 (+ 11.2)	58.8 (+ 27.8)	32.8 (+ 16.4)	56.2 (+ 20.5)
PGD+TR	100.0	86.9	59.4	23.6	30.2	55.1	22.8	48.6
PGD+TR+MA	99.8 (– 0.2)	92.3 (+ 5.4)	76.0 (+ 16.6)	52.6 (+ 29.0)	59.6 (+ 29.4)	78.6 (+ 23.5)	44.4 (+ 21.6)	71.5 (+ 22.9)
MIM	100.0	47.3	21.5	8.7	12.3	21.3	9.60	23.6
MIM+MA	100.0 (+ 0.0)	88.2 (+ 40.9)	52.2 (+ 30.7)	22.0 (+ 13.3)	38.7 (+ 26.4)	41.6 (+ 20.3)	22.6 (+ 13.0)	43.9 (+ 20.3)
MIM+SE	100.0	89.3	55.7	27.7	37.6	52.3	28.2	50.0
MIM+SE+MA	100.0 (+ 0.0)	91.3 (+ 2.0)	63.9 (+ 8.2)	32.6 (+ 4.9)	46.2 (+ 8.6)	62.5 (+ 10.2)	35.2 (+ 7.00)	61.7 (+ 11.7)
MIM+TR	100.0	86.0	60.7	33.1	38.5	63.4	31.1	59.6
MIM+TR+MA	100.0 (+ 0.0)	91.2 (+ 5.2)	73.3 (+ 12.6)	50.0 (+ 16.9)	53.3 (+ 14.8)	75.5 (+ 12.1)	42.5 (+ 11.4)	69.9 (+ 10.3)
DIM	100.0	58.7	26.1	7.1	7.7	8.9	6.1	13.9
DIM+MA	100.0 (+ 0.0)	92.6 (+ 33.9)	57.8 (+ 31.7)	16.7 (+ 9.6)	22.5 (+ 14.8)	31.4 (+ 22.5)	18.0 (+ 11.9)	36.6 (+ 22.7)
DIM+SE	100.0	97.0	77.9	33.8	45.1	50.4	33.6	56.2
DIM+SE+MA	100.0 (+ 0.0)	96.6 (– 0.4)	82.6 (+ 4.7)	43.1 (+ 9.3)	54.9 (+ 9.8)	69.8 (+ 19.4)	42.8 (+ 9.2)	69.1 (+ 12.9)
DIM+TR	100.0	95.4	83.1	40.9	55.7	69.7	41.3	67.3
DIM+TR+MA	100.0 (+ 0.0)	96.0 (+ 0.6)	86.9 (+ 3.8)	61.6 (+ 20.7)	70.9 (+ 15.2)	81.6 (+ 11.9)	54.5 (+ 13.2)	78.3 (+ 11.0)
TIM	100.0	35.3	9.2	2.8	2.3	5.6	2.5	6.8
TIM+MA	99.8 (– 0.2)	77.4 (+ 42.1)	33.8 (+ 24.6)	11.5 (+ 8.7)	15.2 (+ 12.9)	27.5 (+ 21.9)	13.8 (+ 11.3)	30.6 (+ 23.8)
TIM+SE	100.0	88.8	52.8	17.9	25.2	31.9	17.4	36.1
TIM+SE+MA	99.8 (– 0.2)	86.3 (– 2.5)	56.8 (+ 4.0)	27.9 (+ 10.0)	35.3 (+ 10.1)	60.0 (+ 28.1)	34.0 (+ 16.6)	56.4 (+ 20.3)
TIM+TR	100.0	87.2	59.8	23.9	29.8	54.1	22.6	48.8
TIM+TR+MA	99.8 (– 0.2)	93.2 (+ 6.0)	77.4 (+ 17.6)	53.2 (+ 29.3)	59.4 (+ 29.6)	80.3 (+ 26.2)	45.7 (+ 23.1)	71.6 (+ 22.8)
Source Model: Mixer-L/16								
PGD	21.0	100.0	2.5	1.3	0.3	3.4	1.5	4.3
PGD+MA	38.7 (+ 17.7)	99.1 (– 0.9)	7.6 (+ 5.1)	2.1 (+ 0.8)	1.8 (+ 1.5)	10.8 (+ 7.4)	5.0 (+ 3.5)	12.6 (+ 8.3)
PGD+SE	88.2	100.0	38.5	14.5	16.0	28.2	14.3	27.9
PGD+SE+MA	83.4 (– 4.8)	99.6 (– 0.4)	47.1 (+ 8.6)	23.1 (+ 8.6)	26.0 (+ 10.0)	47.1 (+ 18.9)	24.2 (+ 9.9)	44.4 (+ 16.5)
PGD+TR	93.0	100.0	66.7	32.4	42.6	58.0	28.7	53.9
PGD+TR+MA	94.0 (+ 1.0)	99.8 (– 0.2)	79.1 (+ 12.4)	56.7 (+ 24.3)	60.3 (+ 17.7)	78.8 (+ 20.8)	47.7 (+ 19.0)	73.5 (+ 19.6)
MIM	31.9	100.0	8.5	4.9	2.8	16.9	7.0	16.4
MIM+MA	51.8 (+ 19.9)	99.4 (– 0.6)	13.6 (+ 5.1)	6.2 (+ 1.3)	5.9 (+ 3.1)	26.5 (+ 9.6)	12.4 (+ 5.4)	26.9 (+ 10.5)
MIM+SE	89.0	100.0	46.3	23.5	23.7	46.6	23.9	45.7
MIM+SE+MA	85.6 (– 3.4)	99.6 (– 0.4)	50.2 (+ 3.9)	26.9 (+ 3.4)	30.2 (+ 6.5)	56.1 (+ 9.5)	30.6 (+ 6.7)	51.2 (+ 10.5)
MIM+TR	91.9	100.0	70.2	42.7	47.4	67.7	36.3	64.1
MIM+TR+MA	94.4 (+ 2.5)	99.8 (– 0.2)	79.3 (+ 9.3)	59.1 (+ 16.4)	63.6 (+ 16.2)	81.8 (+ 14.1)	51.9 (+ 15.6)	77.5 (+ 13.4)
DIM	36.1	100.0	6.9	2.5	2.1	5.4	3.7	7.4
DIM+MA	92.0 (+ 55.9)	100.0 (+ 0.0)	44.9 (+ 38.0)	11.8 (+ 9.3)	15.1 (+ 13.0)	18.8 (+ 13.4)	11.4 (+ 7.7)	25.8 (+ 18.4)
DIM+SE	95.9	100.0	65.5	28.5	36.3	46.4	27.2	47.9
DIM+SE+MA	98.3 (+ 2.4)	100.0 (+ 0.0)	80.1 (+ 14.6)	40.6 (+ 12.1)	53.2 (+ 16.9)	62.9 (+ 16.5)	40.1 (+ 12.9)	62.4 (+ 14.5)
DIM+TR	96.1	100.0	82.0	46.9	60.4	71.0	41.9	69.1
DIM+TR+MA	97.9 (+ 1.8)	100.0 (+ 0.0)	89.0 (+ 7.0)	66.5 (+ 19.6)	74.0 (+ 13.6)	83.5 (+ 12.5)	57.8 (+ 15.9)	79.6 (+ 10.5)
TIM	21.4	100.0	2.9	1.2	0.4	3.3	2.0	4.4
TIM+MA	38.7 (+ 17.3)	98.8 (– 1.2)	7.8 (+ 4.9)	3.0 (+ 1.8)	1.6 (+ 1.2)	10.9 (+ 7.6)	4.9 (+ 2.9)	12.6 (+ 8.2)
TIM+SE	88.3	100.0	39.1	14.6	16.3	29.0	14.8	29.9
TIM+SE+MA	82.8 (– 5.5)	99.6 (– 0.4)	47.3 (+ 8.2)	23.2 (+ 8.6)	25.6 (+ 9.3)	47.9 (+ 18.9)	24.6 (+ 9.8)	44.6 (+ 14.7)
TIM+TR	92.3	100.0	67.8	34.1	42.7	58.2	28.9	54.0
TIM+TR+MA	93.7 (+ 1.4)	99.8 (– 0.2)	78.6 (+ 10.8)	58.2 (+ 24.1)	60.8 (+ 18.1)	79.0 (+ 20.8)	48.7 (+ 19.8)	74.9 (+ 20.9)

The adversarial examples at $\epsilon \leq 16$. The adversarial examples generated by our proposed MA method have a significantly higher fooling rate. We generated adversarial examples on Mixer-B/16 and Mixer-L/16, and conducted transferability experiments on networks of different architectures. Our MA method combined with existing adversarial attack methods can comprehensively improve the transferability of adversarial examples on MLP-based models, transformer-based models and CNN-based models

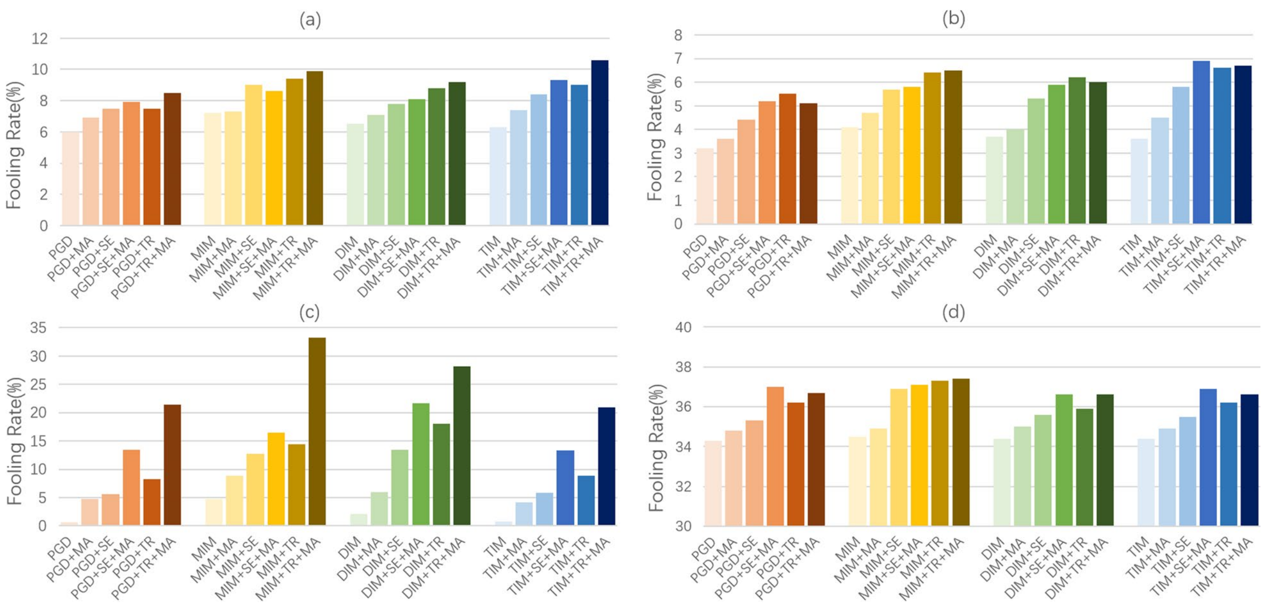

Fig. 3 The attack against Defense Approaches. The fooling rate (%) on 1k ImageNet val. The adversarial examples at $\epsilon \leq 16$. The fooling rate of the original SE and TR methods combined with PGD, MIM, DIM and TIM, and the fooling rate of these methods combined with our method. The source model is Mixer-B/16. The target models are **a** ResNet50_FastAT, **b** ResNet50_FreeAT, **c** AdvEfficientNet-b0 and **d** ResNet152_denoise. Our method can further overcome adversarial defense methods such as FastAT, FreeAT, adversarial training and denoise

Improve transferability to ViTs

As shown in the last three columns of Table 1, we report the experimental results of CNN-based VGG-16, ResNet-50 and MobileNet-V2 as target models. PGD, MIM, DIM and TIM, when combined with our method, can improve the fooling rate of adversarial examples generated by Mixer-B/16 on VGG-16 by more than 20%.

It is worth noting that, the SE and TR method combined with our method improves the transferability by about 20% compared to the original method on VGG-16, ResNet-50 and MobileNet-V2. Although SE and TR are not designed for CNNs, the fooling rate of generated adversarial examples attacking CNNs is further improved after SE and TR are combined with our method. As shown in Fig. 1, we show adversarial examples generated on MLP-B16 and GradCam (Gildenblat 2021) images generated on VGG-16, our method can further force the target model to focus on the wrong regions in the adversarial examples compared to the original method. This fully demonstrates the effectiveness and great potential of our method.

Attack against defense approaches

As shown in Fig. 3, the results on the ResNet50-FastAT, ResNet50-FreeAT, AdvEfficientNet-b0 and ResNet152-denoise models after adversarial training demonstrate that our method can further overcome adversarial defense methods such as FastAT, FreeAT, adversarial training and denoise. Our attack method is not only effective for ordinary CNNs, but also for robust models. PGD, MIM, DIM and TIM, as well as SE and TR combined with our method can further improve the fooling rate of adversarial examples on the robust models.

The experimental results demonstrate that our MA method combined with existing adversarial attack methods can comprehensively improve the transferability of adversarial examples on MLP-based models, transformer-based models, CNN-based models and robust models. This means that using only our method, generating adversarial examples on a single model transfers well on MLP-based models, CNN-based models and transformer-based models. Our method achieves the effect of using an ensemble model on a single model, but uses fewer resources and is faster.

Effect of probability values

As shown in Eq. 3, there are two probability values that need to be set in our method, one is the probability P of whether to mask the input and the other is the probability p of generating the masking matrix using the Bernoulli distribution. We report the mean fooling rate of adjusted probability on each kind of models, the source model is Mixer-B/16, and the attack method is a combination of MIM, TR and our MA method. We first test the probability P of whether to mask the input. We randomly set the probability p of the Bernoulli distribution to 0.8.

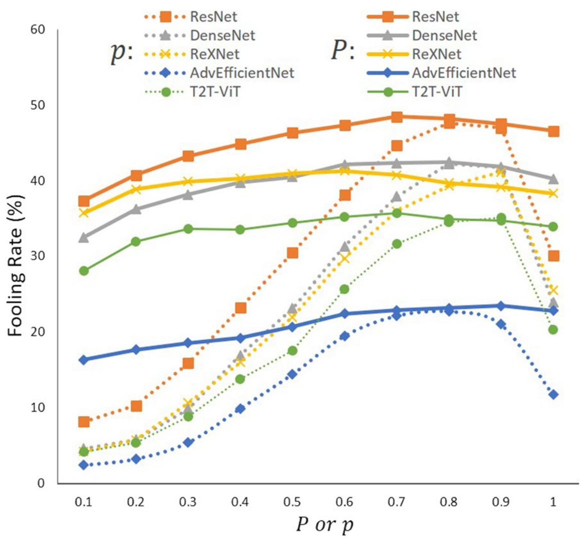

Fig. 4 The experimental results of selecting the probability values. The solid lines are the experimental results of the probability P of whether to mask the input. The dotted lines are the experimental results of the probability p of Bernoulli distribution

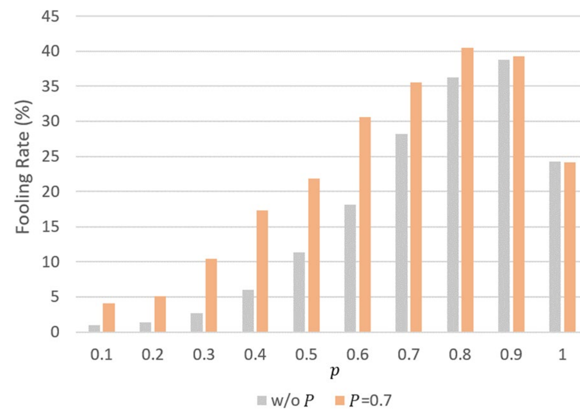

Fig. 5 Ablation experiment. Compare the experimental results of deleting the probability P of whether to mask the input and setting it to 0.7. The horizontal axis is the value of the probability p of the Bernoulli distribution

As shown by the solid line in Fig. 4, when P is 0.7, the fooling rate of generated adversarial examples reaches the maximum value on multiple models. Then we test the probability p of the Bernoulli distribution, setting the probability P to 0.7. As shown by the dotted line in Fig. 4, the fooling rate of the generated adversarial examples on multiple models first rises and then declines. When p is 0.8, the fooling rate reaches the maximum value. So we set P to 0.7 and p to 0.8.

Ablation study

We perform ablation experiments on our method. The attack method is a combination of MIM, TR and our method. The source model is Mixer-B/16, and the target model DenseNet-201. The experimental results are shown in Fig. 5. Deleting the probability P of whether to mask the channel, the fooling rate is decreased compared to setting P to 0.7. Although the fooling rate is flat after the probability p of Bernoulli distribution is greater than 0.9, it does not exceed the maximum value at $P = 0.7$, which proves that every part of our method contributes.

Conclusion

We propose a novel transfer-based attack, called Maxwell's demon Attack (MA). By using MA to mask the part input of each Mixer layer of the MLP-Mixer, we are able to greatly improve the transferability of its generated adversarial examples. On some CNN-based models, the adversarial examples generated by our method on the MLP-Mixer even exceed the transferability of the adversarial examples generated using CNNs. Our method can be simply combined with existing adversarial attack methods against CNNs and ViTs. We conduct experiments on models with multiple architectures, and the experimental results demonstrate the superiority of our method. To our knowledge, we are the first work to study the transferability of MLP-Mixer.

Acknowledgements
There is no any third person/ organisation to acknowledge.

Author contributions
H.L.: Research, Data analysis, Documentation, Reporting, Implementations, Problem formulation, Coding, Testing. Y.W.: Supervision, Management, Validation. All authors read and approved the final manuscript.

Funding
The authors declare that the research doesn't used any funding sources for the work. There are no any funding sources to disclose.

Availability of data and materials
Our data comes from the open source dataset ImageNet at http://image-net.org/.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 8 April 2023 Accepted: 14 November 2023
Published online: 02 March 2024

References
Benz P, Ham S, Zhang C, Karjauv A, Kweon IS (2021) Adversarial robustness comparison of vision transformer and mlp-mixer to cnns. arXiv preprint arXiv:2110.02797
Dong Y, Liao F, Pang T, Su H, Zhu J, Hu X, Li J (2018) Boosting adversarial attacks with momentum. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 9185–9193

Dong Y, Pang T, Su H, Zhu J (2019) Evading defenses to transferable adversarial examples by translation-invariant attacks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4312–4321
Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, et al (2020) An image is worth 16 x 16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929
Gildenblat J (2021) contributors: PyTorch library for CAM methods. GitHub
Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572
Han D, Yun S, Heo B, Yoo Y (2021) Rethinking channel dimensions for efficient model design. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 732–741
He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A (2017) Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083
Naseer M, Ranasinghe K, Khan S, Khan FS, Porikli F (2021) On improving adversarial transferability of vision transformers. arXiv preprint arXiv:2106.04169
Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520
Shafahi A, Najibi M, Ghiasi A, Xu Z, Dickerson JP, Studer C, Davis LS, Taylor G, Goldstein T (2019) Adversarial training for free! In: NeurIPS
Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
Tan M, Le Q (2019) Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. PMLR, pp 6105–6114
Tolstikhin I, Houlsby N, Kolesnikov A, Beyer L, Zhai X, Unterthiner T, Yung J, Keysers D, Uszkoreit J, Lucic M, et al (2021) Mlp-mixer: an all-mlp architecture for vision. arXiv preprint arXiv:2105.01601
Touvron H, Bojanowski P, Caron M, Cord M, El-Nouby A, Grave E, Izacard G, Joulin A, Synnaeve G, Verbeek J, et al (2021) Resmlp: feedforward networks for image classification with data-efficient training. arXiv preprint arXiv:2105.03404
Touvron H, Cord M, Douze M, Massa F, Sablayrolles A, Jégou H (2021) Training data-efficient image transformers & distillation through attention. In: International conference on machine learning. PMLR, pp 10347–10357
Wightman R (2019) PyTorch image models. GitHub. https://doi.org/10.5281/zenodo.4414861
Wong E, Rice L, Kolter JZ (2020) Fast is better than free: revisiting adversarial training. ArXiv arXiv:2001.03994
Xie C, Wu Y, van der Maaten L, Yuille AL, He K (2019) Feature denoising for improving adversarial robustness. 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 501–509
Xie C, Zhang Z, Zhou Y, Bai S, Wang J, Ren Z, Yuille AL (2019) Improving transferability of adversarial examples with input diversity. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2730–2739
Yuan L, Chen Y, Wang T, Yu W, Shi Y, Jiang Z, Tay FE, Feng J, Yan S (2021) Tokens-to-token vit: training vision transformers from scratch on imagenet. arXiv preprint arXiv:2101.11986

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.