Improving the Adversarial Transferability of Vision Transformers with Virtual Dense Connection

Jianping Zhang[1], Yizhan Huang[1], Zhuoer Xu[2], Weibin Wu[3*], Michael R. Lyu[1]

[1]Department of Computer Science and Engineering, The Chinese University of Hong Kong
[2]Tiansuan Lab, Antgroup
[3]School of Software Engineering, Sun Yat-sen University
{jpzhang, yzhuang22, lyu}@cse.cuhk.edu.hk, xuzhuoer.xze@antgroup.com, wuwb36@mail.sysu.edu.cn

Abstract

With the great achievement of vision transformers (ViTs), transformer-based approaches have become the new paradigm for solving various computer vision tasks. However, recent research shows that similar to convolutional neural networks (CNNs), ViTs are still vulnerable to adversarial attacks. To explore the shared deficiency of models with different structures, researchers begin to analyze the cross-structure adversarial transferability, which is still under-explored. Therefore, in this work, we focus on the ViT attacks to improve the cross-structure transferability between the transformer-based and convolution-based models. Previous studies fail to thoroughly investigate the influence of the components inside the ViT models on adversarial transferability, leading to inferior performance. To overcome the drawback, we launch a motivating study by linearly down-scaling the gradients of components inside the ViT models to analyze their influence on adversarial transferability. Based on the motivating study, we find that the gradient of the skip connection most influences transferability and believe that back-propagating gradients from deeper blocks can enhance transferability. Therefore, we propose the Virtual Dense Connection method (VDC). Specifically, without changing the forward pass, we first recompose the original network to add virtual dense connections. Then we back-propagate gradients of deeper Attention maps and Multi-layer Perceptron (MLP) blocks via virtual dense connections when generating adversarial samples. Extensive experiments confirm the superiority of our proposed method over the state-of-the-art baselines, with an 8.2% improvement in transferability between ViT models and a 7.2% improvement in cross-structure transferability from ViTs to CNNs.

Introduction

Transformers have become the dominant solutions in the natural language processing field with the state-of-the-art performance on various downstream tasks. Vision transformers (ViTs) (Dosovitskiy et al. 2020) first adapt the self-attention mechanism of the transformers (Vaswani et al. 2017) to the computer vision field for image recognition. Subsequently, diverse transformer-based approaches (Touvron et al. 2021a; Heo et al. 2021) have been proposed to better adapt the transformer structure to the computer vision field. Nowadays, ViTs have become the new paradigm for solving various vision tasks such as object detection (Zhang et al. 2021) and semantic segmentation (Zheng et al. 2021), with competitive performance compared with convolutional neural networks (CNNs).

Recent research reveals that both convolution-based and transformer-based models are vulnerable to adversarial attacks (Wu et al. 2020c; Zhang et al. 2023b). Adversarial attacks inject human-imperceptible noise into the original image to mislead the deep neural network (DNN) models with high confidence. This phenomenon raises security concerns with the wide application of deep neural networks (Zhang et al. 2023c,d; Wu et al. 2019). Furthermore, the adversarial examples crafted by the attacking algorithms manifest adversarial transferability. That is, the adversarial examples generated from a local surrogate model have the ability to mislead the target victim model (Wu et al. 2020b; Zhang et al. 2022). Therefore, adversarial transferability provides an efficient way to craft adversarial examples for testing the victim models without any access to the victim model under the black-box setting. Since victim models are usually deployed in the black-box setting, it is imperative to devise transferable attacking algorithms to assess their robustness before their deployment in real-world applications.

The transfer-based attacks have achieved high attack success rates against convolution-based models. Nevertheless, recent studies have discovered the robustness of the transformer-based models and the low cross-structure transferability, when we transfer the adversarial examples generated by attacking transformer-based models to mislead convolution-based models or vice versa (Zhang et al. 2023b). Some research believes that the low cross-structure transferability is due to the model structure difference between transformer-based models and convolution-based ones (Naseer et al. 2021). Convolution-based models utilize the convolutional layers to capture the local information of the input features in a small receptive field (Luo et al. 2016). Transformer-based models divide the input image into small patches and feed a sequence of small patches into the network. With the help of the self-attention mechanism, ViT models can learn the global features at each stage of the network, which shows distinct properties to CNN models. Therefore, enhancing the adversarial transferability from

*Corresponding author.

transformer-based models to other transformer-based and convolution-based models is of great significance, which facilitates finding the common defects inside transformer-based and convolution-based models in practice.

However, the adversarial transferability of transformer-based models is still under-explored. Although some works have been proposed to improve the adversarial transferability based on the special design of the transformer-based models, they still fail to thoroughly investigate the influence of the components inside the ViT models on adversarial transferability, leading to inferior performance. To overcome the drawback, we launch a motivating study by linearly down-scaling the gradient of selected components inside the transformer-based models to find their influences on adversarial transferability. Based on the motivating study, we find that the skip connections influence transferability the most and believe that back-propagating deeper gradients to generate adversarial samples can boost their transferability. Therefore, we propose the Virtual Dense Connection method (VDC). Specifically, without changing the forward pass, we first recompose the original model to add virtual dense connections. We then densely back-propagate gradients of Attention maps and Multi-layer Perceptron (MLP) blocks via virtual dense connections to generate adversarial samples. Extensive experiments show that our proposed approach significantly outperforms the state-of-the-art baselines by an 8.2% improvement in the transferability between transformer-based models and a 7.2% improvement in the cross-structure transferability.

In summary, the contributions of this paper are:

- We launch a motivating study to analyze the influence of each component inside the transformer-based models on adversarial transferability. To this end, we linearly down-scale the gradient of each component to observe the transferability changes. We find that the gradient of the skip connection most influences the adversarial transferability.
- Based on the motivating study, we believe that back-propagating the gradient from deeper blocks to generate adversarial samples can improve their transferability. Therefore, we propose the Virtual Dense Connection method (VDC). VDC recomposes the original network to add virtual dense connections and then back-propagates gradients via virtual dense connections to generate transferable adversarial samples.
- Extensive experiments confirm that our method can outperform the state-of-the-art attacking approaches by a margin of 8.2% on the transferability between ViT models, and 7.2% on the cross-structure transferability from ViT models to CNN models.

Related Work

Transfer-based Adversarial Attacks

The transfer-based adversarial attack is one category of adversarial attacks under the black-box setting, which is built on the transferability of adversarial examples. Transferability is the phenomenon that the adversarial examples crafted by a local surrogate model can also mislead the target victim model. Therefore, black-box attackers can generate adversarial examples of a fully accessible surrogate model by white-box attacking algorithms and directly transfer the examples to the target victim model. Representative white-box attacks include Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2014) and Project Gradient Descent (PGD) (Madry et al. 2017). However, those white-box approaches reveal limited transferability, because the adversarial examples are model-specific and fail to mislead other models.

Therefore, researchers begin to boost the transferability of adversarial examples. The current state-of-the-art transfer-based attacks can be roughly classified into two trends: gradient-based approaches, and input transformation-based approaches. The gradient-based approaches utilize advanced optimizers (Dong et al. 2018), or model structures (Wu et al. 2020a; Xu et al. 2023; Deng et al. 2023) to modify the gradient to escape from the local optima and stabilize the update gradient. Momentum Iterative Method (MIM) (Dong et al. 2018) combines the momentum optimizer with the BIM to improve the adversarial transferability. Skip Gradient Method (SGM) (Wu et al. 2020a) utilizes the skip connection in the model structure to improve the transferability. SGM uses a decay factor to reduce the gradient from the residual module and focuses on the transferable low-level information to regularize the gradient. Input transformation-based approaches combine the gradients of the transformed images for generating transferable perturbation (Wu et al. 2021; Dong et al. 2019; Lin et al. 2019; Zhang et al. 2023a). Although those approaches have achieved state-of-the-art performance on boosting the transferability of convolution-based models, their performance drops dramatically on increasing the transferability of transformer-based models, because of the model structure difference between convolution-based models and transformer-based models.

Another category of black-box attacks is query-based attacks (Andriushchenko et al. 2020; Bai et al. 2020; Wu et al. 2023). However, query-based attacks require additional queries to the victim model, which lacks in efficiency in the real-world scenarios. Therefore, we focus on transfer-based attacks in this paper.

Transformer-based Models

The transformer is a neural network architecture utilizing the self-attention mechanism originating from the natural language processing field. Recently, the transformer design has been adapted into the computer vision field. Vision transformers (ViTs) (Dosovitskiy et al. 2020) divide the input image into a sequence of small image patches similar to a sequence of tokens for the language model. ViTs capture the relationship between image patches based on the multi-head self-attention mechanism. Besides the basic version of the ViT, advanced ViTs have been proposed to enhance the performance of ViTs on computer vision tasks. The pooling-based vision transformer (PiT) (Heo et al. 2021) decreases the spatial dimension and increases the channel dimension with pooling to improve the model capability. The data-

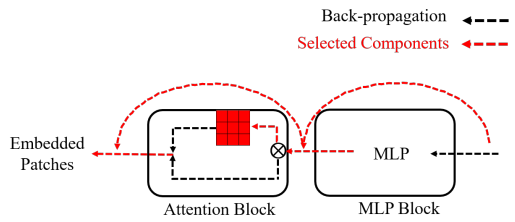

Figure 1: Illustration of the down-scaled gradients in the transformer block. All the dashed lines (red & black) depict the normally back-propagated gradients. The red dashed lines represent the selected gradient to analyze their influence on adversarial transferability.

efficient Vision Transformer (DeiT) (Touvron et al. 2021a) deploys a distillation token to learn knowledge from CNNs. The vision-friendly transformer (Visformer) (Chen et al. 2021) transits a transformer-based model to a convolution-based model.

With the development of transformer-based models, some researchers (Bhojanapalli et al. 2021; Shao et al. 2021) assess the robustness of ViTs based on white-box and black-box attacks. Other research (Mahmood, Mahmood, and Van Dijk 2021) also finds that the cross-structure transferability from transformer-based to convolution-based models is low. To understand the influence of the components in the network on adversarial transferability, we launch a motivating study to explore the influence of the gradient from each network component.

Attacks on Transformer-based Models

Researchers aim to improve transferability by exploring the unique structure of transformer-based models. Naseer et al. proposed Self-Ensemble (SE) to utilize the class token on each layer of ViTs with a shared classification head for the gradient ensemble and Token Refinement module (TR) to refine the class token with fine-tuning (Naseer et al. 2021). The Pay No Attention (PNA) (Wei et al. 2022) method explores the attention mechanism and skips the gradient of the attention during back-propagation to improve the transferability of adversarial examples. Although previous approaches utilize the special architecture of transformer-based models for transferable adversarial attacks, they fail to thoroughly explore the influence of each component on adversarial transferability, leading to limited improvement of transferability. Unlike previous methods, we analyze the influence of the gradient from each component in the transformer-based models on adversarial transferability with a motivating study, and then design an effective attacking method accordingly.

Motivating Study

In this motivating study, we analyze the influence of the gradient from each component in the transformer-based models on adversarial transferability. We select a representative transformer-based model, ViT-B/16 (Dosovitskiy et al. 2020), as the source model to craft adversarial examples.

Block	Component	ViT	CNN	Adv-CNN
Attention	QKV	45.4	24.1	16.6
	Attention Map	64.9	35.8	24.5
	skip Connection	19.0	13.3	8.8
MLP	MLP Layer	44.6	24.5	17.3
	Skip Connection	17.5	11.7	7.1

Table 1: The average adversarial transferability (%) against ViTs, CNNs, and adversarially-trained CNNs by scaling the gradients of different components in ViT-B/16.

We then measure the average transferability of the generated adversarial examples to multiple transformer-based and convolution-based models. The details of the target models are in Section . In order to reflect the influence of each component's gradient on transferability, we follow the idea of attribution (Sundararajan, Taly, and Yan 2017). Therefore, we gradually down-scale the gradient and compute the average adversarial transferability during the down-scaling process.

Specifically, we down-scale the gradient from each component using a linear sampling strategy, where we down-scale the gradient from 1 to 0 with a step size of 0.25. The transformer-based models consist of several transformer blocks. Each transformer block contains an Attention block and a MLP block. The Attention block first computes the QKV values and the attention map by the product of the query and key. Then the Attention block outputs the multiplication of the attention map and the QKV value. The MLP block passes the input through fully connected layers. Both the input and the output of the Attention block and MLP block are connected with a skip connection. Therefore, the components we select are QKV, attention map, the skip connection from the Attention block, MLP layers, and the skip connection from the MLP block, as shown in Figure 1.

We gradually down-scale the gradient of a selected component, fixing the other back-propagated gradients and computing the average transferability during the down-scaling process. As we can see from Table 1, the adversarial transferability drops dramatically with the reduction of the gradient from skip connections in the Attention block or the MLP block. Thus, we believe that the skip connection inside the transformer-based models influences the adversarial transferability the most. This phenomenon implies that the gradient from the deeper block through the skip connection enhances the adversarial transferability, which motivates us to back-propagate more gradients from deeper blocks to improve the adversarial transferability.

Method

Preliminary

We first set up some notations adopted in this paper. We regard a DNN image classifier as $f(\cdot)$. Given a sequence of image patches $\boldsymbol{x}_p = \{\boldsymbol{x}_p^1, \boldsymbol{x}_p^1, \cdots, \boldsymbol{x}_p^N\}$ divided from the original image $\boldsymbol{x}$ with a shape of $H \times W \times C$, $f(\boldsymbol{x})$ is the output of the image classifier. H, W, and C are the original image's height, width, and channel number, respectively. $\boldsymbol{x}_p^i$ denotes the i-th patch of the original image. The patch shape is $P \times P \times C$, where P is the predefined patch size.

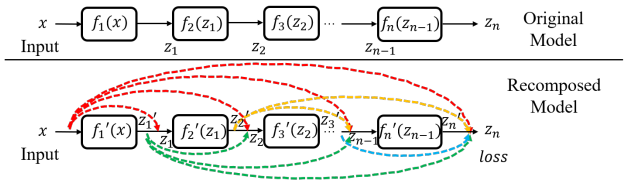

Figure 2: Illustration of model reparametrization by adding virtual dense connections. The inputs of the current blocks are propagated to all later blocks via virtual dense connections. We also modify the weights of the original block to guarantee that the input to the next block remains the same, keeping the forward pass of the original model.

Moreover, the total patch number N of an image is $\frac{H \cdot W}{P^2}$. We set $\boldsymbol{x}^{adv}$ as the adversarial example of image $\boldsymbol{x}$ with true label y. Adversarial examples satisfy two conditions: $f(\boldsymbol{x}^{adv}) \neq f(\boldsymbol{x})$ and $\left\|\boldsymbol{x} - \boldsymbol{x}^{adv}\right\|_p < \epsilon$.

The first condition implies that adversarial examples can mislead the image classifier with a wrong prediction. The second condition guarantees the difference between the adversarial example and the original image is smaller than a budget ϵ, so it is hard for a human to detect the distortion. $\|\cdot\|_p$ represents the L_p norm, and we measure the distortion by L_∞ norm in this paper, which is widely adopted in the literature (Dong et al. 2018).

Model Recomposition

In order to utilize the gradient from deeper blocks, one intuitive idea is to directly back-propagate the gradient through skip connections. However, there are no long-range connections in transformer-based models. Thus, we propose to recompose the original model to add additional virtual connections.

As shown in the upper part of Figure 2, we suppose there are n blocks in the network, and the output of $block_i$ is $\boldsymbol{z}_i = f_i(\boldsymbol{z}_{i-1})$ with input $\boldsymbol{z}_{i-1}$. Therefore, the output of the original network is $\boldsymbol{z}_n = f_n(\boldsymbol{z}_{n-1}) = \cdots = f_n(f_{n-1}(\cdots f_2(f_1(\boldsymbol{x}))\cdots))$.

Then, without changing the forward pass of the network, we aim to recompose the original model to add virtual dense connections. As shown in the lower part of Figure 2, we add virtual dense connections so that the output of each block is densely connected to the input of all the later blocks. Therefore, the additional input to $block_{i+1}$ through virtual dense connections is $\boldsymbol{x}+\boldsymbol{z}'_1+\cdots+\boldsymbol{z}'_{i-1}$, because we densely propagate all the outputs of previous blocks ($block_1$ - $block_{i-1}$) and input $\boldsymbol{x}$ to the input of $block_{i+1}$. Since we should keep the original forward pass of the model, we need to guarantee the input to each block of the recomposed model remains the same. Therefore, the function of $block_i$ is changed from $f_i(\boldsymbol{z}_{i-1})$ to $f'_i(\boldsymbol{z}_{i-1}) = f_i(\boldsymbol{z}_{i-1}) - \boldsymbol{x} - \sum_{k=1}^{i-1} \boldsymbol{z}'_k$.

As a result, we recompose the original model to add virtual dense connections without changing the functionality of the original model. The transformation facilitates back-propagating more gradients from deep blocks to shallow blocks.

Virtual Dense Connection Method

Based on the observation in the motivating study, we think that back-propagating more gradients from deeper blocks in the network can enhance adversarial transferability. Therefore, our Virtual Dense Connection method (VDC) back-propagates more gradients through virtual dense connections after model recomposition.

First, we denote the gradient of $block_i$ as $\boldsymbol{g}_i = \frac{\partial f_i(\boldsymbol{z}_{i-1})}{\partial \boldsymbol{z}_{i-1}}$. Then the gradient of the recomposed $block_i$ is:

$$\boldsymbol{g}'_i = \frac{\partial f'_i(\boldsymbol{z}_{i-1})}{\partial \boldsymbol{z}_{i-1}} = \frac{\partial (f_i(\boldsymbol{z}_{i-1}) - \boldsymbol{x} - \sum_{k=1}^{i-1} \boldsymbol{z}'_k)}{\partial \boldsymbol{z}_{i-1}} \tag{1}$$

$\boldsymbol{z}'_{i-1}$ is the output of the recomposed $block_{i-1}$, and $\boldsymbol{z}_{i-1}$ is the input to $block_i$. In the recomposed model with virtual dense connections, we have:

$$\boldsymbol{z}_{i-1} = \boldsymbol{x} + \sum_{k=1}^{i-1} \boldsymbol{z}'_k. \tag{2}$$

Therefore, the gradient $\boldsymbol{g}'_i = \frac{\partial (f_i(\boldsymbol{z}_{i-1}) - \boldsymbol{z}_{i-1})}{\partial \boldsymbol{z}_{i-1}} = \boldsymbol{g}_i - \mathbf{1}$, where $\mathbf{1}$ is the identity matrix.

To craft adversarial perturbation, we compute the gradient of the loss to the input $\boldsymbol{x}$ of the recomposed model:

$$\begin{aligned}
\frac{\partial loss}{\partial \boldsymbol{x}} &= \frac{\partial loss}{\partial \boldsymbol{z}_n}\frac{\partial \boldsymbol{z}_n}{\partial \boldsymbol{x}} = \frac{\partial loss}{\partial \boldsymbol{z}_n}\frac{\partial (\boldsymbol{z}'_n + \boldsymbol{x} + \sum_{k=1}^{n-1} \boldsymbol{z}'_k)}{\partial \boldsymbol{x}} \\
&= \frac{\partial loss}{\partial \boldsymbol{z}_n}\frac{\partial (f'_n(\boldsymbol{z}_{n-1}) + \boldsymbol{x} + \sum_{k=1}^{n-1} \boldsymbol{z}'_k)}{\partial \boldsymbol{x}} \\
&= \frac{\partial loss}{\partial \boldsymbol{z}_n}\frac{\partial (f'_n(\boldsymbol{z}_{n-1}) + \boldsymbol{z}_{n-1})}{\partial \boldsymbol{x}} \\
&= \frac{\partial loss}{\partial \boldsymbol{z}_n}\frac{\partial (f'_n(\boldsymbol{z}_{n-1}) + \boldsymbol{z}_{n-1})}{\partial \boldsymbol{z}_{\boldsymbol{n-1}}}\frac{\partial \boldsymbol{z}_{n-1}}{\partial \boldsymbol{x}} \\
&= \frac{\partial loss}{\partial \boldsymbol{z}_n}(\boldsymbol{g}'_i + \mathbf{1})\frac{\partial \boldsymbol{z}_{n-1}}{\partial \boldsymbol{x}} = \cdots = \frac{\partial loss}{\partial \boldsymbol{z}_n}\prod_{k=1}^{n}(\boldsymbol{g}'_k + \mathbf{1}).
\end{aligned} \tag{3}$$

To back-propagate more gradients from deeper blocks, VDC reduces the gradient inside recomposed blocks to back-propagate more gradients from deeper blocks via virtual dense connections. We utilize a factor $0 < \lambda < 1$ to reduce the gradient of each recomposed block. Therefore, the updated gradient on the input is:

$$\begin{aligned}
Grad &= \frac{\partial loss}{\partial \boldsymbol{z}_n}\prod_{k=1}^{n}(\lambda \boldsymbol{g}'_k + \mathbf{1}) = \frac{\partial loss}{\partial \boldsymbol{z}_n}\prod_{k=1}^{n}(\lambda(\boldsymbol{g}_k - \mathbf{1}) + \mathbf{1}) \\
&= \frac{\partial loss}{\partial \boldsymbol{z}_n}\prod_{k=1}^{n}(\lambda \boldsymbol{g}_k + (1-\lambda)\mathbf{1}).
\end{aligned} \tag{4}$$

We divide $Grad$ by λ^n for simplicity and denote $\gamma = \frac{1-\lambda}{\lambda}$. Then the gradient is simplified to:

$$Grad = \frac{\partial loss}{\partial \boldsymbol{z}_n}\prod_{k=1}^{n}(\boldsymbol{g}_k + \gamma \mathbf{1}). \tag{5}$$

Nevertheless, it is hard to compute $Grad$ because we cannot directly obtain the gradient $\boldsymbol{g}_k$ inside each block. The computation of $\boldsymbol{g}_k$ is expensive, which requires $O(H \times W \times C)$. Instead, we could acquire the gradient of the loss to the input of each block in $O(1)$, which we denote as $Grad_i = \frac{\partial loss}{\partial \boldsymbol{z}_{i-1}}$.

We expand the terms in $Grad$ and denote their patterns by the expansion of $\boldsymbol{g}_k$ or $\mathbf{1}$. For example, we denote the term $\frac{\partial loss}{\partial \boldsymbol{z}_n}(\boldsymbol{g}_k)(\gamma\mathbf{1})\cdots(\gamma\mathbf{1})$ by the pattern $[\boldsymbol{g}_k, \mathbf{1}, \cdots, \mathbf{1}]$. For the purpose of computing $Grad$ in $O(1)$, we only consider the terms in $Grad$ with one consecutively skip, which means that there is only one consecutive substring of $\mathbf{1}$ in the pattern, and the previous example is one consecutively skip term. Under one consecutive skip assumption, $Grad$ can be approximated by fusing the $Grad_i$ with all $Grad_j$, when $i < j \leq n$. Therefore, the combined gradient $ConGrad_i$ of $block_i$ can be expressed as follows:

$$ConGrad_i = Grad_i + s \cdot \sum_{j=i+1}^{n} Grad_j \cdot \gamma^{n-j+1}, \quad (6)$$

where we set a scaling factor $0 < s < 1$ to control the ratio of the back-propagated gradients from virtually connected deeper blocks. As a result, under the approximation assumption, the gradient from deeper blocks can be easily back-propagated in the backward pass and the computation of $Grad$ in $O(1)$. Finally, VDC updates the target image with the sign of $Grad$ by a small step size $\epsilon' = \frac{\epsilon}{T}$ in each iteration, where T is the iteration number. The update rule is formulated as:

$$x_{t+1}^{adv} = x_t^{adv} + \epsilon' \cdot sgn\{Grad\}. \quad (7)$$

Implementation

We aim to implement our proposed VDC on transformer-based models, taking the special design of ViTs into consideration. We demonstrate the components we select for utilizing VDC. The illustration of implementing VDC on transformer-based models is shown in Figure 3.

Attention Map. The Attention map is the core functionality of the transformer-based models, which computes the relationship between image patches. Although the receptive field of the transformer-based model is the whole image, the deep blocks capture more high-level semantics compared with shallow blocks (Dosovitskiy et al. 2020). The gradients of the attention map from deep blocks are meaningful because the gradients from deep blocks avoid overfitting to the model. Therefore, we deploy VDC on Attention block to densely connect the attention map in the Attention block.

MLP Block. The MLP block is another indispensable component in transformer-based models. Unlike the Attention block, the MLP block aggregates the channel-wise information of each patch. The skip connection of the MLP block also shows the most influence on adversarial transferability in the motivating study. Therefore, we also apply VDC to the MLP block.

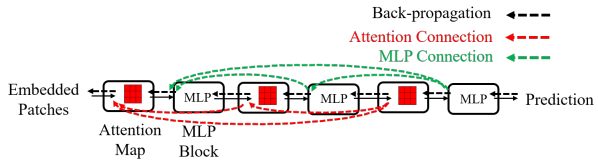

Figure 3: Illustration of Virtual Dense Connection method. The dark dash lines are the backward gradient through recomposed virtual connections on Attention maps and MLP blocks, which are in red and green dash lines to back-propagate more gradient from the deeper blocks.

Comparison with Previous Approaches

We recompose the original model without changing the forward functionality of the original transformer-based models and only modify the backward path through virtual dense connections to the Attention maps and MLP blocks. Previous ViT attacking methods explore the structure of a transformer-based model for boosting adversarial transferability (Wei et al. 2022). Nevertheless, they fail to investigate the advantages of each component in transformer-based models thoroughly. We do a motivating study to explore the benefit of the skip connection and the gradient from deeper blocks.

SGM (Wu et al. 2020a) assigns a decay factor on the gradients of residual modules to use more gradients from existing skip connections. In contrast, VDC utilizes model recomposition to construct virtual dense connections without changing the forward pass. Therefore, SGM can only be applied to models with skip connections, while VDC does not rely on such specific model structures. Moreover, VDC can back-propagate more gradients from deeper blocks through virtual dense connections. For efficiency, we implement VDC under the one consecutive skip approximation to compute the update gradient in $O(1)$.

Experiments

In this section, we first explain our experimental setup. Then we compare our approach with state-of-the-art adversarial attacks against transformer-based models and convolution-based models to demonstrate the effectiveness of our approach on improving the transferability between transformer-based models and the cross-structure transferability. Finally, we do an ablation study on the two components in our VDC as well as the hyper-parameters to understand our proposed approach better.

Experimental Setup

Our experiments mainly focus on the ImageNet dataset (Russakovsky et al. 2015) to attack image classification models, including transformer-based and convolution-based models. For fair comparisons, we follow the protocol (Wei et al. 2022) in the literature for the model and dataset.

Dataset. To align with the previous work, we follow the baseline method (Wei et al. 2022) to randomly sample 1000 images of different classes from the ILSVRC 2012 validation set (Russakovsky et al. 2015). We ensure that almost

Model	Attack	ViT-B/16	PiT-B	DeiT-B	Visformer-S	CaiT-S/24	TNT-S	LeViT-256	ConViT-B
ViT-B/16	MIM	**100.0**	34.5	64.3	36.5	64.1	50.2	33.8	66.0
	SE	99.9	31.8	68.3	40.5	67.4	59.3	43.8	63.7
	SGM	**100.0**	34.3	72.8	38.3	72.2	59.4	39.8	75.0
	PNA	**100.0**	45.2	78.6	47.7	78.6	62.8	47.1	79.5
	VDC	**100.0**	**54.8**	**85.8**	**57.4**	**84.1**	**74.8**	**58.1**	**85.9**
PiT-B	MIM	24.7	**100.0**	33.9	44.5	34.7	43.0	38.3	37.8
	SE	31.7	99.8	40.9	52.1	42.2	52.6	47.3	44.9
	SGM	30.3	**100.0**	44.3	62.3	47.7	62.6	56.4	47.1
	PNA	47.9	**100.0**	62.4	74.6	62.6	70.6	67.3	61.7
	VDC	**57.7**	**100.0**	**74.4**	**83.1**	**72.8**	**83.4**	**79.4**	**75.1**
DeiT-B	MIM	86.3	68.4	**100.0**	71.9	97.7	89.8	68.3	98.3
	SE	91.6	93.7	99.9	82.7	98.4	94.6	80.7	97.8
	SGM	88.3	65.7	**100.0**	73.1	97.7	92.3	74.3	97.4
	PNA	91.0	74.2	**100.0**	82.5	98.1	94.4	80.1	98.4
	VDC	**91.8**	**79.9**	**100.0**	**84.9**	**98.6**	**95.5**	**85.5**	**98.8**
Visformer-S	MIM	28.1	50.3	36.9	99.9	41.0	51.9	49.4	39.6
	SE	35.2	57.0	46.2	99.6	49.4	59.1	56.4	45.4
	SGM	15.5	39.6	25.9	**100.0**	29.5	45.4	41.3	26.3
	PNA	35.4	61.5	51.0	**100.0**	54.7	66.3	64.6	50.7
	VDC	**43.2**	**72.7**	**63.9**	**100.0**	**65.6**	**76.9**	**77.1**	**58.3**

Table 2: The attack success rates (%) against eight models by various transfer-based attacks. The best results are in bold.

all of the selected images can be correctly classified by the target models.

Models. We evaluate the transferability of adversarial examples of ViTs from two perspectives. We first assess the transferability between transformer-based models. We select four different transformer-based models as the local surrogate models to attack eight target transformer-based models, including the four surrogate models. The four surrogate models are ViT-B/16 (Dosovitskiy et al. 2020), PiT-B (Heo et al. 2021), DeiT-B (Touvron et al. 2021a) , and Visformer-S (Chen et al. 2021). The additional four target models are CaiT-S/24 (Touvron et al. 2021b), TNT-S (Han et al. 2021), LeViT-256 (Graham et al. 2021), and ConViT-B (d'Ascoli et al. 2021). We then evaluate the cross-structure transferability between transformer-based and convolution-based models. We choose two kinds of convolution-based models as the target models: normally trained undefended models and adversarially trained defended models. We select four undefended convolution-based models, including Inception-v3 (Inc-v3) (Szegedy et al. 2016), Inception-v4 (Inc-v4) (Szegedy et al. 2017), Inception-Resnet-v2 (IncRes-v2) (Szegedy et al. 2017), and Resnet-v2-152 (Res-v2) (He et al. 2016a,b). We test three adversarially trained models (Tramèr et al. 2017), including Inc-v3$_{ens3}$, Inc-v3$_{ens4}$, and IncRes-v2$_{adv}$. Besides, we evaluate the cross-transferability from convolution-based models to transformer-based models. We select Resnet-v2, Densenet121 (Dense-121) (Huang et al. 2017), and Mobilenetv3-small-075 (Mobile-v3) (Howard et al. 2019) as the surrogate models and test the attack success rate on the eight transformer-based models.

Baseline Methods. We choose MIM as our baseline approach, because all the baseline methods utilize the momentum optimizer to enhance the transferability (Dong et al. 2018). In order to show the advantages of our proposed VDC, we select SGM (Wu et al. 2020a) as our competitive baseline, which utilizes the skip connection structure inside the network with a decay factor to reduce the gradient from the residual module. To show the state-of-the-art performance of our method, we compare our method with two state-of-the-art attacking algorithms against transformer-based models: PNA (Wei et al. 2022) and SE (Naseer et al. 2021). PNA leverages the attention structure in the transformer block to craft transferable adversarial examples, and SE deploys the self-ensemble mechanism to augment the gradient. We do not compare VDC with TR (Naseer et al. 2021), because TR requires more training data and computation resources for fine-tuning on the Imagenet, which is unfair for performance comparison.

Evaluation Metric. We measure the adversarial transferability based on the attack success rate. We compute the ratio of the adversarial examples that successfully mislead the target model among all the generated adversarial examples.

Hyper-parameters. We follow the hyper-parameter setting of the baseline approaches in their implementations for a fair comparison. Following the previous setting in the literature (Wei et al. 2022), we set the budget $\epsilon = 16$, with the image pixel value ranging from 0 to 255. We pick the number of the iteration $T = 10$, so the step length $\alpha = \frac{\epsilon}{T} = 1.6$. Since all the baselines utilize the momentum optimizer, we set the decay factor $\mu = 1.0$. We resize all images to 224×224 as the input and pick the patch size to be 16 for the inputs of transformer-based models. For our proposed VDC, we set the scaling factor and the decay factor to be 0.1 and 0.5, respectively. Some transformer-based models have the same resolution in the whole network, while the others have different resolutions in different stages. Therefore, for the networks keeping the same resolution, we virtually connect all the blocks during the back-propagation. Otherwise, we only virtually connect the blocks with the same resolution.

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-v2	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{adv}$
ViT-B/16	MIM	31.7	28.6	26.1	29.4	22.3	19.8	16.5
	SE	40.8	40.0	31.5	38.8	31.0	30.5	23.8
	SGM	29.5	25.9	21.6	26.0	17.6	17.0	13.9
	PNA	42.7	37.5	35.3	39.5	29.0	27.3	22.6
	VDC	**49.3**	**44.4**	**39.3**	**44.8**	**33.8**	**33.8**	**27.8**
PiT-B	MIM	36.3	34.8	27.4	29.6	19.0	18.3	14.1
	SE	46.4	41.2	35.0	39.4	25.3	22.3	19.5
	SGM	39.8	35.4	29.8	30.8	18.1	16.4	11.5
	PNA	59.3	56.3	49.8	53.0	33.3	32.0	25.5
	VDC	**68.2**	**60.3**	**57.0**	**59.5**	**42.2**	**39.8**	**32.3**
DeiT-B	MIM	56.1	50.9	47.9	52.9	40.8	38.7	32.6
	SE	63.2	57.6	59.7	63.1	48.5	44.3	38.6
	SGM	52.1	45.8	43.2	46.9	31.8	31.5	27.2
	PNA	66.5	60.7	60.9	64.0	49.3	46.1	40.8
	VDC	**69.9**	**63.0**	**63.8**	**65.8**	**53.3**	**52.0**	**45.3**
Visformer-S	MIM	44.5	42.5	36.6	39.6	24.4	20.5	16.6
	SE	55.5	55.0	44.9	48.3	30.9	26.6	24.4
	SGM	33.1	32.7	24.6	26.2	11.7	9.4	6.9
	PNA	55.9	54.6	46.0	51.7	29.3	26.2	21.1
	VDC	**71.9**	**69.8**	**60.9**	**65.0**	**40.8**	**34.8**	**28.3**

Table 3: The attack success rates (%) against seven models by various transfer-based attacks. The best results are in bold.

Experimental Results

We present the experimental results of the adversarial transferability of our approach compared with baselines on different attacking settings. We craft adversarial examples by our approach and other baselines on the surrogate models and transfer the adversarial examples to target models. We measure the transferability between transformer-based models and the cross-structure transferability from transformer-based models to convolution-based models.

We first assess the transferability between transformer-based models. We observe from Table 2 that, our proposed VDC achieves a 100% white-box attack success rate and outperforms all the baselines with a large margin of 8.2% on average under the black-box setting. Compared with SGM, which utilizes the skip connection structure in the network, our approach deploys virtual dense connections to the deeper blocks exerting significant improvement on the transferability. This result validates our assumption in the motivating study that back-propagating more gradients from deeper blocks can boost transferability and shows the advantages of adding virtual dense connections in the backward path. Compared with PNA and SE, which use different architectures of the transformer-based models to enhance transferability, our approach adds more connections virtually based on model recomposition. The remarkable performance also confirms the effectiveness of our selected architecture components.

Moreover, we evaluate the cross-structure transferability from transformer-based to convolution-based models. As shown in Table 3, the cross-structure transferability drops compared with the transferability between transformer-based models, due to the structure difference of models. Compared with baselines, our proposed VDC still enhances the cross-structure transferability by over 7.2% on average, validating the superiority of the proposed VDC.

Component	ViT	CNN	Adv-CNN
None	56.2	29.0	19.5
MLP	66.0	34.4	24.1
Attention	66.8	37.0	24.6
Attention + MLP (VDC)	**75.1**	**44.5**	**31.8**

Table 4: The average adversarial transferability (%) against ViTs, CNNs, and adversarially trained CNNs by using dense connection on different components in ViT-B/16.

Ablation Study

We do an ablation study to explore the contribution of the components in VDC by attacking the ViT-B/16 model. We generate adversarial examples by different combinations of the components in VDC and measure the transferability. The experimental result is shown in Table 4. We can see that both densely connecting the MLP blocks and the Attention maps can enhance adversarial transferability. The transferability improvement by densely connecting the MLP block is a little bit inferior than the Attention map, because the Attention map is the core functionality in transformer-based models.

Conclusion

In this paper, we start with a motivating study to conclude that back-propagating gradients from deeper blocks can enhance transferability. We propose the Virtual Dense Connection method (VDC) to back-propagate more gradients from deeper blocks. Specifically, we recompose the original model to add virtual dense connections without changing the forward pass. Then we back-propagate gradients of deeper Attention maps and MLP blocks via virtual dense connections when generating adversarial samples. Extensive experiments validate the superiority of our approach over the state-of-the-art approaches.

Acknowledgments

The work described in this paper was supported by the National Natural Science Foundation of China (Grant No. 62206318) and the Research Grants Council of the Hong Kong Special Administrative Region, China (CUHK 14206921 of the General Research Fund).

References

Andriushchenko, M.; Croce, F.; Flammarion, N.; and Hein, M. 2020. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, 484–501. Springer.

Bai, Y.; Zeng, Y.; Jiang, Y.; Wang, Y.; Xia, S.-T.; and Guo, W. 2020. Improving query efficiency of black-box adversarial attack. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, 101–116. Springer.

Bhojanapalli, S.; Chakrabarti, A.; Glasner, D.; Li, D.; Unterthiner, T.; and Veit, A. 2021. Understanding robustness of transformers for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10231–10241.

Chen, Z.; Xie, L.; Niu, J.; Liu, X.; Wei, L.; and Tian, Q. 2021. Visformer: The vision-friendly transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 589–598.

Deng, Y.; Wu, W.; Zhang, J.; and Zheng, Z. 2023. Blurred-Dilated Method for Adversarial Attacks. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9185–9193.

Dong, Y.; Pang, T.; Su, H.; and 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*, 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 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

d'Ascoli, S.; Touvron, H.; Leavitt, M. L.; Morcos, A. S.; Biroli, G.; and Sagun, L. 2021. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, 2286–2296. PMLR.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Graham, B.; El-Nouby, A.; Touvron, H.; Stock, P.; Joulin, A.; Jégou, H.; and Douze, M. 2021. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF international conference on computer vision*, 12259–12269.

Han, K.; Xiao, A.; Wu, E.; Guo, J.; Xu, C.; and Wang, Y. 2021. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34: 15908–15919.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Identity mappings in deep residual networks. In *European conference on computer vision*, 630–645. Springer.

Heo, B.; Yun, S.; Han, D.; Chun, S.; Choe, J.; and Oh, S. J. 2021. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 11936–11945.

Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. 2019. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1314–1324.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Lin, J.; Song, C.; He, K.; Wang, L.; and Hopcroft, J. E. 2019. Nesterov accelerated gradient and scale invariance for adversarial attacks. *arXiv preprint arXiv:1908.06281*.

Luo, W.; Li, Y.; Urtasun, R.; and Zemel, R. 2016. Understanding the effective receptive field in deep convolutional neural networks. *Advances in neural information processing systems*, 29.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mahmood, K.; Mahmood, R.; and Van Dijk, M. 2021. On the robustness of vision transformers to adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7838–7847.

Naseer, M.; Ranasinghe, K.; Khan, S.; Khan, F. S.; and Porikli, F. 2021. On improving adversarial transferability of vision transformers. *arXiv preprint arXiv:2106.04169*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252.

Shao, R.; Shi, Z.; Yi, J.; Chen, P.-Y.; and Hsieh, C.-J. 2021. On the adversarial robustness of vision transformers. *arXiv preprint arXiv:2103.15670*.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, 3319–3328. PMLR.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021a. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, 10347–10357. PMLR.

Touvron, H.; Cord, M.; Sablayrolles, A.; Synnaeve, G.; and Jégou, H. 2021b. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 32–42.

Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*.

Wei, Z.; Chen, J.; Goldblum, M.; Wu, Z.; Goldstein, T.; and Jiang, Y.-G. 2022. Towards transferable adversarial attacks on vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2668–2676.

Wu, D.; Wang, Y.; Xia, S.-T.; Bailey, J.; and Ma, X. 2020a. Skip connections matter: On the transferability of adversarial examples generated with resnets. *arXiv preprint arXiv:2002.05990*.

Wu, W.; Su, Y.; Chen, X.; Zhao, S.; King, I.; Lyu, M. R.; and Tai, Y.-W. 2020b. Boosting the transferability of adversarial samples via attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1161–1170.

Wu, W.; Su, Y.; Chen, X.; Zhao, S.; King, I.; Lyu, M. R.; and Tai, Y.-W. 2020c. Towards global explanations of convolutional neural networks with concept attribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8652–8661.

Wu, W.; Su, Y.; Lyu, M. R.; and King, I. 2021. Improving the transferability of adversarial samples with adversarial transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9024–9033.

Wu, W.; Xu, H.; Zhong, S.; Lyu, M. R.; and King, I. 2019. Deep Validation: Toward detecting real-world corner cases for deep neural networks. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 125–137. IEEE.

Wu, W.; Zhang, J.; Wei, V. J.; Chen, X.; Zheng, Z.; King, I.; and Lyu, M. R. 2023. Practical and Efficient Model Extraction of Sentiment Analysis APIs. In *IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, 524–536. IEEE.

Xu, Z.; Gu, Z.; Zhang, J.; Cui, S.; Meng, C.; and Wang, W. 2023. Backpropagation Path Search On Adversarial Transferability. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4663–4673.

Zhang, J.; Huang, J.-t.; Wang, W.; Li, Y.; Wu, W.; Wang, X.; Su, Y.; and Lyu, M. R. 2023a. Improving the Transferability of Adversarial Samples by Path-Augmented Method. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8173–8182.

Zhang, J.; Huang, Y.; Wu, W.; and Lyu, M. R. 2023b. Transferable Adversarial Attacks on Vision Transformers with Token Gradient Regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16415–16424.

Zhang, J.; Huang, Y.-C.; Wu, W.; and Lyu, M. R. 2023c. Towards semantics-and domain-aware adversarial attacks. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 536–544.

Zhang, J.; Wu, W.; Huang, J.-t.; Huang, Y.; Wang, W.; Su, Y.; and Lyu, M. R. 2022. Improving Adversarial Transferability via Neuron Attribution-Based Attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14993–15002.

Zhang, J.; Xu, Z.; Cui, S.; Meng, C.; Wu, W.; and Lyu, M. R. 2023d. On the Robustness of Latent Diffusion Models. *arXiv preprint arXiv:2306.08257*.

Zhang, Z.; Lu, X.; Cao, G.; Yang, Y.; Jiao, L.; and Liu, F. 2021. ViT-YOLO: Transformer-based YOLO for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2799–2808.

Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P. H.; et al. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6881–6890.