How Does a Deep Learning Model Architecture Impact Its Privacy?

Guangsheng Zhang[1] Bo Liu[1] Huan Tian[1] Tianqing Zhu[1]
Ming Ding[2] Wanlei Zhou[3]

[1]*University of Technology Sydney* [2]*Data 61, CSIRO, Australia* [3]*City University of Macau*

Abstract

As a booming research area in the past decade, deep learning technologies have been driven by big data collected and processed on an unprecedented scale. However, the sensitive information in the collected training data raises privacy concerns. Recent research indicated that deep learning models are vulnerable to various privacy attacks, including membership inference attacks, attribute inference attacks, and gradient inversion attacks. It is noteworthy that the performance of the attacks varies from model to model. In this paper, we conduct empirical analyses to answer a fundamental question: *Does model architecture affect model privacy?* We investigate several representative model architectures from CNNs to Transformers, and show that Transformers are generally more vulnerable to privacy attacks than CNNs. We further demonstrate that the micro design of activation layers, stem layers, and bias parameters, are the major reasons why CNNs are more resilient to privacy attacks than Transformers. We also find that the presence of attention modules is another reason why Transformers are more vulnerable to privacy attacks. We hope our discovery can shed some new light on how to defend against the investigated privacy attacks and help the community build privacy-friendly model architectures.

1 Introduction

Deep learning has been gaining massive attention over the past several years. Training deep learning models requires the collection and processing of user data, which raises privacy concerns. The collected training data might contain sensitive information, which could be retrieved or recovered by adversaries. Indeed, deep learning models trained by collected data have been shown to be vulnerable to various privacy attacks. For example, membership inference attacks aim to determine whether a specific data sample belongs to the training data [40, 42]. Attribute inference attacks seek implicit attributes learned by models other than the target attribute [35, 45]. Gradient inversion attacks aim to reconstruct information from the training data [11, 12]. These attacks have demonstrated that deep learning models can be exploited to incur severe privacy leakage.

Previous research discovered that the major reason for privacy leakage from deep learning models is overfitting [5, 16, 27, 42]. In other words, the models tend to learn specific information from the training data, which leads to privacy leakage. However, even with similar degrees of overfitting, the performance of the attacks varies from model to model. Some deep learning models might be more vulnerable than others, which is not well-understood by the research community. We believe that there are some deeper reasons why some deep learning models are more vulnerable to privacy attacks in addition to overfitting. This motivates us to investigate the following topic in this paper: *How does model architecture affect its privacy preservation performance?*

In this paper, we answer this question by analyzing the performance of different deep learning models under a variety of state-of-the-art privacy attacks. We start by comparing two mainstream model architectures: convolutional neural networks (CNNs) and Transformers. CNN-based models are dominant in computer vision, which heavily contribute to the prosperity of deep learning technology in the 2010s. The sliding-window strategy in CNNs enables the model to extract local information in image samples. Transformer-based models were proposed in the late 2010s. They first made an appearance in natural language processing (NLP) and were then introduced to the field of computer vision. The attention mechanism in Transformers makes them have larger receptive fields compared to CNNs, leading to better performance in terms of accuracy. The tremendous achievements and wide acceptance of these two architectures provide us a good opportunity to make a comparison analysis regarding model privacy. We found that, in general, *Transformers tend to be more vulnerable* to mainstream privacy attacks than CNNs.

While Transformers and CNNs have different designs from many perspectives, we further investigate whether there are some key designs in the model that have a major impact on privacy. To this end, we start from a popular CNN-based

model, ResNet-50 [15], and gradually modify the model to incorporate the key designs of Transformers. This leads us to the structure of ConvNeXt [30]. We evaluate the privacy leakage through this process and identify several key components that have a significant impact on privacy: (1) the design of the activation layers; (2) the design of stem layers; (3) the bias parameters in the layer design. We conduct further ablation studies to verify our observations.

What is more, we investigate which module contributes to privacy leakage more in a Transformer architecture. We evaluate the privacy leakage of several major modules by sending only selected gradients to the gradient inversion attacks, and discover that attention modules could also lead to privacy leakage.

In summary, our contributions in this paper are summarized as follows:

- For the first time, we investigate whether and how model architectures affect privacy.
- We evaluate the privacy leakages with three popular privacy attack methods, i.e., membership inference attacks, attribute inference attacks, and gradient inversion attacks, on two types of widely-adopted model architectures, i.e., CNNs and Transformers. We discover that Transformers tend to be more vulnerable to these privacy attacks than CNNs.
- We further identify three key factors: (1) the design of activation layers, (2) the design of stem layers, and (3) the design of bias parameters, as the main reasons that make CNNs more resilient to privacy attacks than Transformers.
- We also discover that the attention modules in Transformers could make them vulnerable to privacy attacks.

2 Related Work

In this section, we review the fundamental concepts and representative models of CNNs and Transformers. In addition, we introduce three major privacy attacks that we use to evaluate the model privacy: membership inference attack, attribute inference attacks and gradient inversion attack.

2.1 Convolutional Neural Networks (CNNs)

Convolutional neural networks (CNNs) are neural networks that use convolutional layers to extract features from input data. In contrast to fully connected networks, CNNs use convolutional kernels to connect small samples to neurons to make feature extraction. Thus, the parameters of the model are downsized, and local features can be recognized. Multiple techniques are applied to build a CNN model, including padding, pooling, dilated convolution, group convolution, and others.

The idea of convolutional neural networks (CNNs) dates back to the 1980s [25]. However, the invention of AlexNet [24] makes CNNs the most representative networks in computer vision. Later research improved the accuracy and efficiency of models [43, 48]. ResNet [15] solved the problem of training deep networks using skip connections. Other representative networks consist of Inceptions [49], MobileNet [19], ResNeXt [54], RegNet [38], ConvNeXt [30].

2.2 Vision Transformers

Derived from natural language processing, vision Transformers partition the input image into several patches to form a 1D sequence of token embeddings. Its outstanding performance is due to the multi-head self-attention modules [53]. This attention mechanism has advanced the field of natural language processing [3, 8, 55], and Transformers have then been introduced to the field of computer vision as ViT [9]. The work of [9, 47] has demonstrated that ViTs could outperform CNNs on multiple downstream tasks. Later research has focused on multiple improvements of ViTs, including tokenization [57], position encoding [29]. Other improvements consist of lightweight Transformers in DeiT [50] and "sliding windows" strategy in Swin Transformers [29].

There has been literature on comparisons between CNNs and Transformers, but on the robustness perspective [2, 37] and the explainability perspective [39]. Different from previous works, our work focuses on the privacy leakage of CNNs and Transformers.

2.3 Privacy Attacks on Deep Learning Models

One major concern of deep learning privacy is that the model may reveal sensitive information from the training dataset. An adversary can predict whether a particular sample is in the model's training dataset via membership inference attacks, or disclose the implicit attributes of data samples via attribute inference attacks, or even recover the private data samples that are used to train a neural network via gradient inversion attacks.

Membership inference attacks were first proposed in [42] using an attack model to differentiate member samples from non-member samples in the training data. In order to launch the attacks, shadow models are used to mimic the behavior of victim models [40, 42]. The prediction results of victim models are collected for attack model training. Usually the confidence scores were utilized [42], but more recent work (label-only attacks) applied prediction labels to successfully launch the attacks [6, 26]. The attacks can also be made by designing a metric with a threshold by querying the shadow model [46]. Other researchers extended the attacks into new

domains, including generative models [4, 14], semantic segmentation [17, 59], federated learning [36, 52], and transfer learning [44, 65].

Attribute inference attacks, as another major group of privacy attack methods, attempt to reveal a specific sensitive attribute of a data sample by analyzing the representation of the victim model trained by the victim dataset. Melis et al. [35] presented the first attribute inference attack against deep learning models, which can be adopted in federated learning. Song and Shmatikov [45] later claimed that the over-learning feature of deep learning models caused the launch of the attacks. A relaxed notion of attribute inference attacks was investigated in [62]. Attributes could also be inferred through model explanations [10].

Gradient inversion attacks focus on the reconstruction of training samples at the local clients in federated learning. Using the publicly shared gradients in the server, the adversary could launch the attacks by reconstructing the training samples using gradient matching. DLG attack [64] and its variant, iDLG attack [63], were the early ones to employ an optimization-based technique to reconstruct the training samples. Later research like Inverting Gradients [12] and GradInversion [56] improved the attack performance by adding regularizations to the the optimization. APRIL [33] and GradViT [13] developed the attack methods to extract sensitive information against Transformers.

There have been several evaluations and reviews between these privacy attacks against deep learning models [16, 20, 27, 28, 46, 60, 61]. However, our goal is to evaluate the model architectures leveraging these privacy attacks. We utilize conventional privacy attacks [12, 35, 40, 42, 45] as the baseline attacks in our analysis, for these attack methods have inspired many follow-up research works, and they are suitable for evaluation on various models and datasets.

3 Evaluating the Impact of the Model Architecture on Privacy

In this section, we present our approach to assessing the impact of model architectures on privacy leakage. In order to organize our study in a thorough and logical manner, We will answer the following research questions sequentially:

- Q1: What attack methods should we choose to analyze the privacy leakage in model architectures?
  - A1: We choose three representative privacy attack methods: membership inference attacks, attribute inference attacks, and gradient inversion attacks. Section 3.1 introduces the privacy threat models of these attacks.
- Q2: What model architectures should we choose for the evaluation of these attacks?
  - A2: We choose two mainstream model architectures: CNNs and Transformers. Section 3.2 provides a detailed list of the models of CNNs and Transformers as our selection, and these models are selected as victim models in privacy attacks.
- Q3: How should we investigate what designs in model architectures contribute to privacy leakage?
  - A3: We first evaluate the attacks against multiple CNN and Transformer-based models, followed by further analysis of the micro designs. In more details, Section 3.3 gives the steps of evaluation on the attacks against model architectures changing from ResNet to ConvNeXt. Then, Section 3.4 analyzes the privacy leakage of the selected modules in ViTs.

We believe the methodology of our evaluation could shed light on the effect of model privacy from the perspective of model architectures.

3.1 Privacy Threat Models

3.1.1 Membership Inference Attacks

Initiating a membership inference attack [40, 42] requires three models: the victim model $\mathcal{V}$ (the target of the attack), the shadow model $\mathcal{S}$ (the model to mimic the behavior of the victim model), and the attack model $\mathcal{A}$ (the classifier to give results whether the sample belongs to the member or non-member data). The following paragraphs provide explanations of how the attacks work, including the attack preparation, the attack model training, and the attack model inference.

Attack Preparation. As the adversary only has black-box access to the victim model $\mathcal{V}$, he can only make queries to the model and record prediction results. The adversary collects a shadow dataset $\mathcal{D}_S$, which is usually a dataset from the same data distribution as the victim dataset $\mathcal{D}_V$. The shadow dataset $\mathcal{D}_S$ can be split into two subsets: $\mathcal{D}_S^{train}$ for training and $\mathcal{D}_S^{test}$ for testing.

Attack Model Training. The shadow model $\mathcal{S}$ and the shadow dataset $\mathcal{D}_S$ are used for the attack model training. The prediction result of a data sample from the shadow dataset $\mathcal{D}_S$ is a vector containing the confidence scores of each class. This vector is concatenated with a label showing whether the prediction is correct or not, and the whole vector is denoted as $\mathcal{P}_S^i$, and $\mathcal{P}_S = \{\mathcal{P}_S^i, i = 1, ..., n\}$. $\mathcal{P}_S$ is considered as the input of the attack model $\mathcal{A}$. As the attack model is a binary classifier, we use a simple three-layer MLP (multi-layer perceptron) model.

Attack Model Inference. The adversary makes queries of the victim model $\mathcal{V}$ with the victim dataset $\mathcal{D}_V$, and receives prediction results as the input of the attack model. Then the attack model $\mathcal{A}$ makes predictions on whether the data sample is a member or non-member data sample.

There are other kinds of membership inference attacks, including metric-based attacks and label-only attacks. Instead of using a neural network to be the attack model, metric-based attacks [46] launch the attacks using a certain metric and threshold to separate member data from non-member data. Label-only attacks [6, 26] relax the assumptions of the threat model leveraging only prediction labels as the input of the attack model. In this paper, we consider only conventional model-based attacks, because we focus on the attack performance against various victim models.

3.1.2 Attribute Inference Attacks

The adversary of attribute inference attacks [35, 45] attempts to extract sensitive attributes from the victim model. For example, a victim model's task is to learn to classify whether the person has a beard or not. The adversary can infer the race of the person based on the representation of the victim model.

Attack Preparation. The victim model $\mathcal{V}$ is trained by the victim dataset $\mathcal{D}_V$ with two subsets $\mathcal{D}_V^{train}$ and $\mathcal{D}_V^{test}$ for the training and testing.

Attack Model Training. The attack model $\mathcal{A}$ is trained using an auxiliary dataset $\mathcal{D}_A^{train}$, which has the representation h and the attribute a, and we have $(h,a) \in \mathcal{D}_A$.

Attack Model Inference. The adversary takes a data sample's representation h as the input, and uses the attack model $\mathcal{A}$ to infer the attribute result.

3.1.3 Gradient Inversion Attacks

The launching of the gradient inversion attack [12, 63, 64] is basically about solving an optimization problem, which is minimizing the difference between the calculated model gradients and the original model gradients. Thus, the input data sample can be reconstructed after a certain amount of iterations.

Attack Preparation. The adversary hides in the central server in federated learning, where model gradients are aggregated to form a centralized model. In the current research, an extreme case is considered when there is only one sample in a batch. This makes it simple to evaluate the reconstruction's effectiveness. In this study, we evaluate a variety of models in relation to this context.

Table 1: Parameter sizes for CNNs and Transformers.

Models	**MobileNetV3-L**	**ResNet-18**	**ResNet-50**	**ResNet-50x3**
Params	4.21M	11.18M	23.52M	132.82M
Models	**ResNet-101**	**ResNet-152**	**ConvNeXt-T**	**ConvNeXt-S**
Params	42.51M	58.16M	27.83M	49.46M
Models	**DeiT-T**	**DeiT-S**	**DeiT-B**	**Swin-T**
Params	5.49M	21.60M	85.66M	27.51M
Models	**Swin-S**	**Swin-B**	**ViT-B**	**ViT-L**
Params	48.80M	86.70M	85.66M	303.12M

Gradient Reconstruction. The aggregated model gradients are denoted as $\nabla_\theta \mathcal{L}_\theta(x,y)$, where θ is the model parameters, x and y are the original input image and its ground truth in a local client, and $\mathcal{L}$ is the cost function for the model. The adversary generates a dummy image x^* to begin the reconstruction. The adversary tries to minimize this cost function: $\arg\min_x ||\nabla_\theta \mathcal{L}_\theta(x,y) - \nabla_\theta \mathcal{L}_\theta(x^*,y)||^2$. The dummy image x^* will be reconstructed to be close to x as much as possible.

3.2 CNNs vs Transformers

We start with investigating the privacy of two mainstream architectures: CNNs and Transformers. We carefully select several popular CNNs and Transformers for the attacks to analyze the privacy leakage, as listed below.

- ResNets [15] are baseline models because of their popularity. The usage of residual blocks makes them backbone networks in model design. Some selected ResNets are ResNet-18, ResNet-50, ResNet-101, ResNet-152 (from small to large in parameter sizes). ResNet-50x3 is a wide ResNet that is larger in network channels than standard ResNet.
- MobileNetV3-L [19] is a lightweight CNN model using depth-wise separable convolutions.
- ConvNeXts [30] are new state-of-the-art CNN-based models that incorporate Transformer-style layer designs. The selected ConvNeXts for comparision are ConvNeXt-T, ConvNeXt-S (from small to large in parameter sizes).
- ViTs [9] are Vision Transformers that interpret images as a sequence of patches and process it by Transformer blocks. The selected ViTs are ViT-B, ViT-L (from small to large in parameter sizes).
- Swin Transformers [29] bring shifted window mechanisms in model design to improve task performance. The selected Swin Transformers are Swin-T, Swin-S, Swin-B (from small to large in parameter sizes).
- DeiTs [50] are lightweight Vision Transformers applying distillation technologies to reduce model sizes. The selected DeiTs are DeiT-T, DeiT-S, DeiT-B (from small to large in parameter sizes).

Table 1 shows these selected models and their parameter sizes.

First, we evaluate the model privacy on membership inference attacks. To make a fair comparison, we group all models by parameter sizes. For CNN-based models, we choose ResNets [15] as baseline models because of their popularity. We also choose a lightweight model, MobileNetV3 [19], for comparison with small parameter sizes. For Transformers, we choose Swin Transformers [29] for their excellent performance on multiple downstream tasks. We also choose a lightweight model, DeiT [50], for comparison with small parameter sizes.

Second, we evaluate the model privacy on attribute inference attacks. Just like the evaluation on membership inference attacks, we also group models of CNNs and Transformers by parameter sizes. ResNets [15] are compared with Swin Transformers [29]. Then lightweight models like MobileNetV3 [19] and DeiT [50] are compared as a group.

Third, we evaluate the model privacy on gradient inversion attacks. This time we extend our analysis to more model architectures. For CNNs, we test MobileNetV3 [19], more ResNet variants [15,58] and the recently Transformer-inspired model ConvNeXt [30]. For Transformers, we test multiple variants of DeiTs [50], Swin Transformers [29], and ViTs [9].

Apart from model architectures, training Transformers requires a modernized training procedure compared to training traditional CNNs. In order to make fair comparisons, we apply the same training recipe for CNNs and Transformers that is close to Swin Transformers [29], including 300 epochs, AdamW optimizers [32], and some other procedures.

Main findings. Our research reveals that Transformer-based models are typically more vulnerable to privacy attacks. However, some modernized CNNs like ConvNeXt (which borrows some design ideas from Transformers) also suffer from more severe privacy leakage when gradient inversion attacks are undertaken. This phenomenon suggests that some special designs may have a greater impact on privacy. The detailed evaluations are presented in Section 4.2, Section 4.3, and Section 4.4. Therefore, we further dig into the designs of model architectures to see what design poses the most major impact on privacy attacks.

3.3 Impact of Micro Designs on Privacy

We follow the steps of designing ConvNeXt [30] but make several adjustments based on our analysis. ConvNeXt is a CNN-based model that incorporates multiple schemes of a Transformer model, referring to Swin Transformer [29]. Since our prior study in the previous subsection shows that gradient inversion attacks on ConvNeXt achieve higher accuracy than the attacks on ResNet, we suppose that some designs in ConvNeXt might have an impact on privacy leakage. As ConvNeXt is built step by step from a starting point of ResNet, our study analyzes each model architecture in the process to investigate which step makes major contributions in terms of causing privacy leakage. We leverage ResNet-50 and ConvNeXt-T in our analysis, and the following procedures are made based on these two model architectures. Based on ResNet-50 and ConvNeXt-T, we outline 15 model architectures and test them on a randomly selected sample in CIFAR10, and illustrate the MSE results between ground truth images and reconstructed images in each step to compare the privacy leakage. The actions in the 15 steps and key results are as follows:

- **ResNet-50.** We start from this model, and the MSE result is 1.9235.
- **Changing channel dimensions.** ResNet-50 uses different channel dimensions in each stage (i.e. $(64, 128, 256, 512)$). We change them to align with ConvNeXt-T (i.e. $(96, 192, 384, 768)$). Here the MSE result increases to 2.4774.
- **Changing the stage compute ratio.** ResNet applies a multi-stage design by changing channel dimensions. ResNet-50 has a stage computer ratio of $(3, 4, 6, 3)$, while ConvNeXts and Swin Transformers change this to $(3, 3, 9, 3)$. We follow this change in this step, and the MSE result decreases to 2.2503.
- **Applying "Patchify".** Vision transformers slide input images into patches for later processing. We replace the stem convolutional layers with kernel (4×4) and stride 4. The MSE result drops to 1.2210.
- **Applying "ResNeXtify".** Grouped convolution used in ResNeXt [54] significantly reduces parameter sizes with comparable performance. We apply depth-wise convolution, which is a convolution layer with the same number of groups and channels. Now the MSE result has changed to 1.5778.
- **Using the inverted bottleneck.** The inverted bottleneck design is utilized in various models, including MobileNet [19], ConvNeXt [30], and Swin Transformers [29]. Here we apply this step, and the MSE result decreases to 0.9602.
- **Enlarging kernel sizes.** We use a larger kernel size of (7×7) instead of (3×3), to bring the parameters in line with ConvNeXt. The MSE result in this step is 0.8317.
- **Forming the new stem layers.** Here we remove the activation layer and the maxpool layer, which are originally in ResNet, reaching an MSE result of 0.1552.
- **Changing ReLU to GELU.** The Gaussian Error Linear Unit (GELU) [18] is a variant of ReLU widely adopted in Transformers. We make this change to the model and receive an MSE result of 0.6828.

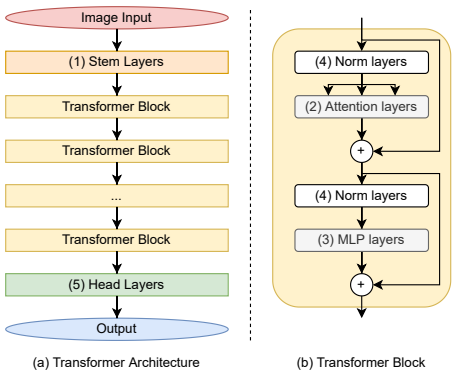


Figure 1: An illustration of a Vision Transformer architecture. Numbers (1) - (5) denote the modules we evaluate in gradient inversion attacks. (a): The whole model architecture. (b): Detailed architecture of each Transformer block.

- **Removing some activation layers.** Transformer blocks usually have fewer activation layers. We leave only one activation layer in the block, and the MSE result is now 0.0316.

- **Removing some normalization layers.** BatchNorm (BN) layers are also reduced to one in the block, and the MSE result changes to 0.0323.

- **Changing BN to LN.** The frequent usage of LayerNorm (LN) layers [1] in Transformers inspires us to replace BN layers in the model. Now our reconstruction MSE is 0.3140.

- **Adding bias.** The convolutional layers in ConvNeXt have bias parameters set as True, while the ones in ResNet do not. In this step, we let all convolutional layers have bias parameters, reaching an MSE result of 0.0075.

- **Separating downsampling layers.** The downsampling layers are moved between stages, and an LN layer is needed to maintain stability during training. We receive an MSE result of 0.0198 after this.

- **Final touches to reach ConvNeXt-T.** Stochastic depth [21] and Layer Scale [51] are added in the end to form the final model, ConvNeXt-T. The final MSE result for ConvNeXt-T is 0.0065.

Main findings. The 15 steps of architecture change disclose how the reconstruction result of gradient inversion attacks gets better and better, and eventually, the data sample could be clearly reconstructed. The attack results with more randomly selected samples are presented in Section 4.5. Please refer to Appendix B for more detailed architecture specifications.

3.4 Segmenting a Transformer Model to Analyze the Privacy Leakage

In this section, we segment a Transformer-based model and evaluate whether some layers of Transformers have an impact on privacy leakage. As there are only prediction results and model representations from membership inference attacks and attribute inference attacks, it is difficult to evaluate some specific layers using these attacks. Gradient inversion attacks require a list of gradients from the victim model, which is perfect for evaluation.

We use ViT-B as the victim model in our analysis and gradient inversion attacks as the attack tool. Instead of providing all the gradients to the attacks, we provide selected gradients. Figure 1 illustrates the architecture of a Vision Transformer (an example of ViT-B). We group the model into five modules based on the layer designs and then evaluate the impact of gradients from each module separately:

- Module 1: Stem layers. This module receives the input of the model and has patch embedding and position embedding layers.

- Module 2: Attention layers. They are in the Transformer block, and this module is the main reason why Transformers are different from CNNs.

- Module 3: MLP layers. They are also in the Transformer block.

- Module 4: Norm layers. They are located right before the attention layers and MLP layers. LayerNorm is often used as Norm layers in Transformers.

- Module 5: Head layers. They are the last few layers for producing the output of the model. A few fully connected layers could be used as head layers.

If the attack with Module A gradients has a higher attack accuracy than the one with Module B, it means Module A has a high chance of leaking more information about data samples than Module B.

Main findings. We test each attack case with only specific gradient layers provided and find that attention layers in Transformers also have an impact on privacy leaks. Please refer to Section 4.5 for more information.

4 Experimental Results

In this section, we will first illustrate the experimental settings and then present the detailed experimental results following the process described in Section 3.

Table 2: Training recipes for victim, shadow ,and attack models of membership inference attacks.

Config	Hyperparameter
optimizer	AdamW
learning rate	0.001
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
batch size	256
training epochs	300
learning rate schedule	cosine annealing
random augmentation	None
label smoothing	0.1

Table 3: Training recipes for victim and attack models of attribute inference attacks.

Config	Hyperparameter
optimizer	SGD
learning rate	0.01
weight decay	0.0005
optimizer momentum	0.9
batch size	256
training epochs	100
random augmentation	None

4.1 Settings

Datasets. In our experiments, we evaluate the privacy leakage under these five datasets.

- **CIFAR10 [23]** contains 60,000 color 32×32 images in 10 classes. Each class has 6,000 images and is a general object.
- **CIFAR100 [23]** is similar to CIFAR10 but with 100 classes. Each class has 600 images and is a general object.
- **ImageNet1K [7]** is a classic dataset with a collection of over 1 million labeled images in 1,000 classes.
- **Tiny-ImageNet [7]** is a subset of ImageNet1K with 200 classes.
- **CelebA [31]** contains over 200,000 face images, each with 40 binary attributes.

We use CIFAR10, CIFAR100, and Tiny-ImageNet in membership inference attacks. When we conduct the experiments in each dataset, we need to split the dataset evenly into four subsets for the training and testing of the victim and the shadow model.

We use CelebA in attribute inference attacks. The purpose of the attacks is to find hidden attributes, so we select CelebA for its abundant attribute labelling. We choose the gender attribute as the classification goal of the victim model, and we try to infer the race attribute. We randomly select 20,000 images of CelebA for the experiment, which splits evenly into four subsets for the training and testing of the victim and the attack model.

Table 4: Settings for gradient inversion attacks.

Config	Hyperparameter
cost function	similarity
optimizer	Adam
learning rate	0.1
total iteration	3000
total variance	0.0001

We use CIFAR10 and ImageNet1K in gradient inversion attacks. As these attacks occur in a federated learning scenario, and the attacks attempt to reconstruct the training batch, we randomly select some images of these datasets for evaluation.

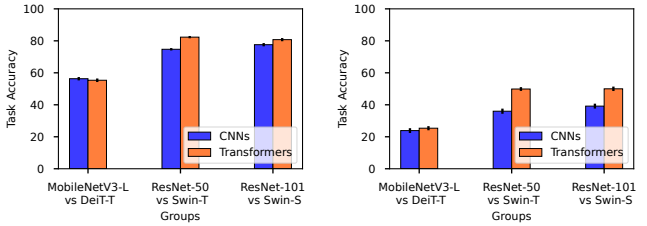
Victim Models. As models with similar parameter sizes should have comparable performances, we pair the CNNs and Transformers for comparisons. For membership inference attacks and attribute inference attacks, we select three groups of CNN and Transformer-based models based on the size of model parameters, including MobileNetV3-L, ResNet-50, ResNet-101, DeiT-T, Swin-T, and Swin-S. For gradient inversion attacks, we evaluate more models, including CNN-based MobileNetV3-L, ResNet variants, and ConvNeXt, as well as Transformer-based DeiTs, Swin Transformers, and ViTs.

Attack Models. For membership inference attacks, we use a three-layer MLP model to process the prediction results from the victim model. For attribute inference attacks, we use a two-layer MLP model to infer the victim model's representations. For gradient inversion attacks, it is an attack framework for optimizing the input gradients and the generated gradients.

Notations for the models. For ResNet variants, ResNet-number means how many layers of residual nets are in the model. ResNet-50x3 has network channels three times wider than ResNet-50. For other models, we append -T, -S, -B, -L to indicate the *tiny*, *small*, *base*, *large* version of the model architecture. For example, Swin-T means the tiny verison of the Swin Transformer.

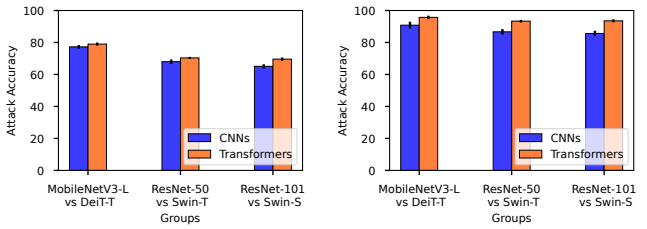
Evaluation Metrics. For membership inference attacks and attribute inference attacks, we use attack accuracy to evaluate the performance. The higher the accuracy is, the more effective the attack is. For gradient inversion attacks, we apply the mean square error (MSE) and the peak signal-to-noise ratio (PSNR) to check the reconstruction results. Smaller MSE and larger PSNR values indicate better reconstruction results (higher attack performance).

Training Settings for Privacy Attacks. Table 2, Table 3 and Table 4 illustrate the training configurations for membership inference attacks, attribute inference attacks and gradient inversion attacks.



(a) Victim models on CIFAR10

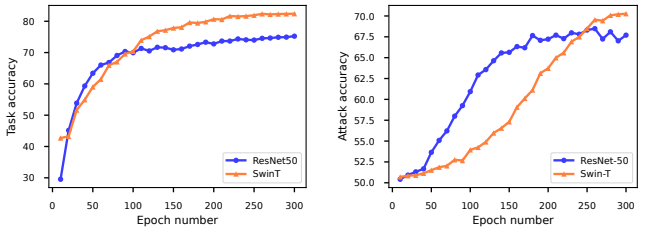
(b) Victim models on CIFAR100



(c) Attack on CIFAR10

(d) Attack on CIFAR100

Figure 2: Results of membership inference attacks. (a), (b): The performance of classification tasks (victim models) for both CNNs and Transformers; (c), (d): Privacy attack accuracy of membership inference attacks.



(a) Performance of victim models on CIFAR10

(b) Performance of privacy attack on CIFAR10

Figure 3: The performance of membership inference attacks against ResNet-50 and Swin-T on CIFAR10 under different numbers of epochs.

4.2 Evaluation on Membership Inference Attacks

Figure 2 shows the performance of membership inference attacks on CIFAR10 and CIFAR100. The accuracy of victim models on CIFAR10 is given in Figure 2a, showing that Transformers have competitive performances compared with CNN models. The accuracy of the attacks on CIFAR10 in Figure 2c illustrates more severe privacy leakage of Transformers than CNN models in every group. Figure 2b and Figure 2d also show the same results that Transformers are more vulnerable to membership inference attacks than CNN models.

Due to the fact that the dataset is divided into four even subsets, the performance of victim models in Figure 2a and in Figure 2b may not have the task accuracy as high as a standard CIFAR10 and CIFAR100 classification task. Consequently, it is natural to have a lower task accuracy with only a subset of the original dataset. Some prior research has also proven this phenomenon [16, 22, 41], and this does not affect the performance of the attacks.

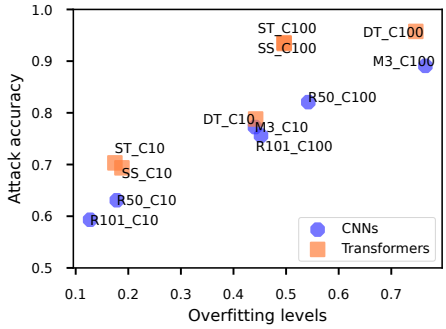


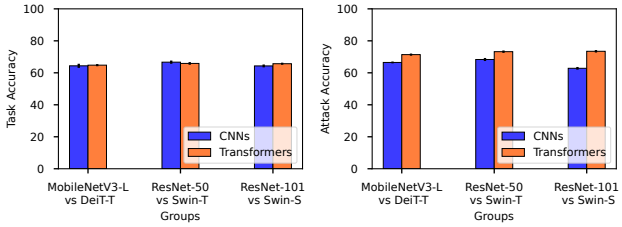
Figure 4: The performance of membership inference attacks against both CNNs and Transformers with various models and datasets under different overfitting levels. C10 for CIFAR10, C100 for CIFAR100, M3 for MobileNetV3-L, DT for DeiT-T, R50 for ResNet-50, ST for Swin-T, R101 for ResNet-101, SS for Swin-S.

Then, we evaluate the victim model accuracy and the attack accuracy of Swin-T and ResNet-50 in 300 epochs in CIFAR10 Figure 3. For the victim model accuracy, Swin-T outperforms ResNet-50 after 100 epochs and reaches an accuracy of over 80, while ResNet-50 stays at an accuracy of around 75. For the attack accuracy, the attack on Swin-T does not gain high accuracy results at first but can eventually surpass ResNet-50.

The privacy leakage varies on different models and datasets. Similar to [40, 42], we analyze the overfitting levels of victims models in Figure 4. The overfitting level indicates the accuracy difference of a model between its training and inference. Figure 4 illustrates the results of Transformers and CNNs in CIFAR10 and CIFAR100. We conclude that a more overfitted model comes with higher membership inference attack accuracy. More importantly, at the same overfitting level, Transformers always suffer from higher attack accuracy.

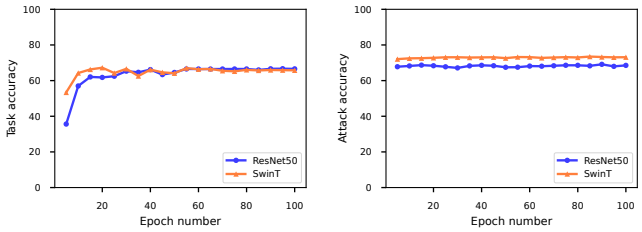
4.3 Evaluation on Attribute Inference Attacks

Figure 5 provides the performance of attribute inference attacks on CelebA. Like the evaluation on membership inference attacks, we still pair the models of CNNs and Transformers into three groups by their parameter sizes. We can see that in each group, CNNs and Transformers have similar performance in terms of task accuracy, while Transformers consistently outperform CNNs in terms of attack accuracy. This means Transformers are more vulnerable to attribute inference attacks than CNNs, just like what we concluded in the previous evaluation.



(a) Victim models on CelebA (b) Attack on CelebA

Figure 5: Results of attribute inference attacks. (a): The performance of classification tasks (victim models) for both CNNs and Transformers; (b): Privacy attack accuracy of attribute inference attacks.



(a) Performance of victim models on CelebA (b) Performance of privacy attack on CelebA

Figure 6: The performance of attribute inference attacks against ResNet-50 and Swin-T on CelebA under different numbers of epochs.

We then illustrate the performance of task accuracy and attack accuracy in attribute inference attacks in 100 epochs with models of ResNet-50 and Swin-T on CelebA Figure 6. In Figure 6a, although the starting task accuracy is different with ResNet-50 and Swin-T, these models can reach similar task accuracies at around 65. The attack accuracy in Figure 6b shows a different view: the attack accuracy on Swin-T is always higher than that on ResNet-50 during the training in 100 epochs. This reveals that Transformers like Swin-T are more vulnerable to attribute inference attacks than ResNet-50 from the start of the training to the end.

We further analyze the relationship between the attack performance and the overfitting levels of victim models in Figure 7. We have made a similar discovery to the previous evaluation: Transformers suffer from higher attack accuracy than CNNs when the victim models are at the same overfitting level.

4.4 Evaluation on Gradient Inversion Attacks

Figure 8 illustrates the performance of gradient inversion attacks against both types of model architectures, respectively. For CNNs, the attack on MobileNetV3-L fails to present a good reconstruction result. Only ResNet-18 of all ResNet

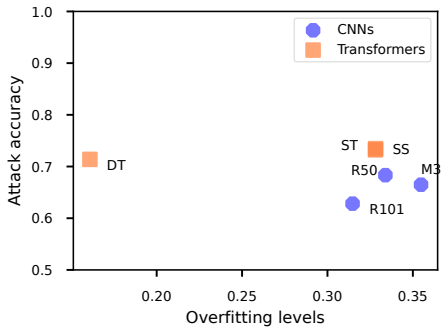


Figure 7: The performance of attribute inference attacks against various models of CNNs and Transformers under different overfitting levels. M3 for MobileNetV3-L, DT for DeiT-T, R50 for ResNet-50, ST for Swin-T, R101 for ResNet-101, SS for Swin-S.

Table 5: The mean and standard deviation MSE and PSNR results of gradient inversion attacks against several selected models from both CNNs and Transformers on 50 randomly selected samples in CIFAR10.

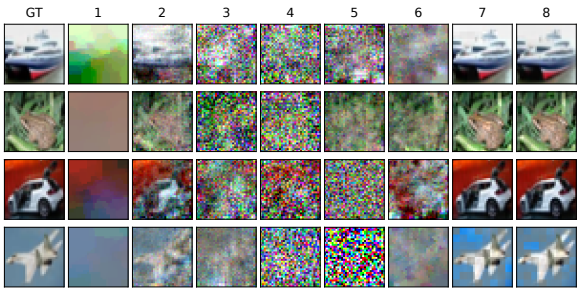
Models	**1. MobileNetV3-L**	**2. ResNet-18**	**3. ResNet-50**	**4. ResNet-50x3**
MSE	0.8892 ± 0.4889	0.3416 ± 0.1078	1.3308 ± 0.6507	1.5290 ± 0.3837
PSNR	13.31 ± 2.86	16.94 ± 1.57	11.30 ± 2.24	10.33 ± 1.21
Models	**5. ResNet-101**	**6. ResNet-152**	**7. ConvNeXt-T**	**8. ConvNeXt-S**
MSE	1.2557 ± 0.6829	0.6983 ± 0.4619	0.0207 ± 0.0251	0.0267 ± 0.0205
PSNR	11.58 ± 2.16	14.43 ± 2.66	31.38 ± 5.00	29.41 ± 4.37
Models	**9. DeiT-T**	**10. DeiT-S**	**11. DeiT-B**	**12. Swin-T**
MSE	0.9993 ± 0.5333	0.5824 ± 0.3524	1.2399 ± 0.7446	0.0069 ± 0.0071
PSNR	12.62 ± 2.31	15.97 ± 4.75	11.82 ± 2.48	36.24 ± 5.21
Models	**13. Swin-S**	**14. Swin-B**	**15. ViT-B**	**16. ViT-L**
MSE	0.0063 ± 0.0083	0.0098 ± 0.0093	0.0007 ± 0.0003	0.0004 ± 0.0001
PSNR	37.85 ± 6.15	35.19 ± 5.98	43.70 ± 1.84	46.35 ± 2.20

variants can provide an acceptable reconstruction image. The attack on the new state-of-the-art models, ConvNeXt, surprisingly receives good performance. For Transformers, the attack on most of the models receives good performance, except for DeiT, the lightweight Transformers.

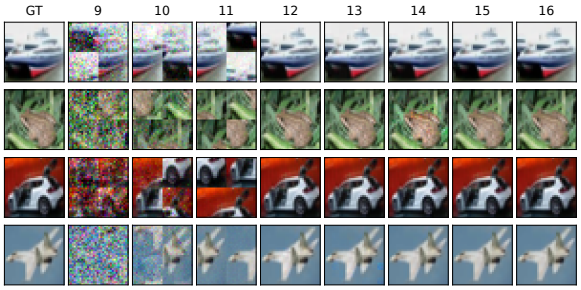
Table 5 provides the corresponding MSE and PSNR results of these attacks on 50 randomly selected samples in CIFAR10. The results match the ones in Figure 8, where low MSE and high PSNR values mean better reconstruction images.

In Figure 9, we can see how a raw dummy image becomes close to the original image as the training continues. We demonstrate the attack performance on several models. For ResNet-50, the reconstruction result shows little information on the original image, but the result on ResNet-101 shows nothing leading to a failed reconstruction. For ConvNeXt-T and some Transformers like Swin-T and ViT-B, it is easy to receive an excellent attack result.

Then we make more evaluations on gradient inversion attacks on ImageNet1K. We choose ResNet-50, ResNet-101, ConvNeXt-T as CNN representatives, and Swin-T, ViT-B as



(a) CNNs



(b) Tranformers

Figure 8: The performance of gradient inversion attacks against both CNNs and Transformers on randomly selected samples in CIFAR10. The sequence of models corresponds to that listed in Table 5 (i.e. 1 to 8 for CNNs and 9 to 16 for Transformers).

Transformer representatives. We randomly select some images from ImageNet1K and provide the reconstruction results, shown in Figure 10. Just like previous experiments on CIFAR10, the attacks on ResNet variants are not successful. The attacks on Transformers and Transformer-style ConvNeXt can produce better reconstruction results.

Our experiments show that most Transformers tend to be more vulnerable to gradient inversion attacks than CNNs. One exceptional example is ConvNeXt, which is a CNN-based model, but its design is heavily influenced by Vision Transformers. It raises a fundamental question of whether some architecture features in ConvNeXt or other Transformers can boost the privacy attack performance.

4.5 Which Architecture Features Can Lead to Privacy Leakage?

In order to determine which architecture features cause the privacy leakage, we modify the model architecture step by step from ResNet-50 to ConvNeXt (as described in Subsection 3.3) and evaluate the performance of gradient inversion attacks on these models. Figure 11 and Table 6 show the performance of gradient inversion attacks on each model architecture. The overall trend is that the reconstruction results are getting better with the architecture changing.

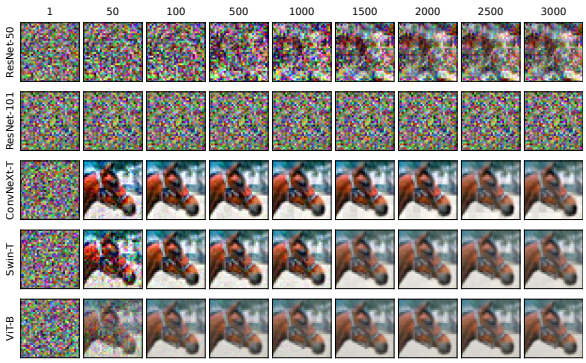


Figure 9: The performance of gradient inversion attacks on several models with different iteration numbers (i.e. 1, 50, 100, 500, 1000, 1500, 2000, 2500, 3000).

Table 6: The MSE and PSNR results of gradient inversion attacks on model architectures from 15 steps. Some significant drops of MSE results are marked bold.

Num	Steps	MSE	PSNR
1	ResNet-50	1.5096 ± 0.5538	10.58 ± 1.87
2	Changing channel dimensions	1.4706 ± 0.5710	10.74 ± 1.97
3	Change stage compute ratio	1.5286 ± 0.5246	10.56 ± 2.05
4	Patchify	**0.9011 ± 0.4376**	**12.97 ± 2.10**
5	ResNeXtify	1.2415 ± 0.6934	11.86 ± 2.77
6	Inverted bottleneck	1.1123 ± 0.4994	12.06 ± 2.19
7	Enlarging kernel sizes	0.8206 ± 0.3543	13.40 ± 2.30
8	Forming new stem	**0.5684 ± 0.3564**	**15.43 ± 3.01**
9	ReLU to GELU	1.0540 ± 0.5075	12.42 ± 2.61
10	Removing activation layers	**0.0215 ± 0.0150**	**29.93 ± 3.58**
11	Removing BN layers	0.0198 ± 0.0139	30.57 ± 4.12
12	BN to LN	0.2247 ± 0.0870	18.80 ± 1.62
13	Adding Bias	**0.0049 ± 0.0044**	**36.86 ± 3.96**
14	Seperating downsampling layers	0.0121 ± 0.0171	33.79 ± 4.69
15	ConvNeXt	0.0177 ± 0.0171	31.88 ± 5.04

Figure 11 provides qualitative results of the attacks. At the beginning of the steps (Steps 1 to 3), the attacks fail to reconstruct proper images. In the middle stages, something can be seen from the reconstruction results but not much. The reconstruction results improve in the later stages (After Step 10). At last, using ConvNeXt-T, which is step 15, the attacks can have a good attack performance.

Table 6 presents more information on the performance of gradient inversion attacks. There are four significant increases in attacking accuracy when the architecture changes step by step: the first one occurs when "Patchify" is applied; the second one happens when new stem layers are formed; the third one occurs when some activation layers are removed; the fourth one happens when bias parameters are added to the convolutional layers. We make some further ablation studies to verify our observations.

One of the differences between a Transformer block and a ResNet block is that a Transformer block has fewer activation layers. We leave fewer activation layers in the model, which boosts the attack performance. Figure 12 provides the comparison of the gradient inversion attack performance when

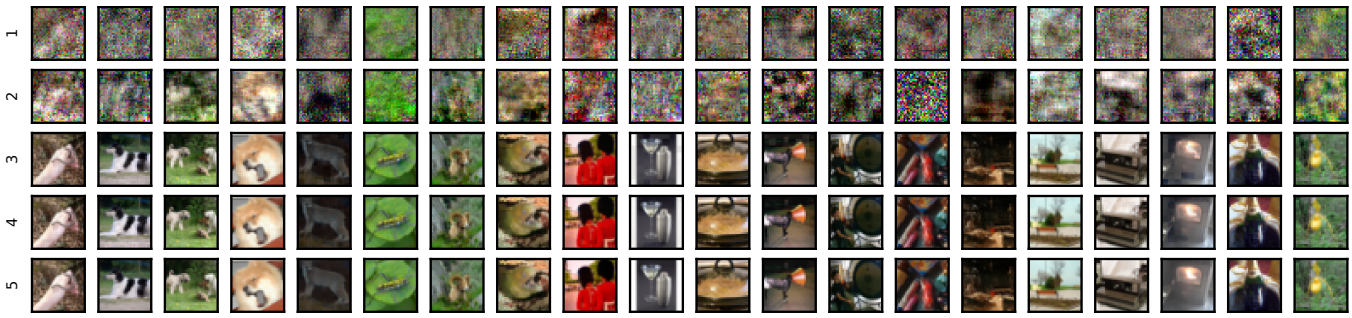


Figure 10: The performance of gradient inversion attacks on several selected models on ImageNet1K. Each line presents the results from the attacks on one model. The number on the left (1 to 5) corresponds to the model listed here (ResNet-50, ResNet-101, ConvNeXt-T, Swin-T, ViT-B).

Figure 11: The performance of gradient inversion attacks on each architecture changing from ResNet-50 to ConvNeXt-T with several selected iterations (i.e. 1, 50, 100, 1000, 3000). Model architectures from 15 steps are shown.

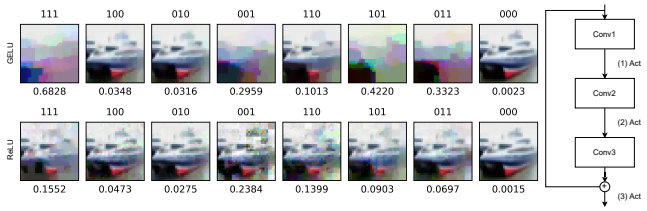


Figure 12: The performance of gradient inversion attacks on models with different positions of activation layers. The position of three activation layers is illustrated on the right, with two between convolutional layers and one after the addition operation. The three digits on the top of the subfigures show whether the activation layer on this position is added or not. The bottom of each subfigure provides MSE values for the attack on this model (i.e. models with GELU or ReLU).

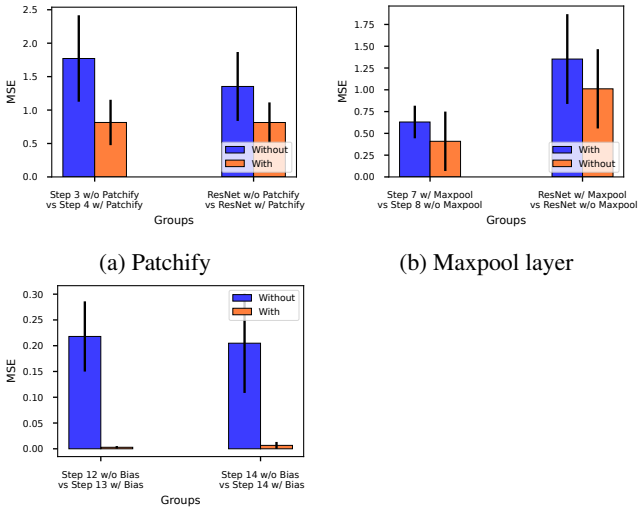
different activate layers are removed. We can see that the attack performance gets better when the third activation layer is removed. This layer is located after the skip connection of the ResNet block. We believe that this layer acts as a non-linear process and reduces the information from the input, which worsens the attack performance. It also illustrates that changing ReLU to GELU does not help to improve the attack performance, and the best reconstruction results are when all the activation layers are removed.

We have also identified other features that could boost the performance of gradient inversion attacks. Figure 13 shows three other features that may have an impact on the model privacy leakage. Figure 13a and Figure 13b illustrate that the usage of "Patchify" and removing maxpool layers reduce the reconstruction MSE results. As these procedures both make changes to the first few stem layers, we believe that stem layers are essential for privacy attacks, and these features contribute to making the model vulnerable to the attacks. Figure 13c shows that adding bias to the layers could get better reconstruction results. As a result, adding bias also makes a contribution to the model privacy leakage.

Table 7 shows the reconstruction results of gradient inversion attacks when only gradients of selected layers are given to the attacks. "All layers" is the attack with gradients of all the layers, which is the default attack. The stem layers contain the process of patch embedding and position embedding, which are only some transformations and the output of stem layers does not change much from the original image sample. As a result, the attack with "stem layers" has the reconstruction result nearly the same as the original image sample, and we rule this out for comparison. For the attacks with other selected layers, the attack with the best attack performance is the attack with "attention layers," with an MSE of 0.0020 and a PSNR of 39.61. This reveals that attention layers could be easier to be attacked than other layers.

5 Discussion

In the previous section, we have discovered that four component designs in Transformers could result in privacy leakage:



(a) Patchify

(b) Maxpool layer

(c) Bias

Figure 13: The performance of gradient inversion attacks on model architectures with or without some features.

Table 7: The performance of gradient inversion attacks when segmenting ViT-B to make selection of gradients.

Layers	Number of layers	MSE	PSNR
All layers	152	0.0007 ± 0.0003	43.70 ± 1.84
Stem layers	4	0.0000 ± 0.0000	67.43 ± 5.03
Attention layers	48	0.0020 ± 0.0009	39.61 ± 2.76
MLP layers	48	0.0036 ± 0.0016	36.98 ± 2.59
Norm layers	48	0.0040 ± 0.0018	36.57 ± 2.56
Head layers	4	0.2776 ± 0.2312	19.01 ± 3.89

activation layers, stem layers, bias parameters, and attention modules. In this section, we would like to provide a more in-depth discussion on these modules.

The design of activation layers, stem layers, and bias parameters. These are micro design components with the potential to leak sensitive data from input samples. Activation layers such as ReLU and GELU add a layer of complexity to the model by making it a non-linear function. Removing some activation layers simplifies the logic of attacks. Stem layers receive an input sample and perform some preliminary processing. As the representation after the stem layers remains similar to the original input image, there is a high possibility of extracting private information because of the design of stem layers. Bias parameters act as another non-linear operation to the model layers. As the gradient inversion attack is an optimisation process that calculates gradient loss at each iteration, adding bias to the model layers is likely to aid the optimisation process and allow the adversary to reconstruct data samples faster.

The design of attention modules. The receptive field of a model refers to the information received within a specified range by a neuron in a model layer. In a fully connected neural network, each neuron is connected to the element values of the entire input sample. Due to the convolution operation, the neuron in a convolutional network receives the values from its receptive field. The range of the receptive field is defined by the convolution templates in CNNs, and it has a theoretical limit. Some researchers have demonstrated that the effective receptive field (i.e. the effective area in the receptive field) is actually smaller than the theoretical receptive field [34]. From a privacy perspective, a CNN model could only reveal part of sensitive information from the input sample due to the design of convolution templates.

Transformers employ the multi-head self-attention mechanism, also known as attention modules. The input sample is taken as a sequence of flattened 2D patches. The attention module receives the input sequence and generates its representation of the sequence by mapping the query and the key-value pairs to the output. Transformers tend to have much larger receptive fields than CNNs due to the fact that their attention module is computed with the entire input sequence [9, 53]. In terms of privacy, a Transformer model is able to extract more sensitive information than a CNN model. Hence, Transformers are more prone to attacks than CNNs, as demonstrated by our evaluation based on three privacy attacks.

The privacy leakage issue of Transformers indicates that Transformers need more privacy treatment than CNNs. When the perturbation is applied to model parameters as a defense mechanism (i.e. differential privacy noises), more noises can be added to those "privacy-leakage" layers (i.e., activation layers, stem layers, bias parameters, and attention modules). Thus, the total level of perturbation could be lowered, while achieving a satisfactory privacy protection. This unequal noise perturbation method across various Transformer modules will be investigated in our future research.

Other factors - Overfitting. Previous work claimed that privacy attacks are mostly caused by the undesirable overfitting issue in deep learning models [28, 42]. Overfitting normally occurs when a model performs well on the training data, but poorly on the validation data. The overfitting issue tends to become severe on an over-trained model with a large number of parameters Deep learning models are exposed to privacy threats due to the overfitting effect. In our work, we find that model architectures have impacts on the performance of privacy attacks, which can *not* be attributed solely to the overfitting effect. Indeed, our experiments validate that the variation in performance is due to the difference in model architectures. For models with the same level of parameter sizes, Transformers tend to be more vulnerable to privacy attacks than CNNs. More importantly, for models with the same overfitting level, our conclusion still holds that Transformers are more vulnerable to privacy attacks than CNNs. We have

then identified some architecture features that could lead to privacy leakage.

6 Conclusion

In this study, for the first time, we perform privacy analysis on model architectures, especially CNNs and Transformers. We have conducted a comparison of three prominent privacy attacks, i.e., membership inference attacks, attribute inference attacks, and gradient inversion attacks. Our analysis indicates that Transformers tend to be more vulnerable to privacy attacks than CNNs. However, many Transformers-inspired CNN designs, such as ConvNeXt, are also susceptible to privacy threats. A number of Transformers' features, including the design of activation layers, the design of stem layers, the design of bias parameters, and the attention modules, may have incurred privacy risks.

It is still challenging to establish accurate and theoretical explanations for why certain architectural features are critical to privacy preservation. We believe that these analyses require further experimental campaigns, and we intend to study this in our future work.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer Normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Yutong Bai, Jieru Mei, Alan L Yuille, and Cihang Xie. Are Transformers more robust than CNNs? In *Advances in Neural Information Processing Systems*, volume 34, pages 26831–26843, 2021.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

[4] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 343–362, 2020.

[5] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. When Machine Unlearning Jeopardizes Privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 896–911, 2021.

[6] Christopher A. Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-Only Membership Inference Attacks. In *International Conference on Machine Learning*, pages 1964–1974, 2021.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*, 2021.

[10] Vasisht Duddu and Antoine Boutet. Inferring Sensitive Attributes from Model Explanations. *arXiv preprint arXiv:2208.09967*, 2022.

[11] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[12] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting Gradients - How easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947, 2020.

[13] Ali Hatamizadeh, Hongxu Yin, Holger R. Roth, Wenqi Li, Jan Kautz, Daguang Xu, and Pavlo Molchanov. GradViT: Gradient Inversion of Vision Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10021–10030, 2022.

[14] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. LOGAN: Membership Inference Attacks Against Generative Models. *Proceedings on Privacy Enhancing Technologies*, 2019:133–152, 2019.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[16] Xinlei He and Yang Zhang. Quantifying and Mitigating Privacy Risks of Contrastive Learning. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 845–863, 2021.

[17] Yang He, Shadi Rahimian, Bernt Schiele, and Mario Fritz. Segmentations-Leak: Membership Inference Attacks and Defenses in Semantic Image Segmentation. In *Computer Vision – ECCV 2020*, pages 519–535, 2020.

[18] Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

[19] Andrew Howard, Mark Sandler, Bo Chen, Weijun Wang, Liang-Chieh Chen, Mingxing Tan, Grace Chu, Vijay Vasudevan, Yukun Zhu, Ruoming Pang, Hartwig Adam, and Quoc Le. Searching for MobileNetV3. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1314–1324, 2019.

[20] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S. Yu, and Xuyun Zhang. Membership Inference Attacks on Machine Learning: A Survey. *ACM Computing Surveys*, 2022.

[21] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep Networks with Stochastic Depth. In *Computer Vision – ECCV 2016*, pages 646–661, 2016.

[22] Yigitcan Kaya and Tudor Dumitras. When Does Data Augmentation Help With Membership Inference Attacks? In *International Conference on Machine Learning*, pages 5345–5355, 2021.

[23] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25, 2012.

[25] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1:541–551, 1989.

[26] Zheng Li and Yang Zhang. Membership Leakage in Label-Only Exposures. In *ACM SIGSAC Conference on Computer and Communications Security (CCS 2021)*, 2021.

[27] Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. When Machine Learning Meets Privacy: A Survey and Outlook. *ACM Computing Surveys*, 54:31:1–31:36, 2021.

[28] Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. ML-Doctor: Holistic Risk Assessment of Inference Attacks Against Machine Learning Models. In *31st USENIX Security Symposium (USENIX Security 22)*, 2022.

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021.

[30] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *2022 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[31] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep Learning Face Attributes in the Wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015.

[32] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019.

[33] Jiahao Lu, Xi Sheryl Zhang, Tianli Zhao, Xiangyu He, and Jian Cheng. APRIL: Finding the Achilles' Heel on Privacy for Vision Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10051–10060, 2022.

[34] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the Effective Receptive Field in Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[35] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting Unintended Feature Leakage in Collaborative Learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706, 2019.

[36] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 739–753, 2019.

[37] Sayak Paul and Pin-Yu Chen. Vision Transformers Are Robust Learners. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36:2071–2081, 2022.

[38] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollar. Designing Network Design Spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020.

[39] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do Vision Transformers See Like Convolutional Neural Networks? In *Advances in Neural Information Processing Systems*, volume 34, pages 12116–12128, 2021.

[40] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. ML-Leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019*, 2019.

[41] Virat Shejwalkar and Amir Houmansadr. Membership Privacy for Machine Learning Models Through Knowledge Transfer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:9549–9557, 2021.

[42] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017.

[43] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2015.

[44] Congzheng Song and Ananth Raghunathan. Information Leakage in Embedding Models. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 377–390, 2020.

[45] Congzheng Song and Vitaly Shmatikov. Overlearning Reveals Sensitive Attributes. In *International Conference on Learning Representations*, 2020.

[46] Liwei Song and Prateek Mittal. Systematic Evaluation of Privacy Risks of Machine Learning Models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2615–2632, 2021.

[47] Andreas Peter Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your ViT? Data, augmentation, and regularization in vision transformers. *Transactions on Machine Learning Research*, 2022.

[48] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[49] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[50] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the 38th International Conference on Machine Learning*, pages 10347–10357, 2021.

[51] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with Image Transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 32–42, 2021.

[52] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Demystifying Membership Inference Attacks in Machine Learning as a Service. *IEEE Transactions on Services Computing*, pages 1–1, 2019.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[54] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017.

[55] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[56] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M. Alvarez, Jan Kautz, and Pavlo Molchanov. See Through Gradients: Image Batch Recovery via GradInversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16337–16346, 2021.

[57] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis E. H. Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 538–547, 2021.

[58] Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks, 2017.

[59] Guangsheng Zhang, Bo Liu, Tianqing Zhu, Ming Ding, and Wanlei Zhou. Label-Only Membership Inference Attacks and Defenses In Semantic Segmentation Models. *IEEE Transactions on Dependable and Secure Computing*, pages 1–1, 2022.

[60] Guangsheng Zhang, Bo Liu, Tianqing Zhu, Andi Zhou, and Wanlei Zhou. Visual privacy attacks and defenses in deep learning: A survey. *Artificial Intelligence Review*, 55:4347–4401, 2022.

[61] Rui Zhang, Song Guo, Junxiao Wang, Xin Xie, and Dacheng Tao. A Survey on Gradient Inversion: Attacks, Defenses and Future Directions. In *Thirty-First International Joint Conference on Artificial Intelligence*, volume 6, pages 5678–5685, 2022.

[62] Benjamin Zi Hao Zhao, Aviral Agrawal, Catisha Coburn, Hassan Jameel Asghar, Raghav Bhaskar, Mohamed Ali Kaafar, Darren Webb, and Peter Dickinson. On the (In)Feasibility of Attribute Inference Attacks on Machine Learning Models. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 232–251, 2021.

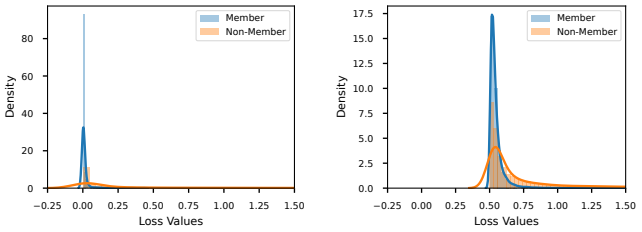
[63] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. iDLG: Improved Deep Leakage from Gradients. *arXiv preprint arXiv:2001.02610*, 2020.

[64] Ligeng Zhu, Zhijian Liu, and Song Han. Deep Leakage from Gradients. *Advances in Neural Information Processing Systems*, 32, 2019.

[65] Yang Zou, Zhikun Zhang, Michael Backes, and Yang Zhang. Privacy Analysis of Deep Learning in the Wild: Membership Inference Attacks against Transfer Learning. *arXiv preprint arXiv:2009.04872*, 2020.

A Additional Experimental Results for Membership Inference Attacks

To further analyze why Transformers are more vulnerable than CNN models, we plot the loss distributions between the member and non-member data for both CNNs and Transformers in Figure 14. We can see that the distributions of member and non-member data for CNNs are more concentrated at 0,



(a) Loss distributions on ResNet-50 (b) Loss distributions on Swin-T

Figure 14: (a): The loss distributions of membership inference attacks against ResNet-50 and Swin-T on CIFAR10.

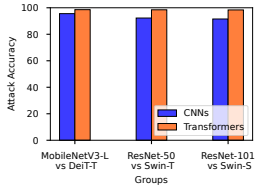


Figure 15: The attack performance of membership inference attacks on Tiny-ImageNet.

and the distributions for Transformers are concentrated at 0.5. This is why the attacks against CNNs and Transformers have different performances.

For membership inference attacks on Tiny-ImageNet, we choose MobileNetV3-L, ResNet-50, ResNet-101 as CNN representatives, and DeiT-T, Swin-T, Swin-S as Transformer representatives. The experimental results in Figure 15 are the same as those on CIFAR10 and CIFAR100, which is that Transformers tend to be more vulnerable to privacy attacks than CNNs. The attack results on Transformers outperforms those on CNNs in each comparison group.

B Detailed Architecture Specifications for Experiments

We present a detailed architecture comparison between models from 15 steps (changing from ResNet-50 to ConvNeXt-T) in Table 8. Changes are marked bold. See Section 3.3 for explanations of these model architectures.

Table 8: Detailed Architecture Specifications for 15 steps in Section 3.3.

	ResNet-50, Step 1	Step 2	Step 3	Step 4	Step 5
stem	7 × 7, 64, stride 2 3 × 3, max pool, stride 2	7 × 7, **96**, stride 2 3 × 3, max pool, stride 2	7 × 7, 96, stride 2 3 × 3, max pool, stride 2	**4** × **4**, 96, stride **4** 3 × 3, max pool, stride 2	4 × 4, 96, stride 4 3 × 3, max pool, stride 2
block1	[1 × 1, 64 3 × 3, 64 1 × 1, 256] × 3	[1 × 1, **96** 3 × 3, **96** 1 × 1, **384**] × 3	[1 × 1, 96 3 × 3, 96 1 × 1, 384] × **3**	[1 × 1, 96 3 × 3, 96 1 × 1, 384] × 3	[1 × 1, 96 **d3** × **3**, 96 1 × 1, 384] × 3
block2	[1 × 1, 128 3 × 3, 128 1 × 1, 512] × 4	[1 × 1, **192** 3 × 3, **192** 1 × 1, **768**] × 4	[1 × 1, 192 3 × 3, 192 1 × 1, 768] × **3**	[1 × 1, 192 3 × 3, 192 1 × 1, 768] × 3	[1 × 1, 192 **d3** × **3**, 192 1 × 1, 768] × 3
block3	[1 × 1, 256 3 × 3, 256 1 × 1, 1024] × 6	[1 × 1, **384** 3 × 3, **384** 1 × 1, **1536**] × 6	[1 × 1, 384 3 × 3, 384 1 × 1, 1536] × **9**	[1 × 1, 384 3 × 3, 384 1 × 1, 1536] × 9	[1 × 1, 384 **d3** × **3**, 384 1 × 1, 1536] × 9
block4	[1 × 1, 512 3 × 3, 512 1 × 1, 2048] × 3	[1 × 1, **768** 3 × 3, **768** 1 × 1, **3072**] × 3	[1 × 1, 768 3 × 3, 768 1 × 1, 3072] × **3**	[1 × 1, 768 3 × 3, 768 1 × 1, 3072] × 3	[1 × 1, 768 **d3** × **3**, 768 1 × 1, 3072] × 3
other specs	ReLU, BN	ReLU, BN	ReLU, BN	ReLU, BN	ReLU, BN
-	-	-	-	-	-
	Step 6	Step 7	Step 8	Step 9	Step 10
stem	4 × 4, 96, stride 4 3 × 3, max pool, stride 2	4 × 4, 96, stride 4 3 × 3, max pool, stride 2	4 × 4, 96, stride 4 **(removed)**	4 × 4, 96, stride 4	4 × 4, 96, stride 4
block1	[1 × 1, 96 d3 × 3, **384** 1 × 1, **96**] × 3	[**d7** × **7**, 96 1 × 1, 384 1 × 1, 96] × 3	[d7 × 7, 96 1 × 1, 384 1 × 1, 96] × 3	[d7 × 7, 96 1 × 1, 384 1 × 1, 96] × 3	[d7 × 7, 96 1 × 1, 384 1 × 1, 96] × 3
block2	[1 × 1, 192 d3 × 3, **768** 1 × 1, **192**] × 3	[**d7** × **7**, 192 1 × 1, 768 1 × 1, 192] × 3	[d7 × 7, 192 1 × 1, 768 1 × 1, 192] × 3	[d7 × 7, 192 1 × 1, 768 1 × 1, 192] × 3	[d7 × 7, 192 1 × 1, 768 1 × 1, 192] × 3
block3	[1 × 1, 384 d3 × 3, **1536** 1 × 1, **384**] × 9	[**d7** × **7**, 384 1 × 1, 1536 1 × 1, 384] × 9	[d7 × 7, 384 1 × 1, 1536 1 × 1, 384] × 9	[d7 × 7, 384 1 × 1, 1536 1 × 1, 384] × 9	[d7 × 7, 384 1 × 1, 1536 1 × 1, 384] × 9
block4	[1 × 1, 768 d3 × 3, **3072** 1 × 1, **768**] × 3	[**d7** × **7**, 768 1 × 1, 3072 1 × 1, 768] × 3	[d7 × 7, 768 1 × 1, 3072 1 × 1, 768] × 3	[d7 × 7, 768 1 × 1, 3072 1 × 1, 768] × 3	[d7 × 7, 768 1 × 1, 3072 1 × 1, 768] × 3
other specs	ReLU, BN	ReLU, BN	ReLU, BN	**GELU**, BN	**Fewer GELU**, BN
-	-	-	-	-	-
	Step 11	Step 12	Step 13	Step 14	ConvNeXt-T, Step 15
stem	4 × 4, 96, stride 4	4 × 4, 96, stride 4	4 × 4, 96, stride 4	4 × 4, 96, stride 4	4 × 4, 96, stride 4
block1	[d7 × 7, 96 1 × 1, 384 1 × 1, 96] × 3	[d7 × 7, 96 1 × 1, 384 1 × 1, 96] × 3	[d7 × 7, 96 1 × 1, 384 1 × 1, 96] × 3	[d7 × 7, 96 1 × 1, 384 1 × 1, 96] × 3	[d7 × 7, 96 1 × 1, 384 1 × 1, 96] × 3
sep ds				**2** × **2, 192, stride 2**	2 × 2, 192, stride 2
block2	[d7 × 7, 192 1 × 1, 768 1 × 1, 192] × 3	[d7 × 7, 192 1 × 1, 768 1 × 1, 192] × 3	[d7 × 7, 192 1 × 1, 768 1 × 1, 192] × 3	[d7 × 7, 192 1 × 1, 768 1 × 1, 192] × 3	[d7 × 7, 192 1 × 1, 768 1 × 1, 192] × 3
sep ds				**2** × **2, 384, stride 2**	2 × 2, 384, stride 2
block3	[d7 × 7, 384 1 × 1, 1536 1 × 1, 384] × 9	[d7 × 7, 384 1 × 1, 1536 1 × 1, 384] × 9	[d7 × 7, 384 1 × 1, 1536 1 × 1, 384] × 9	[d7 × 7, 384 1 × 1, 1536 1 × 1, 384] × 9	[d7 × 7, 384 1 × 1, 1536 1 × 1, 384] × 9
sep ds				**2** × **2, 768, stride 2**	2 × 2, 768, stride 2
block4	[d7 × 7, 768 1 × 1, 3072 1 × 1, 768] × 3	[d7 × 7, 768 1 × 1, 3072 1 × 1, 768] × 3	[d7 × 7, 768 1 × 1, 3072 1 × 1, 768] × 3	[d7 × 7, 768 1 × 1, 3072 1 × 1, 768] × 3	[d7 × 7, 768 1 × 1, 3072 1 × 1, 768] × 3
other specs	Fewer GELU, **Fewer BN**	Fewer GELU, **Fewer LN**	Fewer GELU, Fewer LN **Conv w/ True bias**	Fewer GELU, Fewer LN Conv w/ True bias	Fewer GELU, Fewer LN Conv w/ True bias **Stochastic depth, Layer Scale**