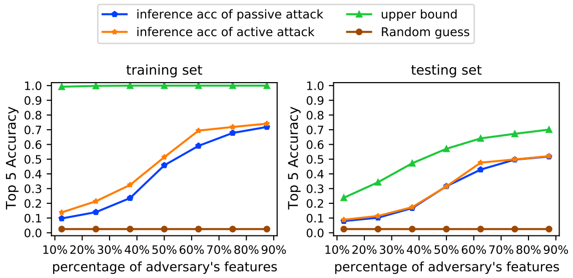

Figure 1: Impact of the quantity of the adversary's features. The experiment is conducted on TinyImageNet. The size of an image sample in TinyImageNet is 64×64 pixels. We split the image vertically, e.g., "25%" means that the adversary owns the left 1/4 of an image, to observe how the feature quantity influence the performance of the passive/active label inference attack. The upper bound is obtained using all the labels to directly train an inference model with the adversary's features.

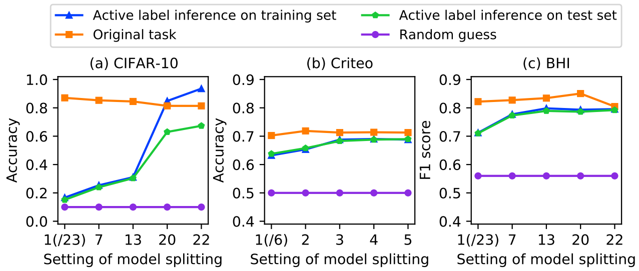

Figure 2: Impact of the complexity of the bottom model. X-axis represents the number of layers in the adversary's bottom model. For example, 1/23 means that the whole federated model consists of 23 layers, and the adversary's bottom model consists of 1 layer. Y-axis represents the corresponding label inference accuracy (F1 score).

References

[1] F. Li, A. Karpathy, and J. Johnson. Tiny ImageNet. https://www.kaggle.com/c/tiny-imagenet.

Appendix

A Details About Datasets

TinyImageNet. TinyImageNet [1] contains 110,000 images of 200 classes (550 for each class) downsized to 64×64 colored images. Following the common practice of using TinyImageNet, we take 100,000 samples as the training dataset and other 10,000 as the test dataset.

B More Sensitivity Evaluations

More Results about the Impact of the Quantity of the Adversary's Features. We also conduct an experiment about the impact of the quantity of the adversary's features on TinyImageNet [1] by splitting pixels in an image. The results are shown in Fig 1. Similar to the results on the Criteo dataset, the active attack steadily outperforms the passive label inference attack, and the attack performance is always within the upper bound decided by the quantity of the adversary's features.

Complexity of the Bottom Model. Our passive and active label inference attacks rely on the assumption that the adversary's bottom model learns a good representation of local data. Thus, if the adversary employs a simple bottom model, he/she may have less capability of conducting the label inference attack. To study the impact of the bottom model's complexity, we conduct experiments on three datasets by keeping the numbers of layers of the whole VFL model unchanged and assigning different number of layers to the bottom models. As shown in Figure 2, the more layers the adversary's bottom model has, the better the active label inference attack performs. This sheds insight for defense that the server should choose a complex top model and limit the expressiveness of bottom models to mitigate label leakage risks. Another insight is that the VFL models with simpler tasks face a greater risk of label leakage. For example, on CIFAR-10, when the adversary has a bottom model composed of only one convolutional layer, the top-1 inference accuracy is only around 0.18 on both the training and testing datasets. However, on Criteo, a dataset for an easier task, with the bottom model composed of only one fully connected layer, the adversary can achieve the top-1 inference accuracy of around 0.63 on both the training and testing datasets.

What if the benign participant uses the malicious optimizer. In our active attack, the adversary applies a malicious local optimizer to boost the ability to infer labels, which raises an interesting question – what if all participants, including the benign ones, accelerate their local learning? How will it influence the federated model's performance on the original task and the attacker's label inference performance? To investigate this, we conduct experiment on six datasets. The results are shown in Table 1. We have two conclusions derived from the results. One conclusion is that whether the benign participant uses the malicious optimizer has little impact on the VFL model's performance on the original task. In other words, if we regard applying the malicious optimizer at the benign participant side as a defense, the cost of this defense is trivial. On all the six datasets, the performance differences on all the metrics with respect to the original task, such as top-1 accuracy and F1 score, are within 1.56%. The other conclusion is that applying the malicious optimizer at the benign participant side cannot effectively defend against the active attack, i.e., suppress the attacker's label inference performance. The strongest suppression effect happens on CIFAR-100, where

Table 1: The malicious local optimizer's influence to the attack performance. "Active (All)" means that both the adversarial and benign participants use the malicious local optimizer, while "Active" means that only the adversary uses the malicious local optimizer.

Dataset	Metric	VFL Model's Performance on the Original Task		Attack Performance			
		Test Set		Train Set		Test Set	
		Active	Active (All)	Active	Active (All)	Active	Active (All)
Criteo	Top1 Acc	**0.7128**	0.7089	**0.6879**	0.6771	**0.6830**	0.6734
YahooAnswers	Top1 Acc	0.7120	**0.7133**	0.6424	**0.6467**	0.6419	**0.6467**
CINIC-10	Top1 Acc	0.7400	**0.7460**	**0.7818**	0.7643	0.5995	**0.6049**
CIFAR-10	Top1 Acc	**0.8139**	0.8108	**0.8484**	0.8252	**0.6342**	0.6251
CIFAR-100	Top5 Acc	**0.7500**	0.7440	**0.6732**	0.6077	**0.4700**	0.4289
BHI	F1 Score	**0.8504**	0.8367	**0.7824**	0.7378	**0.7673**	0.7211

the top-5 inference accuracy of the active attack decreases by 0.0655 on the training dataset. However, even after being suppressed, the adversary's top-5 inference accuracy can still remain 0.6077. On the other datasets, the suppression effect is more trivial. For example, on Criteo, the F1 score of the adversary's inference performance only decreases by 0.0108. On Yahoo Answers, the adversary's top-1 inference accuracy even increases by 0.0043, which indicates that applying the malicious optimizer at the benign participant side is useless for the defense on this dataset. To sum up, applying the malicious optimizer at the benign participant side has little impact on the VFL model's performance on the original task, and it only has a very limited suppression effect on the attacker's label inference performance.

C Further Exploiting the Inferred Labels

In real-world scenarios, correlated features are quite common, thus the leakage of one private feature may cause the leakage of another private feature. We conduct an experiment to show that the adversary can take advantage of the inferred labels to infer more private information on a real-world medical dataset. The illustration of this experiment is shown in Figure 3.

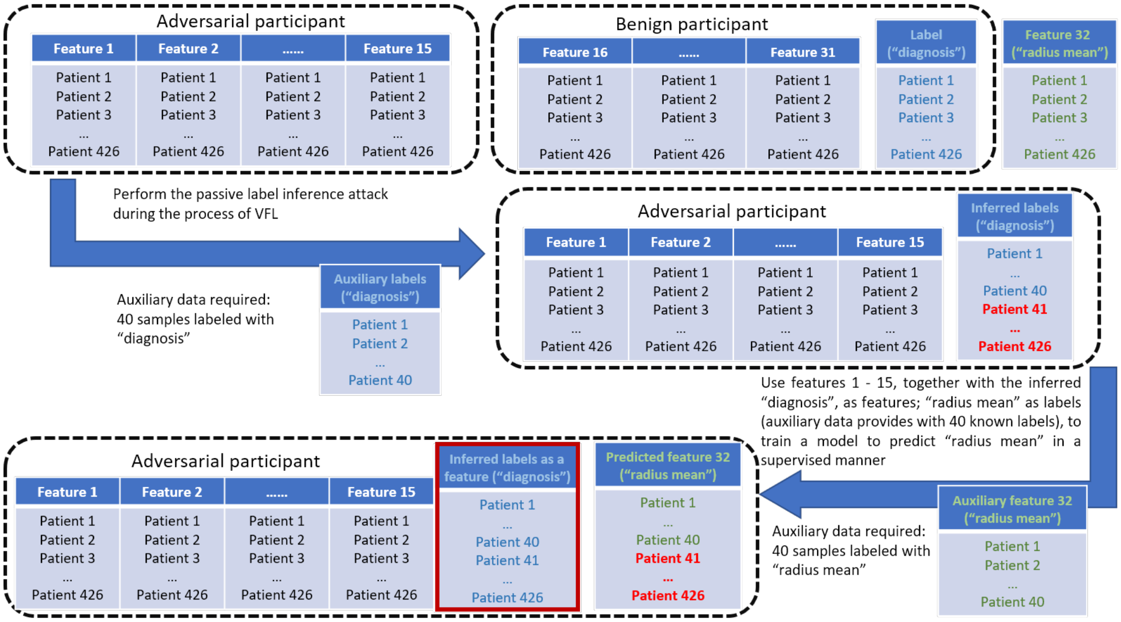
Adversarial participant
Feature 1
Feature 2
......
Feature 15
Patient 1
Patient 2
Patient 3
...
Patient 426
Benign participant
Feature 16
......
Feature 31
Label ("diagnosis")
Feature 32 ("radius mean")
Perform the passive label inference attack during the process of VFL
Auxiliary data required: 40 samples labeled with "diagnosis"
Auxiliary labels ("diagnosis")
Patient 40
Adversarial participant
Inferred labels ("diagnosis")
Patient 41
Use features 1 - 15, together with the inferred "diagnosis", as features; "radius mean" as labels (auxiliary data provides with 40 known labels), to train a model to predict "radius mean" in a supervised manner
Auxiliary data required: 40 samples labeled with "radius mean"
Auxiliary feature 32 ("radius mean")
Adversarial participant
Inferred labels as a feature ("diagnosis")
Predicted feature 32 ("radius mean")

Figure 3: Illustration of how the adversarial participant leverages the inferred labels to infer even more private information, e.g., private features of other participants. The red texts indicate that the samples are inferred by the adversarial participant.