A COMPARISON WITH QUERY-BASED ATTACK

Query-based black-box attack aims to attack the target model with limited queries to the target model. Through these queries, the source model manage to optimize the decision boundary towards the target model. However, most APIs and FL-based systems requires charges to access the service or equips anomaly detection that detects multiple or malicious queries. In this section, we provide a comparison of our proposed attack setting and the query-based black-box attack. To conduct a fair comparison, we follow the experiment setting in Section 4.2. With data from a limited number of users, instead of using the ground-truth class labels to train the source model, we query the target model for the prediction and set the predicted label as ground-truth. Note that, to facilitate the queries for the data from one client, one will have to perform 500 (in 100 clients partition) requests to the target model. The comparison is shown in Figure 5. We can see from Figure 5 that query-based outperforms the our proposed attack setting by a slight margin when the number of clients used to train the source model is small and the transfer rate of query-based attack continues to increase after 20 users where our attack setting begins to decrease. This can be attributed to the queries which help the source model to learn a more similar decision boundary. Through this experiment, we can hypothesize that, with some limited number of queries, our attack setting can be further boosted since it can help the substitute model to learn a similar loss landscape and decision boundary as the ones of target model.

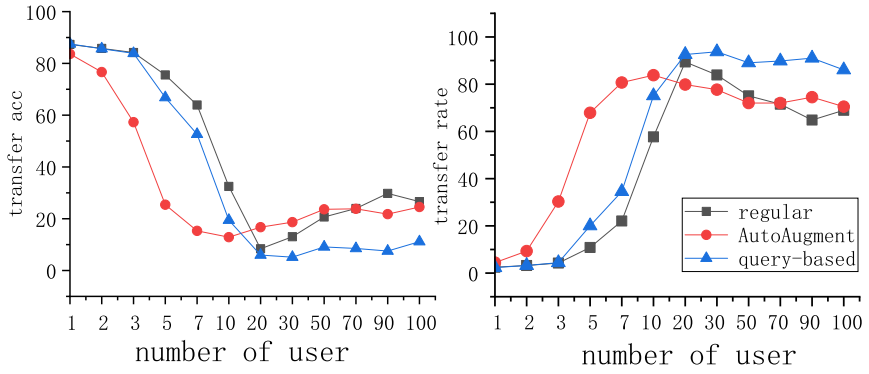


Figure 5: Comparison with query-based black-box attack; Left: transfer accuracy as a function of number of users' data leveraged in source model training; Right: transfer rate as a function of number of users' data leveraged in source model training;

B HETEROGENEITY IN PATHOLOGICAL NON-IID DISTRIBUTION

To prove generalization of our findings with other non-iid distribution, we follow McMahan et al. (2017) to generate pathological distribution (partition by shards). To partition CIFAR10 datatset into 100 clients, we first sort the samples by class label and then divide the data into n shards of size $50000/n$ (*e.g.* 500 shards of size 100) and assign each client $n/100$ shards. In this way, most client will have only samples of $n/100$ classes McMahan et al. (2017). By varying the number of shards n, we control the maximum number of classes in most clients and hence the degree of non-iid of partitions. We conduct experiments on two different settings, the 10 user partition and 100 user partition. For 10 user partition, we generate a partition comprised of 10 users and vary the number of shards n from 20 to 70. For 100 user partition, we generate a partition comprised of 10 users and vary the number of shards n from 200 to 700. We generate the transfer rate v.s. maximum number of classes plots as in the Section 5.1. We can observe from Figure 6 a increase trend in both experiment settings with ResNet50 and CNN which demonstrates that our findings hold under different ways of simulating data heterogeneity. We also perform hypothesis testing for this correlation in Appendix C. All four experiments show significant correlation under a level of 0.01.

Notice we also find that the transfer rate of 10 users experiment is consistently larger than the 100 users setting which further demonstrates our findings in Section 5.1 that more decentralized training leads to lower adversarial transferability for federated model.

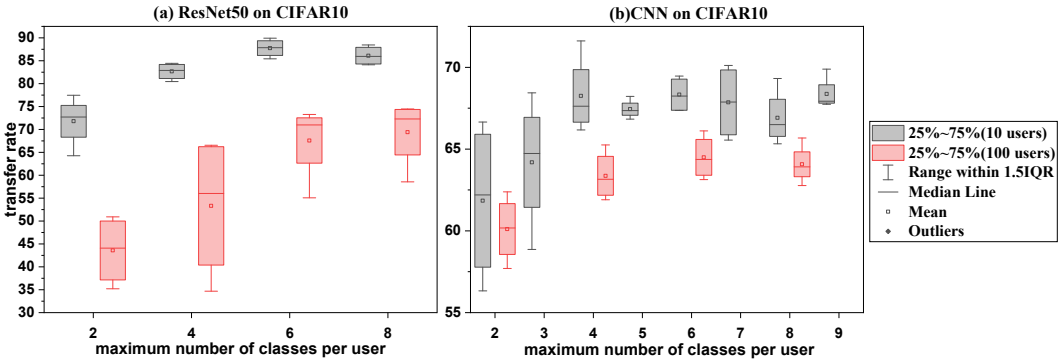


Figure 6: Transfer rate v.s. maximum number of classes per client; (a): Results of ResNet50 on CIFAR10 dataset; (b): Results of CNN on CIFAR10 dataset;

C Statistical Hypothesis Testing on Spearman correlation coefficient

We demonstrates the correlation between different degree of decentralization, heterogeneity of training data and transferability of adversarial examples in Section 5.1 through plots and graphs and show that with more decentralized, heterogeneous data, the federated model is more robust to transfer attack. We also display a negative relation between the number of clients to average and transfer success rate through box charts in Section 5.2.

To statistically validate these correlations, we perform two-tailed Hypothesis Testing on Spearman correlation coefficient. To conduct Hypothesis Testing for Spearman correlation coefficient on a specified correlation, we first calculate the Spearman correlation coefficient ρ on the two sets of points (*e.g.* T.Rate and Dirichlet α):

$$\rho = \frac{cov(X, Y)}{\sigma(X)\sigma(Y)}$$

where $cov(\cdot, \cdot)$ denotes the covariance and $\sigma(\cdot)$ represents the standard deviation. To perform the Hypothesis Test, we first have the Null Hypothesis H_0 and Alternate Hypothesis H_a:

$$H_0 : \rho = 0$$
$$H_a : \rho \neq 0$$

We choose the significance level to be 0.1, which means we reject H_0 is p-value is smaller than 0.1. We report the p-value and Spearman correlation coefficient in Table 3.

Table 3: Spearman correlation coefficient and p-value

	X	p-value (two-tailed)	Spearman coefficient
ResNet50	dirichlet α	.006	.59
	unbalance sgm	.05	-.44
	number of user in partition	.003	-.63
	maximum number of classes (10 users)	$< .001$	0.80
	maximum number of classes (100 users)	$< .001$	.76
	number of user to average (30 users)	$< .001$	-.71
	number of user to average (centralized)	$< .001$	-.79
CNN	dirichlet α	.76	.074
	unbalance sgm	.84	.049
	number of user in partition	$< .001$	-.83
	maximum number of classes (10 users)	.009	.45
	maximum number of classes (100 users)	.005	.67
	number of user to average (30 users)	$< .001$	-.77
	number of user to average (centralized)	$< .001$	-.83

We can see, as reported in Section 5.1 and 5.2, all experiments except the CNN experiments on dirichlet α and unbalance sgm can deomenstrates significant correlation under a significance level of 0.1. More to the point, we report numerous correlations with p-value less than .001 (significant under level of .001). This Hypothesis Testing validated the findings of our investigation.

D DOES DIFFERENT PARTITION MATTERS IN TRANSFER ATTACK?

In Section 4.2, we discover that with source model trained in federated manner, T.Rate can be boosted to the highest of 99%. Since we assume that the attacker have information regarding the data partition of all malicious clients, we leverage the same partition used in target model to train the source model. In this section, we are curious about whether different partitions used by source model effects the transferability of its adversarial examples against the target model. That is, whether using a partition different from the target model to train the source model affects the transfer rate of its adversarial examples. To explore such setting, we first randomly generates two different partitions with distinct random seeds and then perform the source model training and transfer attack the same as in Section 4.2. We repeat the experiment 4 times with different random seeds and report the mean results in Figure 7. We observe no significant difference between the same partition and the different partition setting. To further validate this observation, we perform Hypothesis Test on the obtained results with Paired Sample T-Test. The p-value .393 means that there is no significant difference between the transfer rate obtained by same and different partition. This investigation and result further improve the possibility of attacking a FL system through black-box attack.

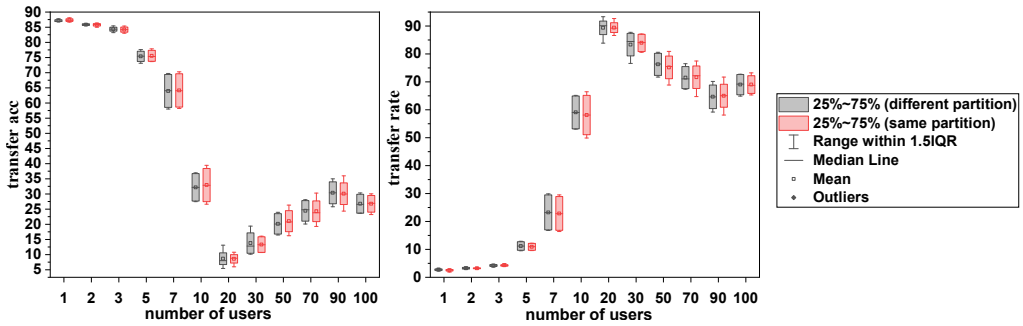
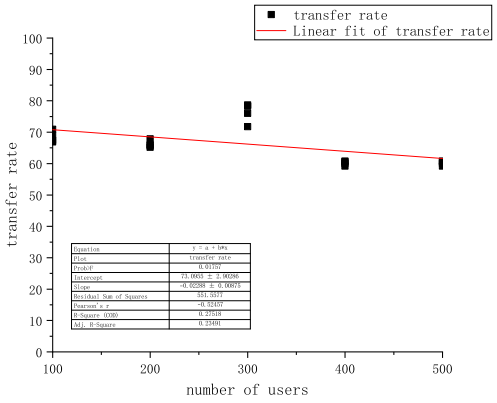


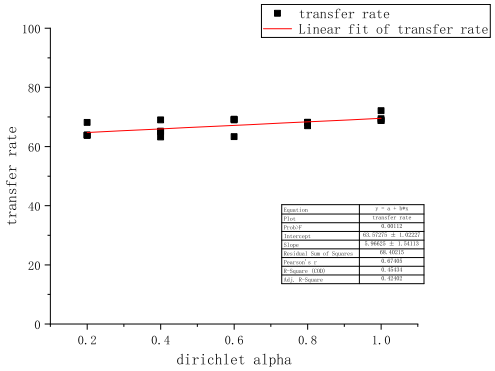
Figure 7: Difference between same partition and different partition; Left: transfer accuracy v.s. number of users' data leveraged in source model training; Right: transfer rate v.s. number of users' data leveraged in source model training;

E LINEAR REGRESSION TO VISUALIZE THE CORRELATION

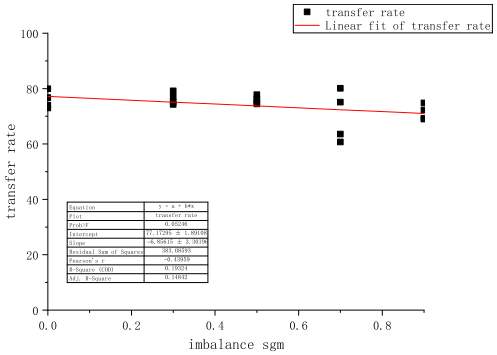
To better demonstrates the correlation between various factors and adversarial transferability, we perform linear regression with hypothesis testing on the experiment results. We plot scatter graph and linear regression line on each of the correlation and corresponding experiment result as shown in the following figures:



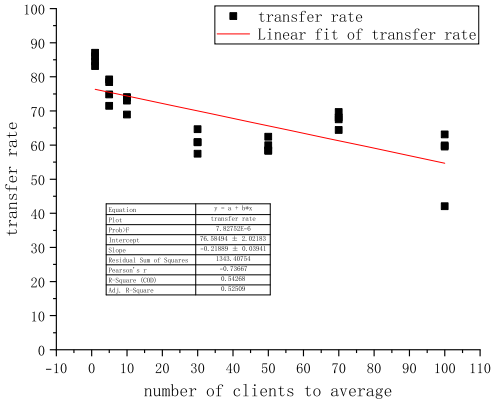
(a) ResNet50: transfer rate v.s. number of total users in the partition.



(b) ResNet50: transfer rate v.s. dirichlet alpha.

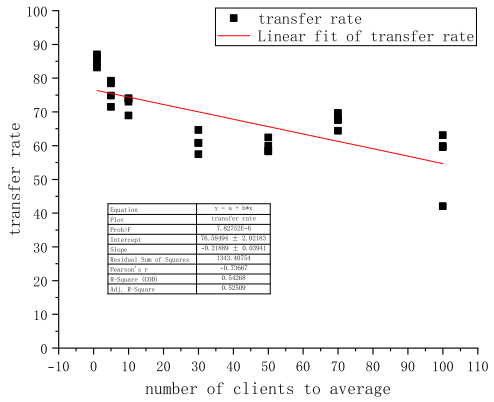


(c) ResNet50: transfer rate v.s. unbalance sgm.

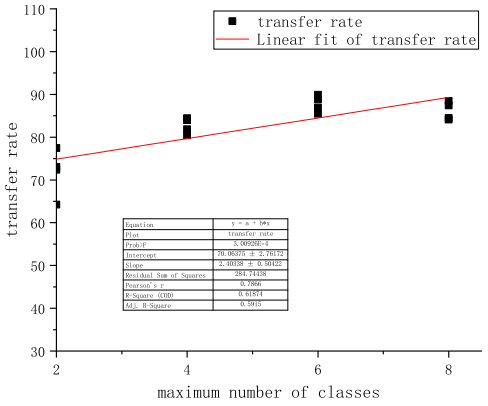


(d) ResNet50: transfer rate v.s. number of users to average per round (source model trained in centralized manner with full training dataset).

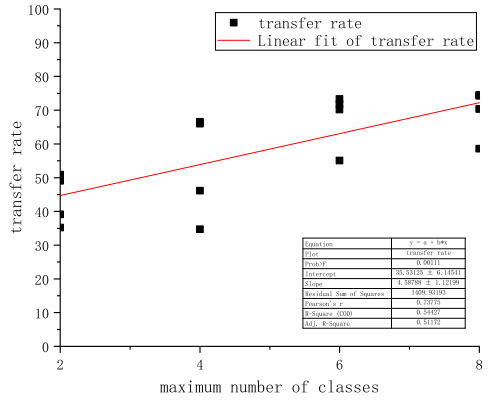
Figure 8: Linear regression visualization



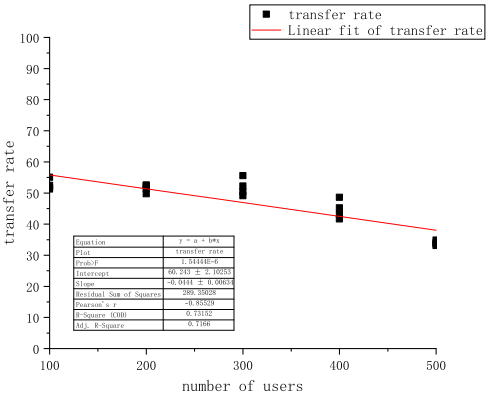
(a) ResNet50: transfer rate v.s. number of users to average per round (source model trained in centralized manner with 30 client's data).



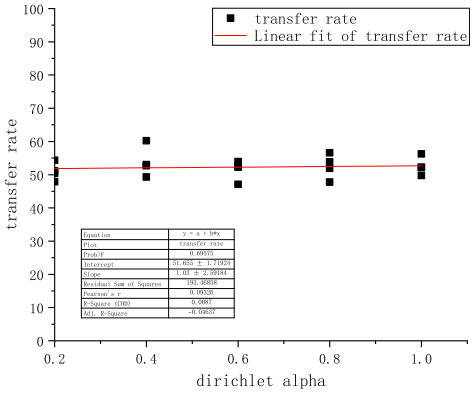
(b) ResNet50: transfer rate v.s. maximum number of classes per user (10 users)..



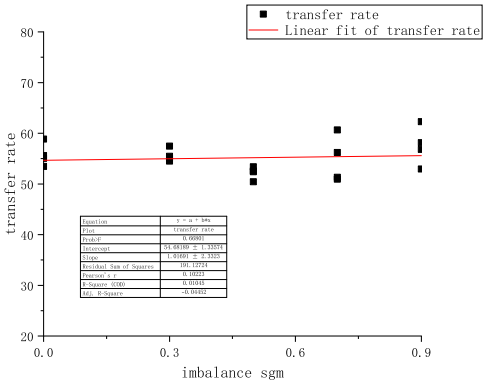
(c) ResNet50: transfer rate v.s. maximum number of classes per user (100 users).



(d) CNN: transfer rate v.s. number of total users in the partition.

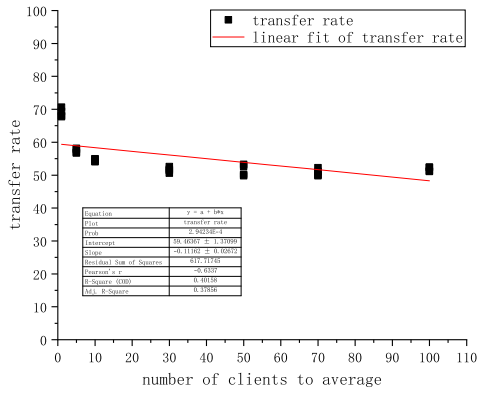


(e) CNN: transfer rate v.s. dirichlet alpha.

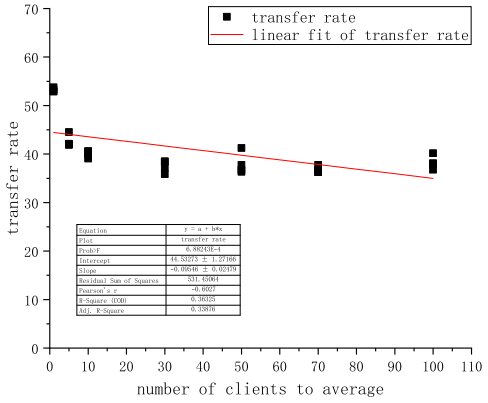


(f) CNN: transfer rate v.s. unbalance sgm.

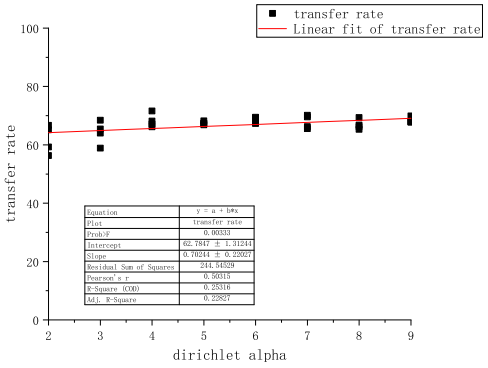
Figure 9: Linear regression visualization



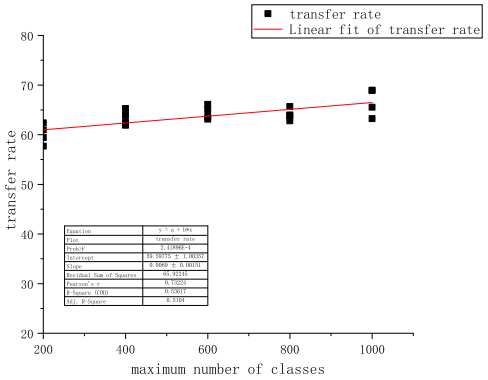
(a) CNN: transfer rate v.s. number of users to average per round (source model trained in centralized manner with full training dataset).



(b) CNN: transfer rate v.s. number of users to average per round (source model trained in centralized manner with 30 client's data).



(c) CNN: transfer rate v.s. maximum number of classes per user (10 users)..



(d) CNN: transfer rate v.s. maximum number of classes per user (100 users).

Figure 10: Linear regression visualization