8 Supplementary Material

8.1 Data Augmentation

Models trained by FixMatch, FlexMatch, and UDA apply both weak and strong augmentation to the unlabeled samples. For the weak augmentation, we apply random cropping with a padding of 4 and random horizontal flipping to each sample following [15,38]. For the strong augmentation, we apply random augmentation [7] to each training sample, which consists of a group of augmentation operations. Specifically, we set $N = 2$ and $M = 10$ where N is the number of transformations to a given sample and M is the magnitude of global distortion.

8.2 Attack Performance with Different Numbers of Labeled Samples

Figure 8, Figure 9, and Figure 10 show the attack performance with 1,000, 2,000, and 4,000 labeled training data. We observe that our proposed data augmentation-based attack $\mathcal{A}_{DA}$ still consistently outperforms baseline attacks.

8.3 What Determines Membership Inference Attack in SSL with Different Numbers of Labeled Samples.

Figure 11, Figure 12, Figure 13, and Figure 14 shows the results of models trained by different SSL methods on the three datasets with 500, 1,000, 2,000, and 4,000 labeled samples. We has the similar finding as Section 5.4, i.e., the models trained by SSL has almost no overfitting, but the JS Distance (Entropy) and the attack performance do increase during the training.

8.4 Ablation Study: Number of Views

For the SSL methods trained on different datasets with 1,000, 2,000, and 4,000 labeled samples, we range the number of views from 1 to 100 and the attack performance is shown in Figure 15, Figure 16, and Figure 17, respectively. We have the similar observation as Section 5.5 that more number of views leads to better performance.

8.5 Ablation Study: Similarity Function

The results for FixMatch, FlexMatch, and UDA trained on three different datasets with 1,000, 2,000, and 4,000 labeled samples are shown in Figure 18, Figure 19, and Figure 20, respectively. We observe that the JS Distance still consistently outperforms the other three distance metrics and achieves the best performance.

8.6 Ablation Study: Data Augmentation and Shadow Model Architecture

Data Augmentation. In the previous evaluation of $\mathcal{A}_{DA}$, we assume the adversary knows the data augmentation used to train the target model and can apply the same data augmentation to conduct the attack. We then relax this assumption to see whether $\mathcal{A}_{DA}$ is still effective with different levels of data augmentations. We take FixMatch trained on CIFAR10 with 500 labeled samples as a case study amd the results are shown in Table 3. We find that $\mathcal{A}_{DA}$ is still effective even with the weakest augmentation (Aug-Level is 0) and the attack performance can be improved with stronger augmentations added. For instance, the attack AUC is 0.876 and 0.882 when the Aug-Level is 0 and 4, respectively. This implies that a successful attack can still be launched even without the exact knowledge of the data augmentation used to train the target model.

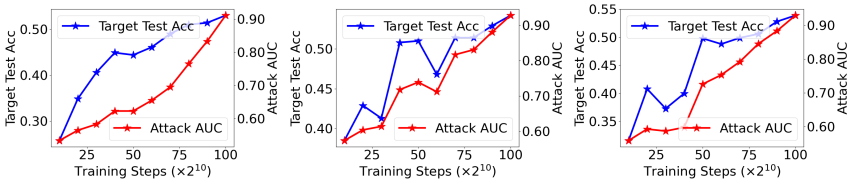
Table 3: The attack performance with respect to different augmentation levels. For Aug-Level=0, we only apply random cropping with flipping as both weak and strong augmentation. For 1-4, we gradually apply 1-4 transformation methods for the strong augmentation to each image from the general augmentation pools. The target model is trained by FixMatch on CIFAR100 with 500 labeled samples. ($\star$) denotes our default setting in the paper.

Aug-Level	0	1	2 ($\star$)	3	4
Attack AUC	0.876	0.880	0.880	0.881	0.882

Shadow Model Architecture. We then relax the assumption where the shadow model has the same architecture as the target model. Given the target model (WRN28-2 trained by FixMatch on CIFAR100 with 500 labeled samples), we train shadow models with WRN28-2, WRN28-1, WRN28-4, WRN28-8, and ResNet50 as the architectures, and the corresponding attack AUCs are 0.880, 0.875, 0.839, 0.835, and 0.859, respectively. Our attack achieves the highest attack AUC when the shadow and target models use the same architecture. However, our attack is still effective even if the shadow model has a different model architecture (e.g., the attack AUC is 0.859 when the shadow model architecture is ResNet50).

8.7 Defense Evaluation

Besides early stopping (ES), we evaluate three more defenses, i.e., top-k posteriors (top-k) [28], model stacking (MS) [27], and DP-SGD [2]. We evaluate both the target model's utility (test acc) and the effectiveness of defenses (attack auc). The results are shown in Table 4. We observe that DP-SGD is the most



(a) FixMatch (b) FlexMatch (c) UDA

Fig. 7: The target model performance and attack AUC with respect to different training steps. The target model is trained on CIFAR100 with 4,000 labeled samples, which has the highest performance on its original classification task. The attack model is $\mathcal{A}_{DA}$, which has the best attack performance.

effective defense since it achieves the lowest attack AUC with 0.598. However, DP-SGD suffers from unacceptable utility drop (with only 0.034 test acc). Existing work [13,18] on DP also show that DP-SGD sacrifices the model's utility substantially in order to achieve good privacy guarantee. Therefore, we consider early stopping as the best defense since it achieves the best privacy-utility trade-off. i.e., it reduces the membership leakage to a large extent (from 0.918 to 0.695) while maintaining the utility (from 0.530 to 0.490).

Table 4: Target model accuracy and attack AUC for different defenses. The target model is WRN28-2 trained by FixMatch on CIFAR100 with 4,000 labeled samples (same setting as the paper). For early stopping (ES), we stop at 70 $\times 2^{10}$ training steps. For top-k, we set k=1 as it leaks the least information. For model stacking (MS), we train four models, i.e., WRN28-{1,2,4,8}, using the same dataset and average their posteriors. For DP-SGD, the noise scale is set to 10^{-5} and the gradient norm is set to 1. Note that we use the Opacus library [1] to implement DP-SGD.

Defense	None	ES	Top-k	MS	DP-SGD
Test ACC	0.530	0.490	0.530	0.549	0.034
Attack AUC	0.918	0.695	0.906	0.905	0.598

Algorithm 1: Our Data Augmentation Based Attack $\mathcal{A}_{DA}$

Input: Given sample x, target model $\mathcal{T}$, NN-based attack model $\mathcal{A}_{DA}$, weak data augmentations $\mathcal{A}ug_{weak}$, strong data augmentations $\mathcal{A}ug_{strong}$, similarity function Sim, number of augmented views K

Output: Member or non-member

`/* Generate augmented views. */`

$\{x^1_{weak}, x^2_{weak}, \cdots, x^K_{weak}\} \leftarrow \mathcal{A}ug_{weak}(x, K)$

$\{x^1_{strong}, x^2_{strong}, \cdots, x^K_{strong}\} \leftarrow \mathcal{A}ug_{strong}(x, K)$

`/* Query the target model and obtain posteriors. */`

$\{p^1_{weak}, p^2_{weak}, \cdots, p^K_{weak}\} \leftarrow \{\mathcal{T}(x^1_{weak}), \mathcal{T}(x^2_{weak}), \cdots, \mathcal{T}x^K_{weak}\}$

$\{p^1_{strong}, p^2_{strong}, \cdots, p^K_{strong}\} \leftarrow \{\mathcal{T}(x^1_{strong}), \mathcal{T}(x^2_{strong}), \cdots, \mathcal{T}x^K_{strong}\}$

`/* Obtain similarity vectors. */`

Similarity vector $v_w(x) \leftarrow \text{SORTED}(\{Sim(p^i_{weak}, p^j_{weak}) | i \in [1, K], j \in [1, K]\})$

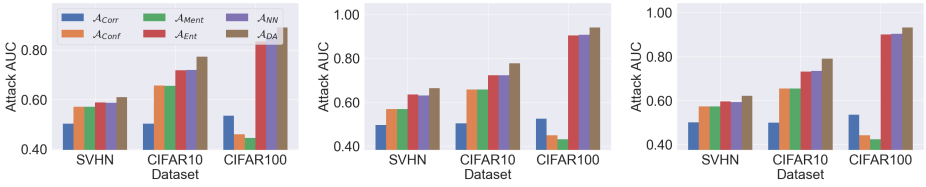
Similarity vector $v_s(x) \leftarrow \text{SORTED}(\{Sim(p^i_{strong}, p^j_{strong}) | i \in [1, K], j \in [1, K]\})$

Similarity vector $v_{ws}(x) \leftarrow \text{SORTED}(\{Sim(p^i_{weak}, p^j_{strong}) | i \in [1, K], j \in [1, K]\})$

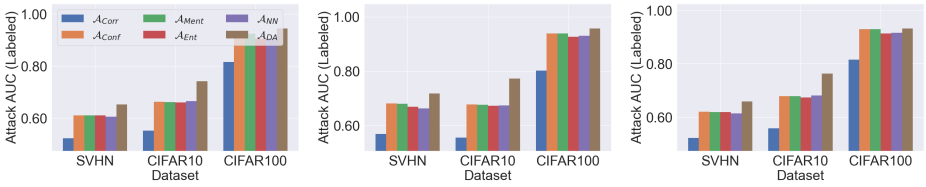
`/* Concatenate similarity vectors and perform the attack. */`

Merged vector $v(x) \leftarrow \text{CONCATENATE}(v_w(x), v_s(x), v_{ws}(x))$

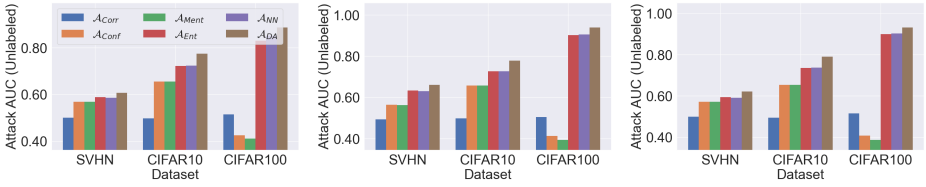
return $\mathcal{A}_{DA}(v(x))$



(a) Attack AUC

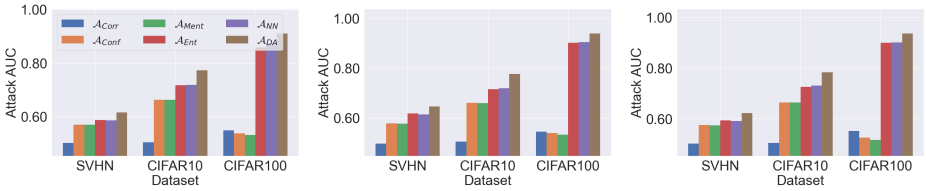


(b) Attack AUC (Labeled)

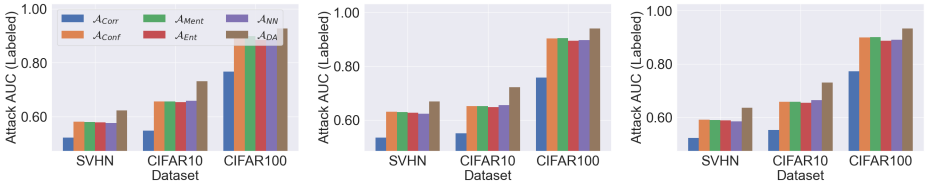


(c) Attack AUC (Unlabeled)

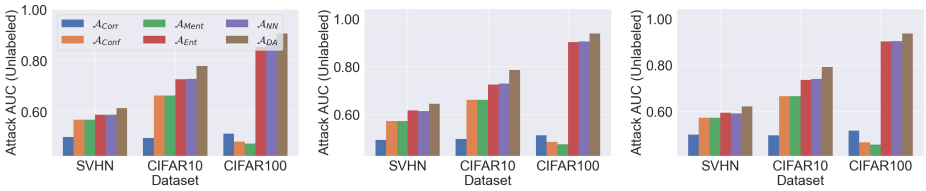
Fig. 8: The AUC of membership inference attacks against models trained by different SSL methods with 1,000 label samples. The first to third columns denotes the model trained by FixMatch, FlexMatch, and UDA, respectively.



(a) Attack AUC

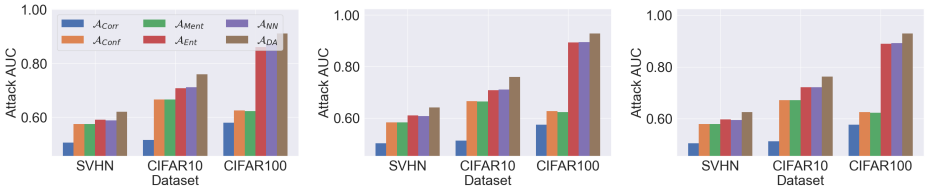


(b) Attack AUC (Labeled)

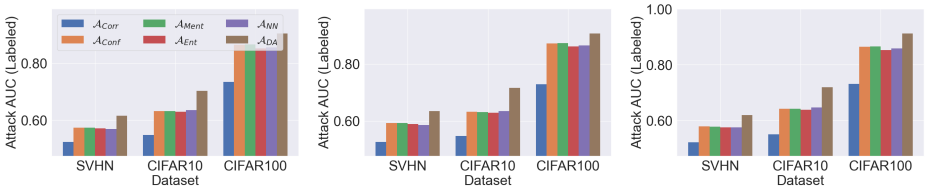


(c) Attack AUC (Unlabeled)

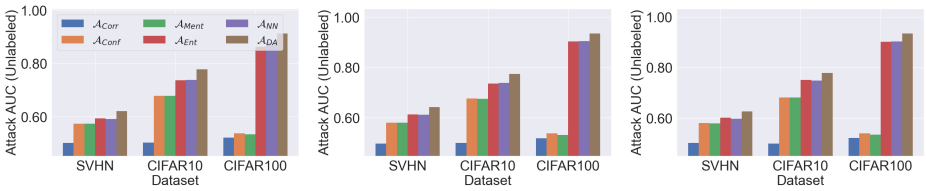
Fig. 9: The AUC of membership inference attacks against models trained by different SSL methods with 2,000 label samples. The first to third columns denotes the model trained by FixMatch, FlexMatch, and UDA, respectively.



(a) Attack AUC

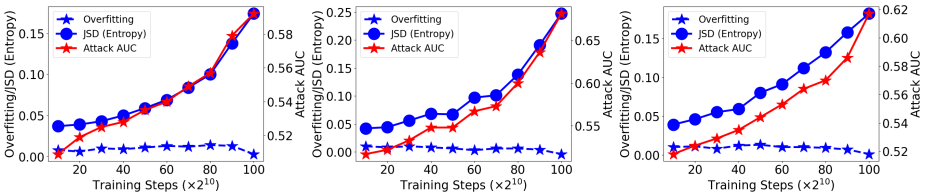


(b) Attack AUC (Labeled)

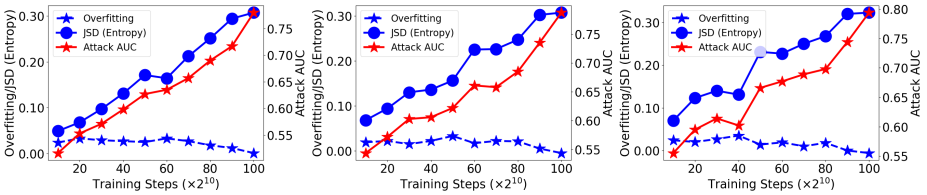


(c) Attack AUC (Unlabeled)

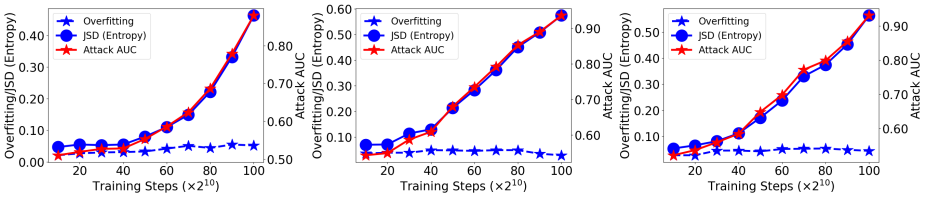
Fig. 10: The AUC of membership inference attacks against models trained by different SSL methods with 4,000 label samples. The first to third columns denotes the model trained by FixMatch, FlexMatch, and UDA, respectively.



(a) SVHN

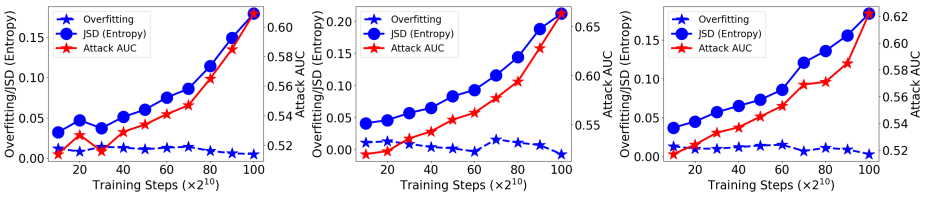


(b) CIFAR10

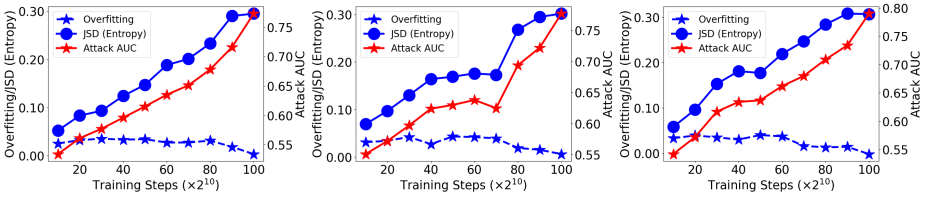


(c) CIFAR100

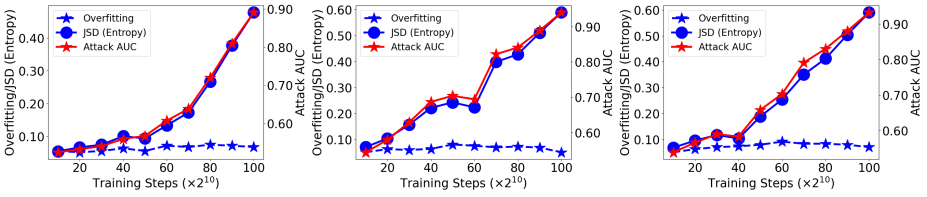
Fig. 11: The overfitting/JS Distance (Entropy) and attack AUC with respect to different training steps. The first to third columns denotes the model trained by FixMatch, FlexMatch, and UDA with 500 labeled samples, respectively. Note that we consider the attack AUC of $\mathcal{A}_{DA}$, which is the strongest attack.



(a) SVHN

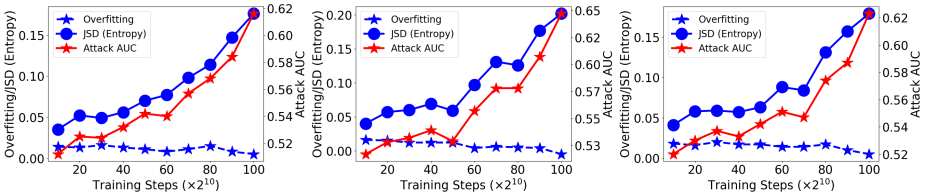


(b) CIFAR10

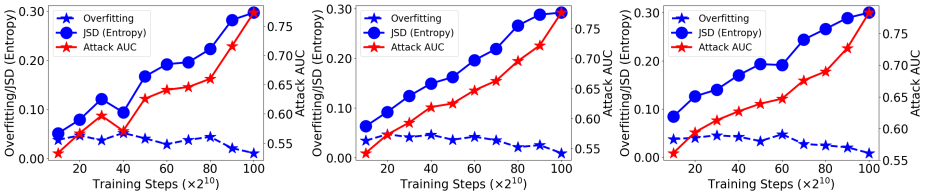


(c) CIFAR100

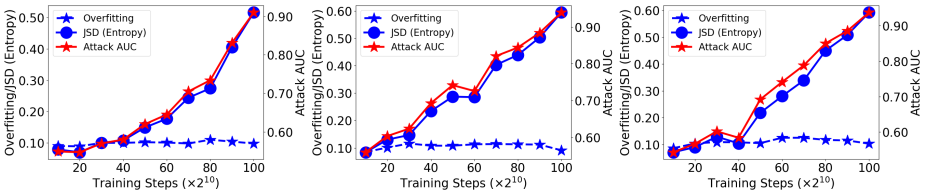
Fig. 12: The overfitting/JS Distance (Entropy) and attack AUC with respect to different training steps. The first to third columns denotes the model trained by FixMatch, FlexMatch, and UDA with 1,000 labeled samples, respectively. Note that we consider the attack AUC of $\mathcal{A}_{DA}$, which is the strongest attack.



(a) SVHN

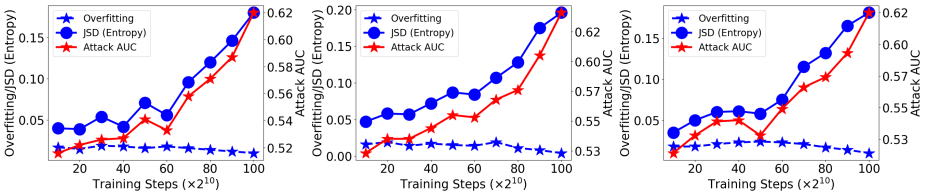


(b) CIFAR10

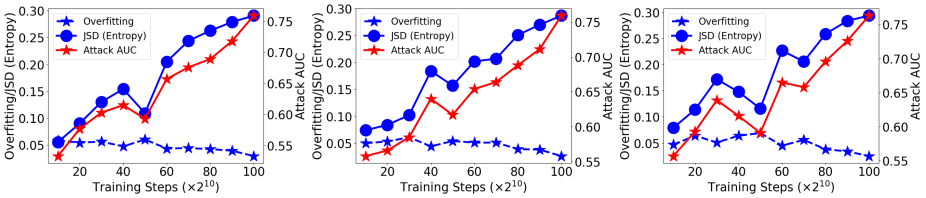


(c) CIFAR100

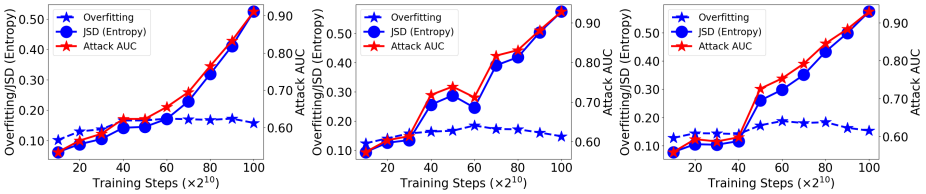
Fig. 13: The overfitting/JS Distance (Entropy) and attack AUC with respect to different training steps. The first to third columns denotes the model trained by FixMatch, FlexMatch, and UDA with 2,000 labeled samples, respectively. Note that we consider the attack AUC of $\mathcal{A}_{DA}$, which is the strongest attack.



(a) SVHN

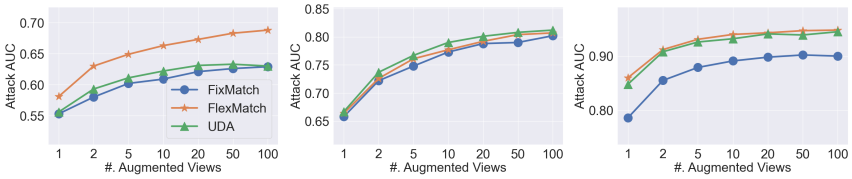


(b) CIFAR10



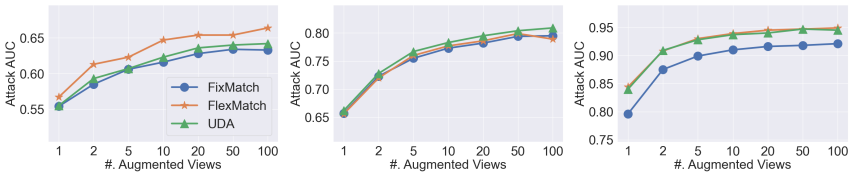
(c) CIFAR100

Fig. 14: The overfitting/JS Distance (Entropy) and attack AUC with respect to different training steps. The first to third columns denote the model trained by FixMatch, FlexMatch, and UDA with 4,000 labeled samples, respectively. Note that we consider the attack AUC of $\mathcal{A}_{DA}$, which is the strongest attack.



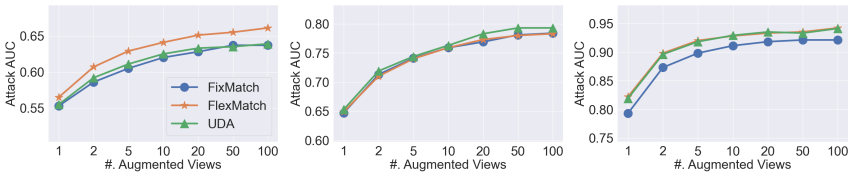
(a) SVHN (b) CIFAR10 (c) CIFAR100

Fig. 15: The attack AUC of $\mathcal{A}_{DA}$ with different numbers of augmented views to query the target model. The target model is trained with 1,000 labeled samples.



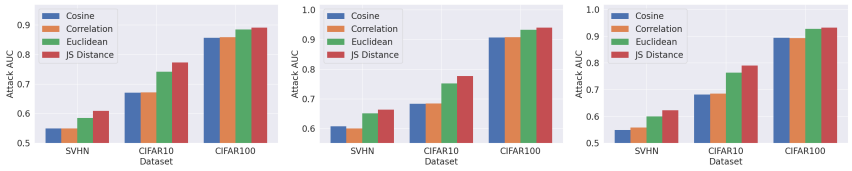
(a) SVHN (b) CIFAR10 (c) CIFAR100

Fig. 16: The attack AUC of $\mathcal{A}_{DA}$ with different numbers of augmented views to query the target model. The target model is trained with 2,000 labeled samples.



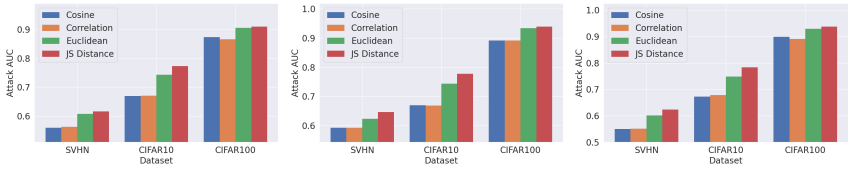
(a) SVHN (b) CIFAR10 (c) CIFAR100

Fig. 17: The attack AUC of $\mathcal{A}_{DA}$ with different numbers of augmented views to query the target model. The target model is trained with 4,000 labeled samples.



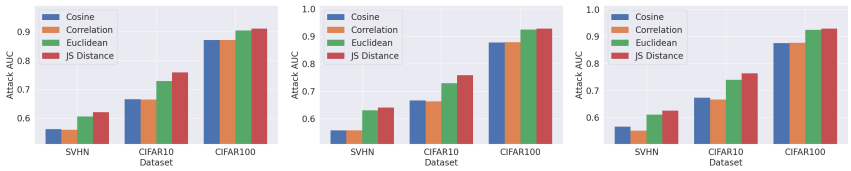
(a) FixMatch (b) FlexMatch (c) UDA

Fig. 18: The attack AUC of $\mathcal{A}_{DA}$ with different similarity functions. The target model is trained with 1,000 labeled samples.



(a) FixMatch (b) FlexMatch (c) UDA

Fig. 19: The attack AUC of $\mathcal{A}_{DA}$ with different similarity functions. The target model is trained with 2,000 labeled samples.



(a) FixMatch (b) FlexMatch (c) UDA

Fig. 20: The attack AUC of $\mathcal{A}_{DA}$ with different similarity functions. The target model is trained with 4,000 labeled samples.