Appendices

A Experiment environment

Experiments are implemented with Python 3.7.5/Pytorch 1.6.0/CUDA 10.1 and run on two servers with (1) Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz CPU and NVIDIA Quadro RTX 8000 GPU and (2) Intel(R) Xeon(R) Silver 4116 CPU @ 2.10GHz CPU and NVIDIA TITAN RTX GPU, respectively. Codes will be open-sourced once the paper is published.

B Comparison with MIAs on data augmented models

Yu et al. (2021) consider the scenario of data augmentation in model training. Specifically, the attacker has the original data, but only some augmented (transformed) versions are in the model training. On the other hand, we consider the scenario where the data that the attacker collects are transformed.

Suppose x is the original data, and $x' = g(x)$ is transformed data with transformation g. If g is invertible, and $x = h(x')$ where h is the mathematical inverse of g, we can switch the role of x and x', that is, x' is now regarded as the original example and x is the transformed version of x' via transformation h. In such cases, our scenario is equivalent to the scenario where the model is trained with single-time data augmentation. However, the assumption is not applicable for most transformations as they lose information, especially for those strictly lossy transformations (e.g., JPEG, down-scaling, feature-missing). In addition, for data with bounded values such as images, even invertible transformations can result in information loss due to value clipping. As a result, it is hard to apply Yu et al.'s attack in our scenario since they must know the exact form of the (inverse) transformation h.

On the contrary, both of our attacks can easily apply to any transformation. More importantly, the reverse transformation attack applies to all situations where the attacker gets slightly different examples from those in model training. In other words, the reverse transformation attack can also apply to data augmentation scenarios or even the combination of both scenarios discussed above. We will leave this topic in our future work.

C Details of current MIAs

Here we provide detailed information for the five current attacks we evaluated in Section 4. Four of them are metric-based attacks, and the other one is an NN-based attack. Suppose f is the target model, x is the target example, τ is a real number represents the threshold, then the membership inference functions $\mathcal{M}$ for the five attacks are:

Classification correctness (CC) Classification correctness attack (Yeom et al., 2018) simply treats all correctly classified examples as training members.

$$\mathcal{M}_{\mathrm{CC}}(x, f) = \mathbf{1}(\mathrm{argmax} f(x) = y)$$

Loss thresholding (LT) Loss thresholding attack (Yeom et al., 2018) treats all examples with losses lower than a preset threshold as training members.

$$\mathcal{M}_{\mathrm{LT}}(x, f) = \mathbf{1}(\mathcal{L}(f(x), y) < \tau_{\mathrm{LT}})$$

where $\mathcal{L}$ is the loss function of the target model.

Confidence score thresholding (ST) Confidence score thresholding attack (Salem et al., 2018) treats all examples whose confidence score of the true label (the original attack selects the maximum confidence score) is larger than a preset threshold as training members.

$$\mathcal{M}_{\mathrm{ST}}(x, f) = \mathbf{1}(f(x)_y > \tau_{\mathrm{ST}})$$

Entropy thresholding (ET) Entropy thresholding attack (Song & Mittal, 2021) treats all examples whose output's entropy is smaller than a preset threshold as training members.

$$\mathcal{M}_{\mathrm{ET}}(x, f) = \mathbf{1}(\mathcal{E}(f(x), y) < \tau_{\mathrm{ET}})$$

where $\mathcal{E}(f, y) = -(1 - f_y) \cdot \log(f_y) - \sum_{i \neq y} f_i \cdot \log(1 - f_i)$ is the entropy metric.

NN-based attack (NN) NN-based attack (Shokri et al., 2017) collects an attack dataset containing output logits from shadow models and then trains a binary classifier for membership inference.

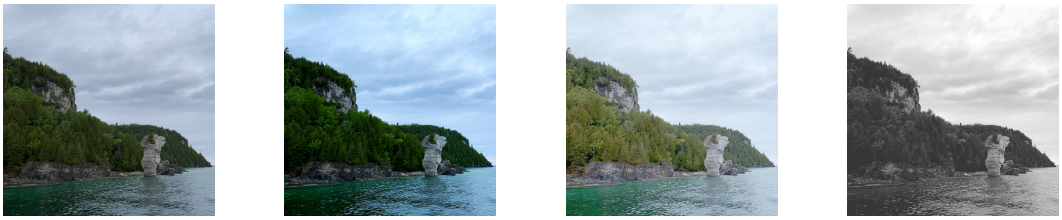
$$\mathcal{M}_{\mathrm{NN}}(x, f) = \mathrm{argmax}\mathcal{A}(f(x))$$

where $\mathcal{A}$ is the binary classifier.

D Details of image filters

The three filters we applied in our experiments are Clarendon, Gingham, and Moon. They are the first three image filters implemented inside Instagram, a most popular photo-sharing platform with one billion active users (version 207.0 by the time of the current submission). The effects of the three filters are shown in Figure 3. The three filters are pretty different and represent various styles of image filtering. For example, Clarendon increases image saturation while Gingham reduces it; Clarendon and Gingham keep color while Moon goes grayscale. For experiment, we apply the implementation from *pilgram* library (Kamakura, 2019).

Our experiments assume that the attacker applies the filters as a black box, which means they do not need to know the detailed filter algorithm. Theoretically, for filters that only contain linear transformations, such as only increasing brightness, attackers can obtain the original image by applying the inverses of those linear transformations. However, in reality, image pixel values are strictly bounded. Even if many filters only contain linear transformations, most filtered images lose information during value clipping and cannot be reverted. On the other hand, we can apply our attacks to any black-box filters with a unified reverse transformation function.



(a) Original (b) Clarendon (c) Gingham (d) Moon

Figure 3: Image filters

E Additional experimental results

E.1 Full evaluation results on CIFAR-10

Table 5 and Table 6 are the full tables of Table 1 and Table 2, respectively.

Table 5: Accuracies of current attacks on CIFAR-10

	Original	Gaussian(σ)			Adversarial			JPEG(q)			Scaling(r)			Filtering			Rotation(δ)		
		[0,0.1]	[0,0.2]	[0,0.3]	1-step	10-step	loss=10	50	10	1	0.5	1.5	10	Clarendon	Gingham	Moon	$[0°, 10°]$	$[0°, 20°]$	$[0°, 30°]$
CC	65.64%	66.21%	65.53%	63.48%	78.89%	50.22%	50.22%	66.68%	66.07%	56.15%	58.65%	68.02%	67.97%	66.72%	63.60%	64.41%	66.10%	63.11%	60.09%
LT	80.47%	77.32%	69.66%	64.78%	62.63%	50.02%	50.02%	74.13%	61.13%	51.98%	53.46%	69.54%	70.32%	72.78%	59.13%	60.56%	69.51%	62.78%	59.03%
ST	78.32%	75.84%	69.19%	64.57%	64.86%	50.10%	50.10%	73.40%	61.42%	52.06%	53.64%	69.61%	70.38%	72.04%	59.29%	60.69%	68.99%	62.63%	58.97%
ET	77.58%	76.44%	70.33%	65.50%	68.49%	50.03%	50.03%	75.59%	64.41%	52.93%	54.94%	72.30%	73.11%	73.81%	61.50%	63.20%	71.32%	64.67%	60.51%
NN	79.46%	76.33%	68.87%	63.14%	54.23%	48.71%	53.25%	73.10%	59.48%	51.37%	51.90%	67.56%	68.33%	72.18%	57.56%	59.75%	67.94%	61.63%	58.14%

Table 6: Accuracies and coverage difference rates ω of our attacks on CIFAR-10

(a) Gaussian noise, σ– range of standard deviation

σ		$[0, 0.1]$			$[0, 0.2]$			$[0, 0.3]$	
LTT		77.34%			71.13%			65.68%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	71.02%	75.04%	77.35%	68.92%	70.00%	70.93%	65.58%	64.14%	65.91%
ω	8.37%	6.96%	1.35%	9.27%	4.14%	3.81%	7.63%	2.52%	8.51%

(b) Adversarial noise

PGD		1-step			10-step			loss=10	
LTT		78.87%			64.32%			56.17%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	79.20%	79.19%	77.92%	60.26%	55.37%	52.81%	64.45%	61.30%	55.57%
ω	2.49%	5.62%	10.33%	38.28%	42.76%	44.85%	42.47%	42.90%	47.45%

(c) JPEG compression, q–compression quality

q		50			10			1	
LTT		75.43%			66.37%			57.00%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	71.10%	74.68%	75.31%	66.57%	67.08%	66.81%	57.31%	57.51%	56.95%
ω	9.02%	5.62%	2.17%	1.66%	7.47%	12.46%	10.97%	18.88%	23.11%

(d) Scaling, r–scaling factor

r		0.5			1.5			10.0	
LTT		58.87%			73.27%			73.78%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	61.09%	61.46%	61.09%	71.81%	73.40%	73.84%	71.98%	73.88%	74.28%
ω	13.53%	18.68%	23.41%	6.93%	2.06%	6.60%	7.11%	1.23%	6.00%

(e) Filtering

Filter		Clarendon			Gingham			Moon	
LTT		74.49%			64.05%			65.49%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	71.00%	74.03%	74.15%	64.11%	64.00%	63.47%	65.51%	65.56%	64.85%
ω	7.59%	3.70%	4.47%	1.66%	5.53%	10.72%	0.80%	5.21%	10.58%

(f) Rotation, δ–range of rotation degree

δ		$[0°, 10°]$			$[0°, 20°]$			$[0°, 30°]$	
LTT		70.73%			64.79%			60.93%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	69.13%	72.26%	77.32%	64.58%	68.64%	76.38%	61.52%	66.44%	75.57%
ω	8.47%	9.84%	14.2%	6.26%	13.73%	29.68%	13.69%	18.72%	39.47%

E.2 Evaluation on SVHN

SVHN (Netzer et al., 2011) is an image dataset for street house numbers. We use the cropped version of the original dataset that contains around 70k of 32×32 images of digits from 0 to 9.

For training SVHN models, we apply the same architecture and training settings as CIFAR-10 in Section 4. The final model of SVHN has a training accuracy of 99% and a testing accuracy of 89%. For MIAs, we also evaluated the six image transformations.

Table 7 and Table 8 show the results of current attacks and our attacks on SVHN with transformations. Similar to the results of CIFAR-10 and Purchase-100, we can see that our attacks can achieve better

accuracies with a properly selected ϵ-robust area, and our two attacks discover different sets of transformed training examples.

Table 7: Accuracies of current attacks on SVHN

	Original	Gaussian(σ)			Adversarial			JPEG(q)			Scaling(r)			Filtering			Rotation(δ)		
		[0,0.1]	[0,0.2]	[0,0.3]	1-step	10-step	loss=10	50	10	1	0.5	1.5	10	Clarendon	Gingham	Moon	$[0°, 10°]$	$[0°, 20°]$	$[0°, 30°]$
CC	55.65%	55.82%	55.99%	55.68%	63.90%	50.29%	50.29%	55.71%	56.14%	50.99%	56.40%	55.75%	55.72%	56.08%	55.58%	55.72%	55.84%	54.43%	52.55%
LT	65.21%	63.24%	60.02%	58.38%	52.96%	49.99%	49.99%	63.39%	56.27%	50.47%	59.39%	64.11%	64.23%	61.77%	56.85%	59.74%	59.30%	55.10%	53.05%
ST	64.54%	62.90%	59.97%	58.39%	53.26%	49.99%	49.99%	62.94%	56.39%	50.49%	59.24%	63.70%	63.80%	61.44%	57.17%	59.59%	59.07%	55.10%	53.10%
ET	65.39%	63.43%	60.14%	58.44%	53.01%	49.99%	49.99%	63.53%	56.33%	50.47%	59.47%	64.25%	64.38%	61.90%	56.87%	59.86%	59.37%	55.13%	53.12%
NN	65.89%	64.37%	60.45%	58.11%	50.46%	50.84%	50.59%	64.27%	56.25%	50.41%	59.48%	65.32%	65.13%	62.76%	55.50%	60.10%	59.97%	56.03%	53.73%

Table 8: Accuracies and coverage difference rates ω of our attacks on SVHN

(a) Gaussian noise, σ– range of standard deviation

σ		$[0, 0.1]$			$[0, 0.2]$			$[0, 0.3]$	
LTT		63.49%			61.65%			60.09%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	58.38%	61.50%	65.06%	58.29%	60.49%	62.45%	57.47%	59.25%	60.03%
ω	21.33%	21.35%	19.31%	22.93%	19.67%	14.82%	20.52%	15.31%	7.32%

(b) Adversarial noise

PGD		1-step			10-step			loss=10	
LTT		61.39%			52.03%			50.72%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	63.33%	61.52%	62.10%	56.02%	53.18%	51.35%	56.76%	56.98%	55.51%
ω	11.86%	1.29%	8.15%	25.55%	32.16%	39.61%	40.71%	43.60%	45.09%

(c) JPEG compression, q–compression quality

q		50			10			1	
LTT		63.55%			57.47%			50.92%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	58.28%	61.44%	65.22%	57.69%	58.74%	58.67%	51.62%	52.21%	52.68%
ω	21.52%	21.42%	19.89%	27.25%	21.69%	11.78%	6.09%	9.94%	13.56%

(d) Scaling, r–scaling factor

δ		0.5			1.5			10	
LTT		59.81%			64.00%			64.10%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	58.85%	61.09%	62.03%	58.40%	61.78%	66.16%	58.38%	61.76%	66.21%
ω	28.61%	25.91%	18.50%	21.87%	21.86%	21.38%	21.68%	21.67%	21.29%

(e) Filtering

δ		Clarendon			Gingham			Moon	
LTT		63.04%			57.45%			60.16%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	58.79%	62.13%	65.00%	57.25%	58.38%	58.46%	58.17%	60.83%	62.55%
ω	23.65%	22.96%	18.78%	29.72%	24.30%	14.65%	28.35%	26.68%	20.49%

(f) Rotation, δ–range of rotation degree

δ		$[0°, 10°]$			$[0°, 20°]$			$[0°, 30°]$	
LTT		59.77%			56.34%			53.74%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	57.69%	59.58%	61.11%	55.19%	56.07%	56.53%	52.91%	53.51%	54.87%
ω	26.24%	23.06%	16.88%	23.33%	15.91%	5.67%	17.82%	11.84%	6.29%

E.3 EVALUATION ON DEFENDED MODELS

We mainly evaluate our attacks on undefended models in Section 4. Here we provide evaluation results of CIFAR-10 models trained with two popular defense strategies: (1) large weight decay rate (Shokri et al., 2017), and (2) deferentially-private (DP) training (Abadi et al., 2016). We assume that the attacker knows the defense algorithms and the parameters so that the attacker can also train shadow models with the same defenses. Note that we only evaluate two defenses as our goal is not to evaluate the performance of state-of-the-art defenses but to show the effectiveness of current defenses against all MIAs.

For the first defense, we set the weight decay rate to 0.1 (target model trained with 5e-4) and trained the model until convergence. For DP, we applied the implementation from *Opacus* (Facebook, 2020) with an RMSprop optimizer for better convergence, a gradient clipping value of 1.2, a noise multiplier of 2.3, and trained for 100 epochs, which achieves (5, 1e-5)-differential privacy. All the other training settings are the same as in Section 4.1 in order to control variates. The final models trained with a large weight decay rate have an average of 85% training accuracy and 64% testing accuracy, and the final models trained with DP have an average of 40% training accuracy and 39% testing accuracy.

Table 9, Table 10 and Table 11 show the attack results of current attacks and our attacks on the defended models. We can see that both defenses are effective against all MIAs, including ours. DP with a small privacy budget provides the best privacy guarantee, reducing all attacks to random guesses, with a trade-off between utility as the model accuracy is considerably lower than undefended ones.

While loss-thresholding with transformations is still working better than the current attacks, we notice that the performance of reverse transformation attack is ineffective under the defenses when the ϵ settings are the same with attacking undefended models in Section 4. This is due to the defenses increasing the average loss so that both training and testing examples are far away from the robust area of a small ϵ. However, an adaptive attacker can always select a better ϵ for the reverse transformation attack. In Figure 4, we compare reverse transformation attacks with different ϵ-robust areas on the model with weight decay defense. We can see that a proper selection of $\epsilon = 0.5$ can still lead to a better attack accuracy than the random guessing baseline and the loss-thresholding attacks. How to optimally select ϵ is an interesting topic and is left as our future work.

Table 9: Accuracies of current attacks on defended models

(a) Weight decay

	Original	Gaussian(σ)			Adversarial			JPEG(q)			Scaling(r)			Filtering			Rotation(δ)		
		[0,0.1]	[0,0.2]	[0,0.3]	1-step	10-step	loss=10	50	10	1	0.5	1.5	10	Clarendon	Gingham	Moon	$[0°, 10°]$	$[0°, 20°]$	$[0°, 30°]$
CC	60.98%	60.50%	58.59%	57.03%	58.54%	50.13%	50.13%	58.60%	54.54%	51.17%	52.55%	59.16%	59.35%	60.67%	59.54%	59.73%	58.63%	56.16%	54.60%
LT	61.33%	60.53%	58.19%	56.34%	56.01%	50.11%	50.11%	57.94%	53.52%	50.78%	51.86%	58.61%	58.90%	61.11%	59.31%	59.26%	58.32%	55.81%	54.23%
ST	61.35%	60.55%	58.20%	56.34%	56.06%	50.12%	50.12%	57.92%	53.52%	50.78%	51.90%	58.63%	58.90%	61.02%	59.30%	59.27%	58.32%	55.78%	54.23%
ET	61.33%	60.55%	58.19%	56.32%	56.02%	50.11%	50.11%	57.95%	53.54%	50.79%	51.86%	58.63%	58.89%	61.11%	59.33%	59.24%	58.33%	55.81%	54.23%
NN	57.81%	57.21%	55.41%	53.53%	46.96%	49.86%	49.89%	54.76%	51.14%	49.88%	50.32%	54.83%	55.26%	57.26%	55.71%	55.73%	55.09%	53.23%	52.29%

(b) DP

	Original	Gaussian(σ)			Adversarial			JPEG(q)			Scaling(r)			Filtering			Rotation(δ)		
		[0,0.1]	[0,0.2]	[0,0.3]	1-step	10-step	loss=10	50	10	1	0.5	1.5	10	Clarendon	Gingham	Moon	$[0°, 10°]$	$[0°, 20°]$	$[0°, 30°]$
CC	50.67%	50.70%	50.62%	50.48%	50.15%	50.00%	50.00%	50.63%	50.50%	50.24%	50.34%	50.59%	50.58%	50.70%	50.52%	50.62%	50.63%	50.70%	50.51%
LT	50.79%	50.72%	50.79%	50.53%	50.18%	50.01%	50.01%	50.72%	50.46%	50.30%	50.27%	50.70%	50.74%	50.88%	50.68%	50.68%	50.70%	50.66%	50.53%
ST	50.55%	50.54%	50.45%	50.32%	50.17%	50.04%	50.04%	50.31%	50.26%	50.08%	50.25%	50.61%	50.65%	50.59%	50.43%	50.50%	50.46%	50.28%	50.19%
ET	50.86%	50.73%	50.82%	50.53%	50.23%	50.00%	50.00%	50.72%	50.49%	50.28%	50.35%	50.79%	50.81%	50.94%	50.69%	50.69%	50.74%	50.68%	50.57%
NN	49.82%	49.93%	50.10%	50.14%	49.63%	50.07%	50.17%	49.90%	50.05%	50.18%	49.97%	50.23%	50.09%	49.92%	50.01%	50.08%	50.13%	50.14%	50.18%

Table 10: Accuracies and coverage difference rates ω of our attacks on CIFAR-10 models trained with large weight decay rate

(a) Gaussian noise, σ– range of standard deviation

σ		$[0, 0.1]$			$[0, 0.2]$			$[0, 0.3]$	
LTT		60.57%			58.50%			56.88%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	50.08%	50.07%	50.09%	49.91%	50.02%	50.08%	50.27%	50.06%	50.07%
ω	49.40%	49.66%	49.83%	48.86%	49.82%	50.28%	49.33%	50.43%	50.12%

(b) Adversarial noise

PGD		1-step			10-step			loss=10	
LTT		60.33%			50.75%			50.75%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	50.06%	49.82%	50.04%	50.05%	50.06%	50.10%	50.04%	49.77%	49.74%
ω	48.98%	50.19%	50.26%	50.04%	49.98%	50.21%	50.30%	50.34%	49.82%

(c) JPEG compression, q–compression quality

q		50			10			1	
LTT		58.50%			54.65%			51.39%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	50.26%	50.27%	49.74%	49.90%	50.17%	49.95%	50.27%	50.11%	50.07%
ω	48.94%	50.48%	49.82%	48.97%	49.64%	49.72%	49.81%	50.17%	50.10%

(d) Scaling, r–scaling factor

δ		0.5			1.5			10	
LTT		52.98%			59.13%			59.38%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	50.14%	49.89%	50.23%	50.33%	50.25%	50.09%	50.13%	50.20%	49.92%
ω	49.03%	50.27%	50.28%	48.66%	49.67%	49.90%	48.87%	50.31%	49.83%

(e) Filtering

δ		Clarendon			Gingham			Moon	
LTT		61.03%			59.55%			59.75%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	49.98%	49.96%	49.76%	50.13%	49.72%	49.89%	50.03%	49.70%	50.08%
ω	49.13%	50.01%	49.95%	49.08%	50.51%	49.83%	48.79%	50.09%	49.99%

(f) Rotation, δ–range of rotation degree

δ		$[0°, 10°]$			$[0°, 20°]$			$[0°, 30°]$	
LTT		58.64%			56.28%			54.63%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	50.17%	50.01%	49.89%	49.88%	50.01%	49.81%	49.86%	49.95%	49.97%
ω	43.00%	43.20%	43.70%	38.79%	39.28%	39.15%	34.65%	34.91%	34.93%

Table 11: Accuracies and coverage difference rates ω of our attacks on CIFAR-10 models trained with DP

(a) Gaussian noise, σ– range of standard deviation

σ		$[0, 0.1]$			$[0, 0.2]$			$[0, 0.3]$	
LTT		50.76%			50.64%			50.61%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	50.22%	50.25%	50.02%	49.95%	50.01%	50.37%	50.19%	49.96%	50.04%
ω	47.32%	50.06%	49.92%	47.61%	50.18%	49.62%	48.22%	50.05%	49.99%

(b) Adversarial noise

PGD		1-step			10-step			loss=10	
LTT		50.76%			50.67%			50.67%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	50.14%	50.06%	49.96%	50.04%	50.11%	49.79%	50.08%	49.96%	50.07%
ω	47.98%	49.57%	49.79%	49.47%	49.93%	50.07%	49.05%	49.86%	49.76%

(c) JPEG compression, q–compression quality

q		50			10			1	
LTT		50.70%			50.50%			50.27%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	49.96%	50.08%	50.05%	50.17%	50.05%	50.28%	49.98%	50.11%	50.12%
ω	46.91%	49.38%	50.12%	47.48%	50.05%	49.52%	47.81%	50.25%	49.71%

(d) Scaling, r–scaling factor

δ		0.5			1.5			10	
LTT		50.39%			50.70%			50.73%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	49.96%	49.92%	50.07%	49.99%	50.12%	49.93%	50.09%	49.90%	50.00%
ω	48.04%	50.21%	50.02%	47.00%	49.86%	49.89%	47.33%	49.70%	50.00%

(e) Filtering

δ		Clarendon			Gingham			Moon	
LTT		50.86%			50.68%			50.70%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	50.25%	49.71%	50.04%	49.94%	50.28%	49.78%	49.90%	50.16%	49.89%
ω	46.54%	49.79%	49.97%	46.94%	49.71%	49.96%	47.67%	49.88%	50.07%

(f) Rotation, δ–range of rotation degree

δ		$[0°, 10°]$			$[0°, 20°]$			$[0°, 30°]$	
LTT		50.66%			50.52%			50.46%	
ϵ	0.1	0.01	0.001	0.1	0.01	0.001	0.1	0.01	0.001
RT	50.35%	50.07%	50.16%	49.56%	49.91%	50.14%	49.70%	50.12%	49.90%
ω	48.28%	49.37%	49.63%	47.58%	48.77%	48.86%	46.64%	48.10%	48.34%

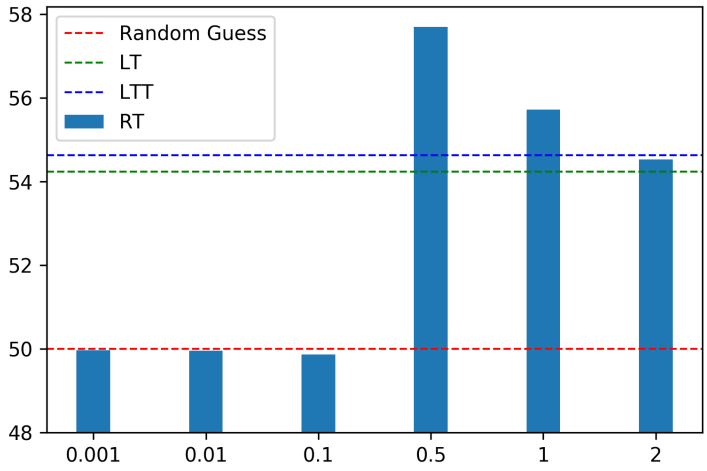


Figure 4: Comparison of different ϵ-robust areas for RT attack (CIFAR-10 with weight-decay defense, Rotation of $[0°, 30°]$). The x-axis represents the ϵ, and the y-axis represents the attack accuracy.