Membership Inference Attack Using Self Influence Functions (Supplementary Material)

Gilad Cohen
Tel Aviv University
giladco1@post.tau.ac.il

Raja Giryes
Tel Aviv University
raja@tauex.tau.ac.il

The supplementary material is organized as follows:

- Appendix A provides pseudo codes to outline the fitting and inference of our membership inference (MI) attack model.
- Appendix B explains in detail how the self-influence measure was approximated for SIF and adaSIF.
- Appendix C lists the hardware (CPUs & GPUs) we used for training the target models and for fitting/evaluating our attack models.
- Appendix D reports the train/test accuracies for all the target models.
- Appendix E compares our MI attack to baselines for target models trained on AlexNet and DenseNet architectures.
- Appendix F reports the precision and recall metrics on members and non-members, for our attack model and baselines.
- Appendix G advocates the use of our proposed adaSIF over a naive SIF ensemble.
- Appendix H compares the attack performance of adaSIF to baselines, for target models trained with data augmentations on AlexNet and DenseNet.
- Appendix I shows that SIF and adaSIF attack can be applicable to setups with limited membership knowledge.
- Appendix J compares our MI attack methods to an additional white-box attack.

Appendix A: SIF algorithm

Algorithm A1 summarizes the fitting of our self-influence function (SIF) attack model $\mathcal{A}$. For every sample in the training set $\mathcal{D}_{mem}^{train}$ or $\mathcal{D}_{non-mem}^{train}$ (defined in Section 4.2 in the main paper), we collect the I_{SIF} measure (Eq. (3)) together with a variable m that indicates if the target model h predicted the same class as the groundtruth label. These values are then used to calculate the parameters, τ_1 and τ_2, of the attack model $\mathcal{A}$ that is provided by Algorithm A2 (line #25).

The procedure in Algorithm A2 aims to find an interval (τ_1, τ_2) that best encapsulates only the members, i.e., we want to have that most of the members' SIF values are inside (τ_1, τ_2) and most of the non-members' SIF values are outside this range. Since the SIF values distribution does not resemble a Gaussian (see Figure 1 in the main paper), we consider 1000 samples distributed uniformly around both the members' minimum and maximum values (lines #10-11). For every possible pair τ_1, τ_2 (#lines 14-15) we calculate the balanced accuracy as defined in Eq. (5) in the main paper. The optimal threshold pair is selected based on a maximization of the balanced accuracy on the training set.

Lastly, Algorithm A3 shows the inference of our attack model $\mathcal{A}$. Given a target model h and data sample $z = (x, y)$, we calculate the SIF value s and query h for its class prediction. If both

conditions are met: (i) $s \in (\tau_1, \tau_2)$ and (ii) $y = \hat{y}$ (where $\hat{y} = h(x; \theta)$), then $\mathcal{A}$ predicts z as a member. Otherwise, z is predicted as a non-member.

Algorithm A1 Fitting self-influence function (SIF) attack

Input: Training set $\left\{\mathcal{D}_{mem}^{train} \cup \mathcal{D}_{non-mem}^{train}\right\} \subset \mathcal{X} \times \mathcal{Y}$
Input: $h(x; \theta)$ Pre-trained target model with parameters θ
Output: Attack model $\mathcal{A}(x, y; \tau_1, \tau_2)$ ▷ A membership inference predictor

Initialize: SIF_m=[], SIF_{nm}=[] ▷ SIF values
Initialize: M_m=[], M_{nm}=[] ▷ Whether the $h(x, \theta)$ class predictions matches the label
for $z = (x, y)$ in $\mathcal{D}_{mem}^{train}$ **do**
    $s \leftarrow I_{SIF}(z)$ ▷ Eq. (3) in the main paper
    $\hat{y} \leftarrow h(x; \theta)$ ▷ Query target model
    if $\hat{y} == y$ **then**
        $m \leftarrow 1$
    else
        $m \leftarrow 0$
    end if
    SIF_m.append(s)
    M_m.append(m)
end for
for $z = (x, y)$ in $\mathcal{D}_{non-mem}^{train}$ **do**
    $s \leftarrow I_{SIF}(z)$
    $\hat{y} \leftarrow h(x; \theta)$
    if $\hat{y} == y$ **then**
        $m \leftarrow 1$
    else
        $m \leftarrow 0$
    end if
    SIF_{nm}.append(s)
    M_{nm}.append(m)
end for
set τ_1, τ_2 := SETTHRESHOLDS(SIF_m, M_m, SIF_{nm}, M_{nm})

Algorithm A2 Setting τ_1 and τ_2 thresholds for attack model $\mathcal{A}$

procedure SETTHRESHOLDS(SIF_m, M_m, SIF_{nm}, M_{nm})
    $N_1 = |SIF_m|$ ▷ N_1 is the total number of members
    $N_2 = |SIF_{nm}|$ ▷ N_2 is the total number of non-members
    $best\_acc \leftarrow 0$
    $best\_\tau_1 \leftarrow -\infty$
    $best\_\tau_2 \leftarrow \infty$
    $SIF_m^{min} \leftarrow min(SIF_m)$
    $SIF_m^{max} \leftarrow max(SIF_m)$
    $\delta \leftarrow SIF_m^{max} - SIF_m^{min}$

    $min\_arr :=$ linspace$(SIF_m^{min} - \frac{\delta}{2}, SIF_m^{min} + \frac{\delta}{2}, 1000)$
    $max\_arr :=$ linspace$(SIF_m^{max} - \frac{\delta}{2}, SIF_m^{max} + \frac{\delta}{2}, 1000)$
    for i in $[1 : 1000]$ **do**
        for j in $[1 : 1000]$ **do**
            $\tau_1 \leftarrow min\_arr[i]$
            $\tau_2 \leftarrow max\_arr[j]$
            Initialize: $\hat{y}_m$=[], $\hat{y}_{nm}$=[] ▷ Set MI prediction vectors for members and non-members
            for k in $[1 : N_1]$ **do**
                if $\tau_1 < SIF_m[k]$ and $SIF_m[k] < \tau_2$ and $M_m[k] == 1$ **then**
                    $\hat{y}_m$.append(1)
                else
                    $\hat{y}_m$.append(0)
                end if
            end for
            for k in $[1 : N_2]$ **do**
                if $\tau_1 < SIF_{nm}[k]$ and $SIF_{nm}[k] < \tau_2$ and $M_{nm}[k] == 1$ **then**
                    $\hat{y}_{nm}$.append(1)
                else
                    $\hat{y}_{nm}$.append(0)
                end if
            end for
            $acc \leftarrow$ Balanced Acc($\hat{y}_m$, $\hat{y}_{nm}$) ▷ Eq. (5) in the main paper
            if $acc > best\_acc$ **then**
                $best\_acc \leftarrow acc$
                $best\_\tau_1 \leftarrow \tau_1$
                $best\_\tau_2 \leftarrow \tau_2$
            end if
        end for
    end for
    return $best\_\tau_1$, $best\_\tau_2$
end procedure

Algorithm A3 SIF inference

Input: $h(x;\theta)$ Pre-trained target model with parameters θ
Input: $\mathcal{A}(x, y; \tau_1, \tau_2)$ Pre-trained attack model with parameters τ_1 and τ_2
Input: $z = (x, y)$ Data sample
Output: Membership inference prediction ▷ 1 for member and 0 for non-member

$s \leftarrow I_{SIF}(z)$ ▷ Eq. (3) in the main paper
$\hat{y} \leftarrow h(x;\theta)$ ▷ Query target model
if $\tau_1 < s$ and $s < \tau_2$ and $\hat{y} == y$ **then**
    return 1
else
    return 0
end if

Appendix B: SIF and adaSIF calculation

Here we explain in detail how we calculated the I_{SIF} and I_{adaSIF} values in Eq. (3) and Eq. (4) in the main paper, respectively.

B1. SIF

The vanilla SIF value is given by:

$$I_{SIF}(z) = -\nabla_\theta L(z, \hat{\theta})^T H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta}).$$

Since the Hessian H_θ and its inverse are not feasible to compute due to millions of parameters in deep neural networks (DNNs), we avoid their computation completely and follow the method shown in [3]. We approximate I_{SIF} using Hessian vector products (HVPs):

$$I_{SIF}(z) = -\underbrace{H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta})}_{s(z)} \cdot \underbrace{\nabla_\theta L(z, \hat{\theta})}_{grad_z}. \tag{B1}$$

Koh and Liang [3] employed this HVP and approximated $s(z)$ using stochastic estimation [1], while iterating over data points from the training set. In our vanilla SIF case, we use their $s(z)$ approximation with one iteration since we consider the self-influence of a single data point. The $grad_z$ value is the gradient map from the loss to the image plane, and is calculated with a simple back-propagation pass.

B2. adaSIF

Here we consider a scenario where the target model was trained with data augmentations. Let $z = (x, y)$ denote an original sample and I be a random data augmentation operator sampled from the family of training augmentation distribution $\mathcal{T}\big(I \sim \mathcal{T}\big)$. We approximate $s(z)$ and $grad_z$ in Eq. (B1) by taking their expected value over these transformations. Formally, we calculate:

$$\begin{aligned} I_{adaSIF}(z) &\stackrel{\text{def}}{=} -\mathbb{E}_{I\sim\mathcal{T}}\big[s(z)\big] \cdot \mathbb{E}_{I\sim\mathcal{T}}\big[grad_z\big] \\ &= -\underbrace{\mathbb{E}_{I\sim\mathcal{T}}\Big[H_{\hat{\theta}}^{-1} \nabla_\theta L\big(I(x), y, \hat{\theta}\big)\Big]}_{(i)} \cdot \underbrace{\mathbb{E}_{I\sim\mathcal{T}}\Big[\nabla_\theta L\big(I(x), y, \hat{\theta}\big)\Big]}_{(ii)}. \end{aligned} \tag{B2}$$

For approximating (i) we employ the same stochastic estimation as used by Koh and Liang, but instead of iterating over different data points, we iterate over a set of image transformations. (ii) is calculated by averaging gradient maps of 128 different image transformations $I(x)$.

Appendix C: Hardware setup

All the target models and attack models were trained and evaluated on a machine with a GPU of type NVIDIA GeForce RTX 2080 Ti, which has 11 GB of VRAM. For training the target models we utilized 4 threads of Intel Xeon Silver 4114 CPU. The target models were evaluated using a single CPU core. All the attack models' fitting and inference were performed using a single GPU and a single CPU core.

Appendix D: Accuracy of target models

Table D1 and Table D2 report the *training*, *validation*, and *test* accuracies of target models trained without and with data augmentations, respectively, as defined in Section 4.1 in the main paper. Notice that Tiny ImageNet was not trained on $\mathcal{M}$-1 since the dataset has 200 labels whereas the smallest target model has only 100 data points for training. All target models exhibit sufficient test accuracy for a meaningful MI analysis.

Target Models		CIFAR-10			CIFAR-100			Tiny ImageNet		
		Train	Val	Test	Train	Val	Test	Train	Val	Test
AlexNet	$\mathcal{M}$-1	21.00	19.80	19.95	1.00	2.00	1.74	-	-	-
	$\mathcal{M}$-2	54.30	35.56	33.27	100.00	6.00	5.75	0.60	1.20	0.96
	$\mathcal{M}$-3	100.00	50.52	51.77	36.88	13.00	12.59	20.02	4.84	4.12
	$\mathcal{M}$-4	100.00	61.72	60.29	99.98	17.20	18.38	10.99	4.32	3.93
	$\mathcal{M}$-5	100.00	65.68	64.33	99.99	27.24	26.60	100.00	8.44	7.30
	$\mathcal{M}$-6	100.00	67.56	67.70	99.98	26.56	27.32	23.02	11.22	10.46
	$\mathcal{M}$-7	100.00	71.28	70.55	99.96	34.20	33.40	22.42	14.54	13.76
ResNet18	$\mathcal{M}$-1	100.00	19.32	19.49	100.00	3.48	3.35	-	-	-
	$\mathcal{M}$-2	100.00	39.00	38.62	100.00	11.88	10.91	99.60	2.80	2.87
	$\mathcal{M}$-3	100.00	57.24	56.94	100.00	22.84	22.80	99.98	8.56	8.51
	$\mathcal{M}$-4	100.00	67.88	65.00	99.99	31.24	30.19	100.00	14.38	14.40
	$\mathcal{M}$-5	100.00	76.20	74.18	100.00	38.32	40.10	100.00	19.90	19.42
	$\mathcal{M}$-6	100.00	76.04	74.42	100.00	48.20	47.12	100.00	23.38	23.04
	$\mathcal{M}$-7	100.00	76.96	76.30	99.99	48.56	47.88	100.00	25.84	25.07
DenseNet	$\mathcal{M}$-1	100.00	24.80	24.82	100.00	3.24	2.86	-	-	-
	$\mathcal{M}$-2	100.00	45.84	45.86	99.80	11.40	10.74	99.20	3.24	3.04
	$\mathcal{M}$-3	100.00	65.84	64.88	99.64	27.48	25.96	42.74	11.14	10.72
	$\mathcal{M}$-4	100.00	75.16	74.28	99.99	36.24	36.11	34.02	16.20	15.18
	$\mathcal{M}$-5	100.00	77.80	77.51	97.27	41.32	40.48	27.79	19.38	19.01
	$\mathcal{M}$-6	100.00	81.08	79.92	94.24	45.64	44.70	46.77	23.46	23.27
	$\mathcal{M}$-7	100.00	82.96	81.97	81.71	47.16	46.30	41.97	24.90	25.06

Table D1: The *training*, *validation*, and *test* accuracies [%] for all the target models used in the paper, that were trained without data augmentations.

Target Models		CIFAR-10			CIFAR-100			Tiny ImageNet		
		Train	Val	Test	Train	Val	Test	Train	Val	Test
AlexNet	$\mathcal{M}$-1	39.00	25.76	26.43	15.00	3.12	2.23	-	-	-
	$\mathcal{M}$-2	99.40	44.28	42.06	97.60	8.96	7.71	98.60	2.86	2.77
	$\mathcal{M}$-3	99.60	69.68	67.64	41.42	14.80	13.46	100.00	9.50	9.11
	$\mathcal{M}$-4	99.76	74.56	73.70	49.00	22.36	21.45	13.24	6.70	6.51
	$\mathcal{M}$-5	99.86	78.16	77.14	99.98	28.60	28.54	14.47	9.18	8.82
	$\mathcal{M}$-6	99.91	80.40	79.08	99.96	32.88	31.89	99.99	13.76	12.49
	$\mathcal{M}$-7	99.44	79.92	80.04	99.96	36.72	36.46	99.90	17.74	16.82
ResNet18	$\mathcal{M}$-1	100.00	21.96	22.66	100.00	3.80	3.97	-	-	-
	$\mathcal{M}$-2	99.80	42.16	41.15	100.00	12.80	12.07	99.80	3.54	2.92
	$\mathcal{M}$-3	100.00	69.16	67.83	100.00	31.28	31.42	100.00	15.50	15.27
	$\mathcal{M}$-4	100.00	79.08	76.91	99.99	48.96	48.27	100.00	26.40	24.87
	$\mathcal{M}$-5	100.00	86.56	85.89	100.00	56.52	57.09	99.99	30.26	30.67
	$\mathcal{M}$-6	100.00	91.60	90.25	100.00	64.44	62.89	99.99	35.18	35.51
	$\mathcal{M}$-7	100.00	91.84	90.29	100.00	63.72	63.62	99.99	39.40	38.95
DenseNet	$\mathcal{M}$-1	100.00	24.52	24.23	100.00	4.36	4.17	-	-	-
	$\mathcal{M}$-2	100.00	51.48	49.55	97.50	10.72	10.29	98.40	3.32	3.35
	$\mathcal{M}$-3	99.74	73.76	71.99	99.38	34.20	34.57	88.92	13.06	12.23
	$\mathcal{M}$-4	99.62	80.28	79.15	88.64	44.56	43.09	71.37	22.02	20.42
	$\mathcal{M}$-5	99.63	84.88	84.09	93.07	50.12	48.76	57.25	26.16	25.70
	$\mathcal{M}$-6	99.47	87.72	85.25	88.41	54.24	52.40	59.69	30.88	29.89
	$\mathcal{M}$-7	99.36	87.32	85.96	79.97	55.56	54.71	63.91	33.86	33.10

Table D2: The *training*, *validation*, and *test* accuracies [%] for all the target models used in the paper, that were trained with data augmentations.

Appendix E: Comparison of MI attacks

Here we continue the MI attack comparison from Section 5.2 in the main paper, and include other architectures. Figure E1 presents the balanced accuracy on target models trained on AlexNet and DenseNet. We observe that in most cases SIF performs on par with current state-of-the-art (SOTA). A new SOTA is achieved for CIFAR-10 trained on DenseNet (Figure E1(d)).

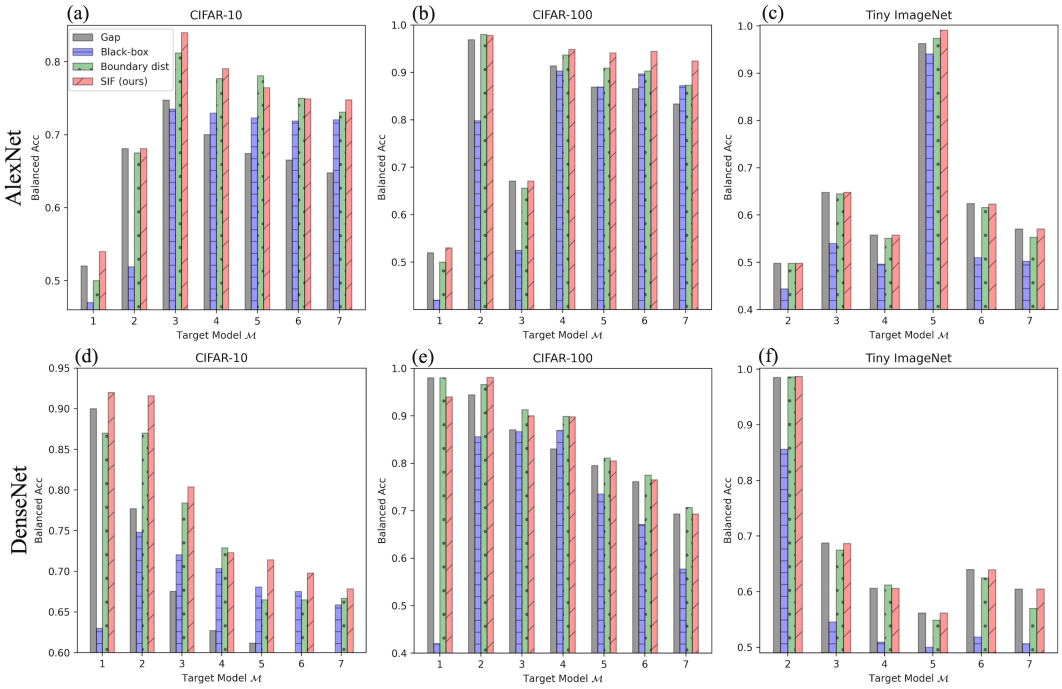

Figure E1: Comparison of our SIF attack with some baseline MI attacks: Gap, Black-box, and Boundary distance. The top and bottom rows show target models trained using AlexNet and DenseNet architectures, respectively. The x-axis indicates the attacked target model and the y-axis shows the balanced attack accuracy (Eq. (5) in the main paper).

Appendix F: Precision and recall

Tables F1-F3 show all attack models' precision, recall, and accuracy metrics for members and non-members, on target models trained with ResNet18, for CIFAR-10, CIFAR-100, and Tiny ImageNet, respectively. In addition to the excellent balanced accuracy we reported in the main paper (Section 5.2, SIF also achieves superb precision for members and recall for non-members, surpassing previous SOTA by a large margin for the majority of cases. These results demonstrate that our SIF attack does not suffer from the high False Alarm Rate (FAR) observed in many other MI inference attacks [6], making it a very reliable method for detecting training set samples as portrayed in [2, 9]. We Also observe prefect recall (~ 1.0) for members, matching our baselines and the results from [8, 10].

Target model	Attack model	Member			Non-member			Balanced Acc
		Acc	Precision	Recall	Acc	Precision	Recall	
$\mathcal{M}$-1	Gap	**1.00**	0.82	**1.00**	0.78	**1.00**	0.78	0.89
	Black-box	0.60	0.54	0.60	0.48	0.55	0.48	0.54
	Boundary dist	0.98	0.93	0.98	0.92	0.98	0.92	0.95
	SIF (ours)	**1.00**	**0.98**	**1.00**	**0.98**	**1.00**	**0.98**	**0.99**
$\mathcal{M}$-2	Gap	**1.00**	0.72	**1.00**	0.61	**1.00**	0.61	0.81
	Black-box	**1.00**	0.72	**1.00**	0.62	**1.00**	0.62	0.81
	Boundary dist	0.99	0.84	0.99	0.81	0.99	0.81	0.90
	SIF (ours)	**1.00**	**0.91**	**1.00**	**0.91**	**1.00**	**0.91**	**0.95**
$\mathcal{M}$-3	Gap	**1.00**	0.63	**1.00**	0.41	**1.00**	0.41	0.71
	Black-box	**1.00**	0.75	**1.00**	0.67	**1.00**	0.67	0.83
	Boundary dist	0.95	0.79	0.95	0.75	0.93	0.75	0.85
	SIF (ours)	**1.00**	**0.85**	**1.00**	**0.82**	**1.00**	**0.82**	**0.91**
$\mathcal{M}$-4	Gap	**1.00**	0.61	**1.00**	0.36	**1.00**	0.36	0.68
	Black-box	0.99	0.74	0.99	0.65	0.98	0.65	0.82
	Boundary dist	0.91	0.74	0.91	0.68	0.89	0.68	0.80
	SIF (ours)	0.99	**0.80**	0.99	**0.75**	0.99	**0.75**	**0.87**
$\mathcal{M}$-5	Gap	**1.00**	0.57	**1.00**	0.25	**1.00**	0.25	0.63
	Black-box	**1.00**	0.67	**1.00**	0.51	**1.00**	0.51	0.76
	Boundary dist	0.95	0.70	0.95	0.59	0.92	0.59	0.77
	SIF (ours)	0.99	**0.73**	0.99	**0.64**	0.98	**0.64**	**0.81**
$\mathcal{M}$-6	Gap	**1.00**	0.57	**1.00**	0.24	**1.00**	0.24	0.62
	Black-box	0.89	0.71	0.89	**0.63**	0.85	**0.63**	0.76
	Boundary dist	0.97	0.66	0.97	0.50	0.94	0.50	0.74
	SIF (ours)	0.98	**0.72**	0.98	0.62	0.96	0.62	**0.80**
$\mathcal{M}$-7	Gap	**1.00**	0.57	**1.00**	0.23	**1.00**	0.23	0.62
	Black-box	**1.00**	**0.70**	**1.00**	0.58	**1.00**	0.58	**0.79**
	Boundary dist	0.91	**0.70**	0.91	**0.62**	0.87	**0.62**	0.76
	SIF (ours)	**1.00**	0.69	**1.00**	0.55	**1.00**	0.55	0.78

Table F1: Accuracy, precision, and recall for members and non-members. Target models were trained on CIFAR-10 with ResNet18.

Target model	Attack model	Member			Non-member			Balanced Acc
		Acc	Precision	Recall	Acc	Precision	Recall	
$\mathcal{M}$-1	Gap	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	Black-box	0.14	0.39	0.14	0.78	0.48	0.78	0.46
	Boundary dist	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	SIF (ours)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
$\mathcal{M}$-2	Gap	**1.00**	0.87	**1.00**	0.85	**1.00**	0.85	0.92
	Black-box	**1.00**	0.87	**1.00**	0.85	**1.00**	0.85	0.93
	Boundary dist	**1.00**	0.96	**1.00**	0.95	**1.00**	0.95	0.98
	SIF (ours)	**1.00**	**0.99**	**1.00**	**0.99**	**1.00**	**0.99**	**1.00**
$\mathcal{M}$-3	Gap	**1.00**	0.81	**1.00**	0.76	**1.00**	0.76	0.88
	Black-box	**1.00**	0.94	**1.00**	0.93	**1.00**	0.93	0.97
	Boundary dist	0.99	0.94	0.99	0.94	0.99	0.94	0.96
	SIF (ours)	**1.00**	**0.98**	**1.00**	**0.98**	**1.00**	**0.98**	**0.99**
$\mathcal{M}$-4	Gap	**1.00**	0.76	1.00	0.69	**1.00**	0.69	0.85
	Black-box	**1.00**	0.92	1.00	0.91	**1.00**	0.91	0.96
	Boundary dist	**1.00**	0.91	1.00	0.90	**1.00**	0.90	0.95
	SIF (ours)	**1.00**	**0.96**	1.00	**0.96**	**1.00**	**0.96**	**0.98**
$\mathcal{M}$-5	Gap	**1.00**	0.71	**1.00**	0.60	**1.00**	0.60	0.80
	Black-box	**1.00**	0.91	**1.00**	0.91	**1.00**	0.91	0.95
	Boundary dist	0.97	0.88	0.97	0.86	0.97	0.86	0.92
	SIF (ours)	**1.00**	**0.96**	**1.00**	**0.95**	**1.00**	**0.95**	**0.98**
$\mathcal{M}$-6	Gap	**1.00**	0.68	**1.00**	0.53	**1.00**	0.53	0.77
	Black-box	0.99	0.90	0.99	0.89	0.99	0.89	0.94
	Boundary dist	0.99	0.86	0.99	0.84	0.98	0.84	0.91
	SIF (ours)	**1.00**	**0.94**	**1.00**	**0.93**	**1.00**	**0.93**	**0.97**
$\mathcal{M}$-7	Gap	**1.00**	0.68	**1.00**	0.52	**1.00**	0.52	0.76
	Black-box	0.99	0.88	0.99	0.86	0.99	0.86	0.93
	Boundary dist	0.98	0.83	0.98	0.80	0.97	0.80	0.89
	SIF (ours)	**1.00**	**0.91**	**1.00**	**0.90**	**1.00**	**0.90**	**0.95**

Table F2: Accuracy, precision, and recall for members and non-members. Target models were trained on CIFAR-100 with ResNet18.

Target model	Attack model	Member			Non-member			Balanced Acc
		Acc	Precision	Recall	Acc	Precision	Recall	
$\mathcal{M}$-2	Gap	**1.00**	0.96	**1.00**	0.96	**1.00**	0.96	0.98
	Black-box	0.94	0.92	0.94	0.91	0.94	0.91	0.93
	Boundary dist	0.99	**0.98**	0.99	**0.98**	0.99	**0.98**	0.98
	SIF (ours)	0.99	**0.98**	0.99	**0.98**	0.99	**0.98**	**0.99**
$\mathcal{M}$-3	Gap	**1.00**	0.91	**1.00**	0.91	**1.00**	0.91	0.95
	Black-box	**1.00**	0.96	**1.00**	0.96	**1.00**	0.96	0.98
	Boundary dist	**1.00**	0.98	**1.00**	0.98	**1.00**	0.98	0.99
	SIF (ours)	**1.00**	**0.99**	**1.00**	**0.99**	**1.00**	**0.99**	**1.00**
$\mathcal{M}$-4	Gap	**1.00**	0.87	**1.00**	0.86	**1.00**	0.86	0.93
	Black-box	**1.00**	0.98	**1.00**	0.98	**1.00**	0.98	**0.99**
	Boundary dist	**1.00**	0.97	**1.00**	0.96	**1.00**	0.96	0.98
	SIF (ours)	**1.00**	**0.99**	**1.00**	**0.99**	**1.00**	**0.99**	**0.99**
$\mathcal{M}$-5	Gap	**1.00**	0.84	**1.00**	0.81	**1.00**	0.81	0.90
	Black-box	**1.00**	0.97	**1.00**	0.97	**1.00**	0.97	0.99
	Boundary dist	0.99	0.96	0.99	0.96	0.99	0.96	0.98
	SIF (ours)	**1.00**	**0.99**	**1.00**	**0.99**	**1.00**	**0.99**	**1.00**
$\mathcal{M}$-6	Gap	**1.00**	0.82	**1.00**	0.78	**1.00**	0.78	0.89
	Black-box	**1.00**	0.95	**1.00**	0.95	**1.00**	0.95	0.97
	Boundary dist	0.99	0.95	0.99	0.94	0.99	0.94	0.97
	SIF (ours)	**1.00**	**0.97**	**1.00**	**0.97**	**1.00**	**0.97**	**0.98**
$\mathcal{M}$-7	Gap	**1.00**	0.80	**1.00**	0.75	**1.00**	0.75	0.88
	Black-box	0.99	**0.96**	0.99	**0.96**	0.99	**0.96**	**0.98**
	Boundary dist	**1.00**	0.95	**1.00**	0.95	**1.00**	0.95	0.97
	SIF (ours)	**1.00**	0.93	**1.00**	0.93	**1.00**	0.93	0.96

Table F3: Accuracy, precision, and recall for members and non-members. Target models were trained on Tiny ImageNet with ResNet18.

Appendix G: Naive SIF ensemble

In our paper we propose to use a naive ensemble of SIF measures (named "avgSIF") to attack target models that are trained with data augmentation. Here we formally define avgSIF and compare its results to adaSIF. Let $z = (x, y)$ denote an original sample and I be a random data augmentation operator sampled from the family of training augmentation distribution $\mathcal{T}\big(I \sim \mathcal{T}\big)$. Then we define the naive ensemble of SIF measures of z as:

$$\begin{aligned} I_{avgSIF}(z) &\stackrel{\text{def}}{=} \mathbb{E}_{I\sim\mathcal{T}}\Big[I_{SIF}\big(I(x), y\big)\Big] \\ &= \mathbb{E}_{I\sim\mathcal{T}}\Big[-\nabla_\theta L\big(I(x), y, \hat{\theta}\big)^T H_{\hat{\theta}}^{-1} \nabla_\theta L\big(I(x), y, \hat{\theta}\big)\Big]. \end{aligned} \tag{G1}$$

The above term calculates I_{SIF} scores (from Eq. (3) in the main paper) for 8 different transformations of the input image x, and averages them to get the I_{avgSIF} measure. The fitting and inference of the avgSIF attack are done similarly to the vanilla SIF attack (see Appendix A).

Table G1 shows the accuracy, precision, and recall metrics for members and non-members, calculated for avgSIF and adaSIF, for target models trained on ResNet18. We observe that adaSIF outperforms avgSIF, achieving a higher balanced accuracy for the vast majority of the target models. In addition, adaSIF maintains a higher precision for the members which translates to a lower FAR (False Alarm Rate). Therefore, adaSIF was chosen for evaluating MI with data augmentation in the main paper.

Dataset	Target model	Attack model	Member			Non-member			Balanced Acc
			Acc	Precision	Recall	Acc	Precision	Recall	
CIFAR-10	$\mathcal{M}$-1	adaSIF	0.940	**0.825**	0.940	0.800	0.930	0.800	**0.870**
		avgSIF	0.960	0.814	0.960	0.780	0.951	0.780	**0.870**
	$\mathcal{M}$-2	adaSIF	1.000	**0.808**	1.000	0.762	1.000	0.762	**0.881**
		avgSIF	0.998	0.800	0.998	0.750	0.997	0.750	0.874
	$\mathcal{M}$-3	adaSIF	0.993	**0.761**	0.993	0.688	0.990	0.688	**0.841**
		avgSIF	0.992	0.750	0.992	0.670	0.988	0.670	0.831
	$\mathcal{M}$-4	adaSIF	0.998	**0.679**	0.998	0.528	0.996	0.528	**0.763**
		avgSIF	0.998	0.659	0.998	0.484	0.995	0.484	0.741
	$\mathcal{M}$-5	adaSIF	0.985	**0.625**	0.985	0.408	0.965	0.408	**0.697**
		avgSIF	0.994	0.608	0.994	0.359	0.982	0.359	0.676
	$\mathcal{M}$-6	adaSIF	0.992	**0.606**	0.992	0.354	0.977	0.354	**0.673**
		avgSIF	0.997	0.600	0.997	0.334	0.992	0.334	0.666
	$\mathcal{M}$-7	adaSIF	0.982	0.599	0.982	0.342	0.951	0.342	0.662
		avgSIF	0.995	**0.601**	0.995	0.339	0.986	0.339	**0.667**
CIFAR-100	$\mathcal{M}$-1	adaSIF	1.000	**1.000**	1.000	1.000	1.000	1.000	**1.000**
		avgSIF	1.000	**1.000**	1.000	1.000	1.000	1.000	**1.000**
	$\mathcal{M}$-2	adaSIF	0.990	**0.994**	0.990	0.994	0.990	0.994	0.992
		avgSIF	0.996	**0.994**	0.996	0.994	0.996	0.994	**0.995**
	$\mathcal{M}$-3	adaSIF	0.995	**0.960**	0.995	0.958	0.995	0.958	**0.977**
		avgSIF	0.995	0.957	0.995	0.955	0.995	0.955	0.975
	$\mathcal{M}$-4	adaSIF	0.994	**0.883**	0.994	0.868	0.993	0.868	**0.931**
		avgSIF	0.988	0.866	0.988	0.848	0.986	0.848	0.918
	$\mathcal{M}$-5	adaSIF	0.997	**0.838**	0.997	0.807	0.997	0.807	**0.902**
		avgSIF	0.998	0.831	0.998	0.797	0.997	0.797	0.897
	$\mathcal{M}$-6	adaSIF	0.986	**0.837**	0.986	0.808	0.983	0.808	**0.897**
		avgSIF	0.990	0.795	0.990	0.744	0.987	0.744	0.867
	$\mathcal{M}$-7	adaSIF	0.986	**0.839**	0.986	0.811	0.983	0.811	**0.898**
		avgSIF	0.999	0.812	0.999	0.769	0.999	0.769	0.884
Tiny ImageNet	$\mathcal{M}$-2	adaSIF	0.996	0.996	0.996	0.996	0.996	0.996	0.996
		avgSIF	0.994	**1.000**	0.994	1.000	0.994	1.000	**0.997**
	$\mathcal{M}$-3	adaSIF	0.998	0.989	0.998	0.988	0.998	0.988	**0.993**
		avgSIF	0.989	**0.990**	0.989	0.990	0.989	0.990	0.990
	$\mathcal{M}$-4	adaSIF	0.994	**0.974**	0.994	0.973	0.994	0.973	**0.984**
		avgSIF	0.999	0.964	0.999	0.963	0.999	0.963	0.981
	$\mathcal{M}$-5	adaSIF	0.998	**0.960**	0.998	0.958	0.998	0.958	**0.978**
		avgSIF	0.997	0.956	0.997	0.954	0.997	0.954	0.976
	$\mathcal{M}$-6	adaSIF	0.999	**0.941**	0.999	0.937	0.999	0.937	**0.968**
		avgSIF	0.998	0.934	0.998	0.929	0.997	0.929	0.963
	$\mathcal{M}$-7	adaSIF	0.997	**0.935**	0.997	0.931	0.997	0.931	**0.964**
		avgSIF	0.992	0.930	0.992	0.926	0.991	0.926	0.959

Table G1: Comparison between MI attack performances of adaSIF and avgSIF. adaSIF is marginally better than avgSIF. We boldface the best member's precision and balanced accuracy.

Appendix H: Comparison of MI attacks with data augmentation

Here we continue the adaptive MI attack comparison from Section 5.4 in the main paper, and include other architectures. Figure H1 presents the balanced accuracy on target models trained on AlexNet and DenseNet, with data augmentations (random crop and horizontal flipping). We observe that SIF performs on par with current SOTA, however, utilizing adaSIF (red bar) achieves new SOTA in most cases.

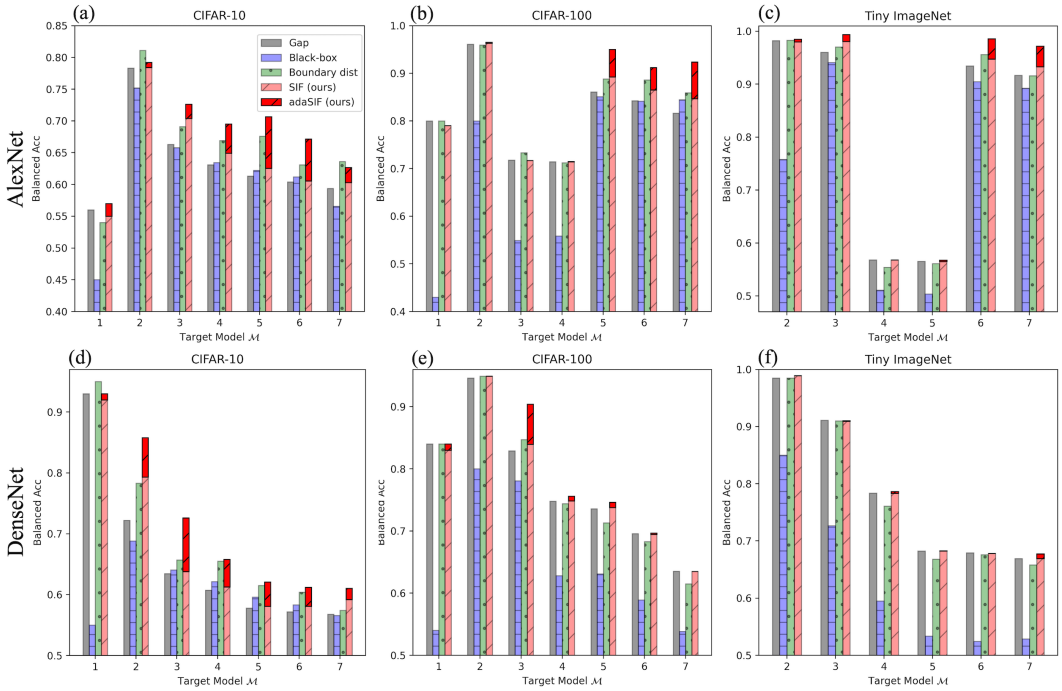

Figure H1: Comparison of our SIF (pink bar) and adaSIF (red bar) attacks with some baseline MI attacks: Gap, Black-box, and Boundary distance. The top and bottom rows show target models trained with data augmentations on AlexNet and DenseNet architectures, respectively. The x-axis indicates the attacked target model and the y-axis shows the balanced attack accuracy (Eq. (5) in the main paper).

Appendix I: Limited membership knowledge

In the main paper we reported the performances of SIF and adaSIF methods where they were trained using thousands of member and non-member data points. Here we evaluate our SIF and adaSIF attacks where the adversary has limited access to member data points, which is a more realistic scenario. Table I1 presents the balanced accuracy of our attacks on $\mathcal{M}$-7 target models. We show that in most cases, one can fit our attacks on merely 10 member data points and still obtain comparable membership leakage.

Target Model	$\lvert\mathcal{D}_{mem}^{train}\rvert$=2500	$\lvert\mathcal{D}_{mem}^{train}\rvert$=10
CIFAR-10	0.813	0.661
CIFAR-10 w. Data Aug	0.671	0.677
CIFAR-100	0.939	0.916
CIFAR-100 w. Data Aug	0.860	0.854
Tiny ImageNet	0.994	0.995
Tiny ImageNet w. Data Aug	0.960	0.919

Table I1: Balanced accuracies of SIF and adaSIF attacks for two data knowledge: (i) the attacks are fitted on 2500 members and 2500 non-members (middle column), and (ii) the attacks are fitted on 10 members and 10 non-members (right column). adaSIF and SIF were utilized on target models trained with- and without data augmentations, respectively.

Appendix J: Comparison to a white-box attack

In the main paper, we compare SIF and adaSIF to other SOTA black-box MI attacks, since researchers found that they perform similarly to white-box attacks [7, 6, 4]. However, Nasr et al. presented higher balanced accuracy for their white-box attack compared to other black-box methods, by training a large DNN attack model which gets as input all the hidden activations and gradients along the target model's layers [5]. Since they did not publish a code, we compare our MI attacks to their reported performances on CIFAR-100 for the same pre-trained target models they used: AlexNet, ResNet110, and DenseNet[1].

Table J1 compares the balanced accuracy of different MI attacks on CIFAR-100 for the pre-trained models used in [5]. We show that our adaSIF attack achieves a new SOTA for all the pre-trained networks, outperforming the white-box attack of Nasr et al.. We emphasize that we trained ResNet110 for our experiments since the pre-trained ResNet110 weights in the repository that [5] relied on cannot be used anymore. Our ResNet110 train/test accuracies are $99\%/71\%$, whereas Nasr et al. used a model with train/test accuracies of $89\%/73\%$. This might explain the large gap in MI performance for ResNet110 between their method and adaSIF.

We point out that our attack model utilizes only two fitted parameters (τ_1, τ_2), while Nasr et al. trained a heavy DNN for their attack model; this makes our method much more favorable for MI attack.

Architecture	Attack model	Balanced Acc
AlexNet	Gap	0.7421
	Black-box	0.6549
	Boundary dist	0.7362
	Nasr et al.	0.7510
	SIF	0.7454
	avgSIF	**0.7594**
	adaSIF	0.7516
ResNet110	Gap	0.6450
	Black-box	0.6640
	Boundary dist	0.6680
	Nasr et al.	0.6430
	SIF	0.6616
	avgSIF	0.6906
	adaSIF	**0.6944**
DenseNet	Gap	0.5885
	Black-box	0.7019
	Boundary dist	0.5380
	Nasr et al.	0.7430
	SIF	0.7242
	avgSIF	0.7402
	adaSIF	**0.7474**

Table J1: Comparison between our SIF/avgSIF/adaSIF MI attacks and the white-box attack proposed by Nasr et al.[5]. For completeness, we report also the balanced accuracies of the black-box methods we used in the paper.

[1]We utilized AlexNet, and DenseNet pre-trained DNNs from https://github.com/bearpaw/pytorch-classification, which is the same repository that was used in [5] for getting pre-trained models. CIFAR-100 was trained on ResNet110 using the script in https://github.com/bearpaw/pytorch-classification/blob/master/TRAINING.md since its pre-trained weights could not be loaded on the updated architecture.

References

[1] Naman Agarwal, Brian Bullins, and Elad Hazan. Second Order Stochastic Optimization in Linear Time. *ArXiv*, abs/1602.0, 2016.

[2] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership Inference Attacks From First Principles. In *43rd {IEEE} Symposium on Security and Privacy, {SP} 2022, San Francisco, CA, USA, May 22-26, 2022*, pages 1897–1914. IEEE, 2022.

[3] Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions. In *ICML*, volume 70, pages 1885–1894, 2017.

[4] Klas Leino and Matt Fredrikson. Stolen Memories: Leveraging Model Memorization for Calibrated White-Box Membership Inference. In *USENIX Security Symposium*, pages 1605–1622, 2020.

[5] Milad Nasr, R Shokri, and Amir Houmansadr. Comprehensive Privacy Analysis of Deep Learning: Stand-alone and Federated Learning under Passive and Active White-box Inference Attacks. *ArXiv*, abs/1812.0, 2018.

[6] Shahbaz Rezaei and Xin Liu. On the Difficulty of Membership Inference Attacks. *CVPR*, pages 7888–7896, 2021.

[7] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs Black-box: Bayes Optimal Strategies for Membership Inference. In *ICML*, 2019.

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

[9] Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. On the Importance of Difficulty Calibration in Membership Inference Attacks. In *ICLR*, 2022.

[10] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282, 2018.