# Silver Linings in the Shadows: Harnessing Membership Fingerprinting for Machine Unlearning (Extended Evaluation Section)

## I. EXPERIMENTS AND RESULTS

### A. Experimental Workflows

We used the following datasets to evaluate ReMI:

- FMNIST dataset [1] consisting a comprehensive set of 70,000 grayscale images. These images encompass a diverse array of 10 fashion items, including various types of clothing, dresses, shoes, handbags, and more.
- UTKFace [2] dataset comprises over 23,000 face images annotated with age, gender, and ethnicity, displaying varied pose, facial expression, illumination, occlusion, and resolution.
- STL10 [3] dataset comprises 13,000 images in 10 classes: airplane, bird, car, cat, deer, dog, horse, monkey, ship, and truck.
- CIFAR-10 [4] dataset features 60K color images ($32 \times 32$ pixels) across 10 classes, with 6K images per class. There are 50000 training and 10000 test images.

We divided each dataset into four equal-sized subsets. We use two subsets to train our target model, and the remaining two subsets are treated as shadow distribution, which will be utilized to conduct a membership inference attack on the target model for unlearning evaluation. All the image samples in the datasets are pre-processed before using them for model training, including re-sizing (64×64) and normalization. We test the efficacy of ReMI on four different architectures: SimpleCNN, ResNet18 [5], Xception [6], and VGG19 [7]. The SimpleCNN model (hereafter CNN) consists of 3 convolutional layers, each followed by a ReLU activation function and a max-pooling layer, along with two fully connected layers for classification. For the Xception architecture, we have implemented the model as specified in [6]. We have adopted the ResNet model with 18 layers [5]. The VGG19 model consists of a total of 19 layers, including 16 convolutional layers, 3 fully connected layers, and ReLU as its activation function [7]. All four deep learning architectures employ a softmax classifier in the output layer. The network configurations and training procedures are implemented in PyTorch. Each target model undergoes training for 50 epochs using the SGD optimizer with a learning rate of 1e-2, a momentum of 0.9, and a weight decay of 5e-4.

In ReMI we use the membership fingerprinting model to conduct unlearning. Then, we assess its effectiveness through a membership inference attack conducted before and after unlearning in white-box and black-box settings. The MIA utilizes the framework proposed in prior work and assumes the same threat model [8]. Subsequently, we test ReMI on four *forgetting datasets*, each with an increasing number of target samples selected evenly from each class that were identified as training data by the MF model with high probability. The sizes of these forgetting datasets are 0.01, 0.1, 0.25, and 0.5 of the target training datasets. Table I summarizes the number of data points in the forgetting data curated across all datasets.

**TABLE I:** Statistics of data in experiment dataset.

| Target | Samples | | Forgetting dataset | | | |
|---|---|---|---|---|---|---|
| Dataset | Train | Test | 0.01 | 0.1 | 0.25 | 0.5 |
| FMNIST | 17500 | 17500 | 179 | 1755 | 4380 | 8753 |
| UTKFace | 5503 | 5503 | 57 | 552 | 1378 | 2753 |
| STL10 | 3250 | 3250 | 40 | 331 | 818 | 1627 |
| CIFAR-10 | 15000 | 15000 | 154 | 1504 | 3754 | 7504 |

**TABLE II:** Hyperparameters of ReMI Unlearning

| Dataset | Architecture | Learning rate $\eta$ | | | | $\lambda_2$ |
|---|---|---|---|---|---|---|
| | | $\lvert D_f\rvert = 0.01$ | $\lvert D_f\rvert = 0.1$ | $\lvert D_f\rvert = 0.25$ | $\lvert D_f\rvert = 0.5$ | |
| FMNIST | CNN | 0.01 | 0.01 | 0.01 | 0.01 | 0.98 |
| | ResNet18 | 0.01 | 0.01 | 0.01 | 0.01 | 0.98 |
| | Xception | 0.01 | 0.01 | 0.01 | 0.01 | 0.98 |
| | VGG19 | 0.001 | 0.001 | 0.001 | 0.001 | 0.98 |
| UTKFace | CNN | 0.1 | 0.1 | 0.1 | 0.1 | 0.98 |
| | ResNet18 | 0.1 | 0.1 | 0.1 | 0.1 | 0.98 |
| | Xception | 0.1 | 0.1 | 0.1 | 0.1 | 0.98 |
| | VGG19 | 0.001 | 0.001 | 0.001 | 0.001 | 0.98 |
| STL10 | CNN | 0.1 | 0.1 | 0.1 | 0.1 | 0.98 |
| | ResNet18 | 0.1 | 0.1 | 0.1 | 0.1 | 0.98 |
| | Xception | 0.1 | 0.05 | 0.05 | 0.05 | 0.98 |
| | VGG19 | 0.0005 | 0.0005 | 0.0005 | 0.0005 | 0.98 |
| CIFAR10 | CNN | 0.01 | 0.01 | 0.01 | 0.01 | 0.98 |
| | ResNet18 | 0.01 | 0.01 | 0.01 | 0.01 | 0.98 |
| | Xception | 0.05 | 0.05 | 0.01 | 0.01 | 0.98 |
| | VGG19 | 0.1 | 0.05 | 0.05 | 0.001 | 0.98 |

Given a pre-trained target model, we apply our unlearning mechanism (ReMI) to update the parameters of the target model for each forgetting dataset. In particular, ReMI refines the parameters of the target model by minimizing the classification loss and unlearning loss based on the forgetting dataset, as elaborated in Section **??**. We also benchmark our algorithm with two unlearning approaches: Naive Retraining [9] and Fisher Unlearning [10]. The Naive Retraining algorithm retrains the target model on the remaining data by excluding specific forgetting data from the training dataset, while maintaining consistent hyper-parameter configurations similar to those used in the original training process. In the case of Fisher Unlearning, we followed the same implementation outlined in prior research to ensure the consistency of unlearning performance [10].

To assess the robustness of our unlearning framework, each

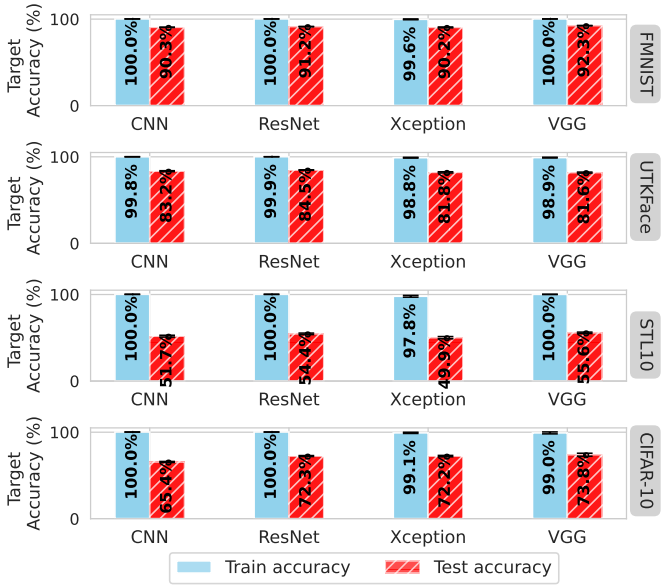


**Fig. 1:** The train and test classification accuracy of four datasets across four deep learning architectures before unlearning.

target model is trained using 20 different initializations (seed numbers), resulting in a total of 80 trained models. Each of these models is then assessed in both black-box and white-box settings. Following the implementation of white-box MIA models outlined in [8], we selected four attack features from the target model to serve as inputs to out MF and MIA models. These features include $I(w) \subset \{\hat{p}, \hat{y}, \mathcal{L}(\hat{y}, y), \nabla\mathcal{L}\}$. The white-box attack model was trained for 50 epochs using the Adam optimizer with a learning rate 1e-5. For black-box attack, we used the MIA implementation from IBM's Adversarial Robustness Toolbox [11].

Upon completion of the target model and the attack model, we initiate the unlearning process using the specified forgetting data as the input. In our unlearning framework, we explore various hyperparameter values that are used in ReMI's loss function, including the unlearning weight $\lambda_2$ and the learning rate $\eta$ during the unlearning process. It's worth noting that in our experimental setup, we defined $\lambda_1 = 1 - \lambda_2$. To ensure optimal unlearning performance, we identified the most suitable values for these hyperparameters across all deep learning architectures and datasets. The details of these optimal hyperparameter values are available in Table II.

### *B. Target Model and Approximation Function*

We first present the training and test accuracy of the target models based on different neural network architectures trained on four datasets. Figure 1 provides insight into the dataset complexities used in our experiments. Notably, the FMNIST dataset yields the highest overall test classification accuracy across all architectures, with CNN (90.3%), ResNet18 (91.2%), Xception (90.2%), and VGG19 (92.3%). In the case of the UTKFace and CIFAR-10 datasets, both training and test accuracies achieve lower averaged classification accuracy when compared to the four deep learning architectures trained

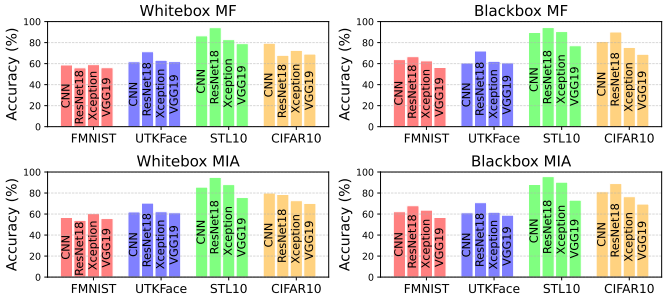


**Fig. 2:** Accuracy of the privacy approximation functions (MIA and MF) before unlearning in both white-box and black-box forms across all datasets. The results suggest that MIA and MF models perform similarly. Thus, either of these models can be used as the privacy approximation function to guide ReMI's unlearning process.

**TABLE III:** Classification accuracy of target models before and after unlearning for forgetting and remaining data ($D_f = 10\%$).

| Data | | | Phase | UTKFace | FMNIST | STL10 | CIFAR10 |
|---|---|---|---|---|---|---|---|
| CNN | Train | $D_f$ | Before | 100 | 100 | 100 | 100 |
| | | | After | 100 | 100 | 100 | 100 |
| | | $D_r$ | Before | 100 | 100 | 100 | 100 |
| | | | After | 96.3 | 98.5 | 98.6 | 99 |
| | Test | | Before | 83.4 | 90.1 | 51.9 | 65.4 |
| | | | After | 82.6 | 89.9 | 53.3 | 64 |
| ResNet18 | Train | $D_f$ | Before | 100 | 100 | 100 | 100 |
| | | | After | 100 | 100 | 100 | 100 |
| | | $D_r$ | Before | 100 | 100 | 100 | 100 |
| | | | After | 99 | 100 | 100 | 100 |
| | Test | | Before | 84.3 | 91.1 | 54.6 | 72.6 |
| | | | After | 84.3 | 91.6 | 53.2 | 71.3 |
| Xception | Train | $D_f$ | Before | 100 | 100 | 100 | 100 |
| | | | After | 100 | 100 | 100 | 100 |
| | | $D_r$ | Before | 100 | 100 | 100 | 99.5 |
| | | | After | 99 | 100 | 100 | 99 |
| | Test | | Before | 81.6 | 90.3 | 49.9 | 72.2 |
| | | | After | 82 | 93.2 | 57.5 | 77.6 |
| VGG19 | Train | $D_f$ | Before | 100 | 100 | 100 | 100 |
| | | | After | 100 | 100 | 100 | 100 |
| | | $D_r$ | Before | 100 | 100 | 100 | 99.6 |
| | | | After | 99 | 99 | 99 | 86.8 |
| | Test | | Before | 81.7 | 92.1 | 55.9 | 73.8 |
| | | | After | 82.2 | 91.8 | 53.4 | 71.8 |

on the FMNIST dataset. Conversely, the STL10 dataset exhibits the lowest test accuracy among the three datasets for CNN (51.7%), ResNet18 (54.4%), Xception (49.9%), and VGG (55.6%), showing its greater complexity in terms of classification challenges.

We further evaluate the performance of two membership approximation functions, *i.e.,* MF in white-box and black-box settings, on inferring whether a data sample belongs to the training datasets or not. The outcome of this experiment shed light on the suitability of MF in guiding ReMI unlearning process. Figure 2 demonstrates that MF achieves similar results in both black-box and white-box settings. We also included the MIA results (second row) as a baseline for evaluating the efficacy of MF model. Notably, the results suggests a similar performance between MF and MIA in inferring member samples. Moreover, the MF model (similar to MIA) achieves higher prediction accuracy on the STL10 dataset, which is partly due to the train and test accuracy gap in the target classification task (Figure 1). These observations align with prior findings [8], suggesting that membership fingerprinting

**TABLE IV:** Whitebox MIA accuracy of unlearned models before and after unlearning on the forgetting data.

| | Df | Phase | UTKFace | FMNIST | STL10 | CIFAR10 |
|---|---|---|---|---|---|---|
| CNN | 1% | Before | 1.00 | 1.00 | 0.975 | 1.00 |
| | | After | 0.052 | 0.318 | 0.15 | 0.019 |
| | 10% | Before | 0.981 | 1.00 | 0.975 | 1.00 |
| | | After | 0.106 | 0.365 | 0.166 | 0.013 |
| | 25% | Before | 0.962 | 1.00 | 0.979 | 1.00 |
| | | After | 0.129 | 0.485 | 0.106 | 0.026 |
| | 50% | Before | 0.941 | 1.00 | 0.982 | 0.999 |
| | | After | 0.19 | 0.588 | 0.101 | 0.044 |
| ResNet18 | 1% | Before | 1.00 | 1.00 | 1.00 | 1.00 |
| | | After | 0.052 | 0 | 0.1 | 0.006 |
| | 10% | Before | 1.00 | 0.997 | 1.00 | 1.00 |
| | | After | 0.001 | 0.002 | 0.009 | 0.001 |
| | 25% | Before | 1.00 | 0.997 | 1.00 | 0.998 |
| | | After | 0 | 0.002 | 0.004 | 0.0005 |
| | 50% | Before | 1.00 | 0.987 | 1.00 | 0.982 |
| | | After | 0 | 0.001 | 0.007 | 0.001 |
| Xception | 1% | Before | 1.00 | 0.977 | 1.00 | 1.00 |
| | | After | 0.21 | 0.005 | 0.47 | 0 |
| | 10% | Before | 1.00 | 0.879 | 1.00 | 1.00 |
| | | After | 0.21 | 0.002 | 0.314 | 0 |
| | 25% | Before | 0.929 | 0.883 | 1.00 | 0.999 |
| | | After | 0.156 | 0.002 | 0.067 | 0.0002 |
| | 50% | Before | 0.849 | 0.884 | 1.00 | 1.00 |
| | | After | 0.039 | 0.004 | 0.031 | 0 |
| VGG19 | 1% | Before | 1.00 | 0.631 | 0.825 | 1.00 |
| | | After | 0.456 | 0.111 | 0.6 | 0.24 |
| | 10% | Before | 1.00 | 0.837 | 0.835 | 1.00 |
| | | After | 0.599 | 0.267 | 0.528 | 0.501 |
| | 25% | Before | 1.00 | 0.857 | 0.828 | 1.00 |
| | | After | 0.334 | 0.211 | 0.604 | 0.228 |
| | 50% | Before | 0.997 | 0.877 | 0.81 | 1.00 |
| | | After | 0.765 | 0.208 | 0.619 | 0.135 |

(inference) accuracy tends to be higher when the target models experience overfitting.

### C. Unlearning Evaluation: Attack Resiliency

In this section, we will evaluate ReMI in terms of classification fidelity and unlearning resiliency to attacks. For the classification fidelity, we evaluate the difference in the classification accuracy of the target model before and after unlearning. Table III presents a comprehensive summary of the predictive accuracy of the training (*i.e.,* forgetting data samples ($D_f$) and the remaining training data ($D_r$)) and the test phases, both before and after unlearning. The results indicate that the unlearned model consistently maintains accuracy levels comparable to the original target model across all architectures and datasets. After the unlearning, we observe minimal changes in classification performance compared to the target model, with a maximum accuracy drop of less than 10% (CIFAR10 on VGG19). Interestingly, one can observe that in some cases (STL10 on ResNet18, Xception, and VGG19), ReMI has marginally improved the test accuracy of the unlearned models. This accuracy boost is due to the higher generalizability of the unlearned model as a result of using out-of-sample data during the unlearning process.

Table IV demonstrates the resiliency of our unlearning algorithm based on the white-box MIA success against forgetting data. Prior to unlearning, all forgetting data samples within $D_f$ received high probabilities from the MIA model, indicating that all data points in $D_f$ were likely used in training the target model. For instance, when considering forgetting data that

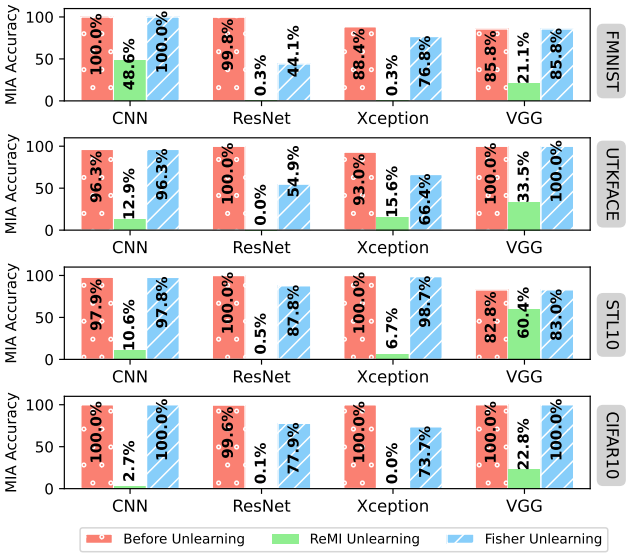


**Fig. 3:** MIA attack before unlearning on target model and after unlearning with Fisher and ReMI when the ratio of $D_f$ is 0.25

includes 50% of the training data samples, all target models from the four deep learning architectures initially exhibited nearly 100% accuracy in membership fingerprinting across all four datasets.

Applying ReMI, however, significantly reduces the white-box MIA accuracy on all four target models across all four datasets. Specifically, after applying the unlearning process to the target models trained on the CIFAR-10 dataset, the MIA accuracy on the forgetting data ($D_f|_{0.5}$) for each of the four deep learning methods is as follows: 4.4% (CNN), 0.1% (ResNet18), 0.0% (Xception), and 13.5% (VGG19). Similar unlearning performance is observed on the STL10, UTKFace, and FMNIST datasets. For example, on the FMNIST dataset, the attack accuracy decreases from the range 88%–98% for the original target models to a range of 3.9% for the Xception model and 0.0% for the ResNet18 model. On the STL10, the attack accuracy drops from the range 81%–100% to the range 3.1%–10% across all architectures except VGG19. The VGG19 did not perform as well as other architectures in unlearning the samples. We attribute this behavior to the lack of skip connections in the VGG architecture, which results in highly non-smooth loss surfaces [12].

We also compared MIA accuracy against ReMI and Fisher Unlearning across all datasets and architectures combinations (Figure 3 for forget ratio of 0.25). Before Unlearning, one can observe a high MIA accuracy, indicating significant privacy vulnerabilities across models. After unlearning, both ReMI and Fisher Unlearning exhibit lower MIA accuracies. Specifically, ReMI demonstrates significant normalized gains over Fisher unlearning in reducing MIA attack accuracy across all datasets, as measured by the *proportional effectiveness* (the percentage of ReMI reduction relative to the total reduction): STL10 exhibits the highest score at 95.76%, followed by CIFAR10 at 88.63%, UTKFace at 82.03%, and FMNIST at 81.91%. See the Appendix (Figures 7–9) for the complementary results of other forgetting ratios.

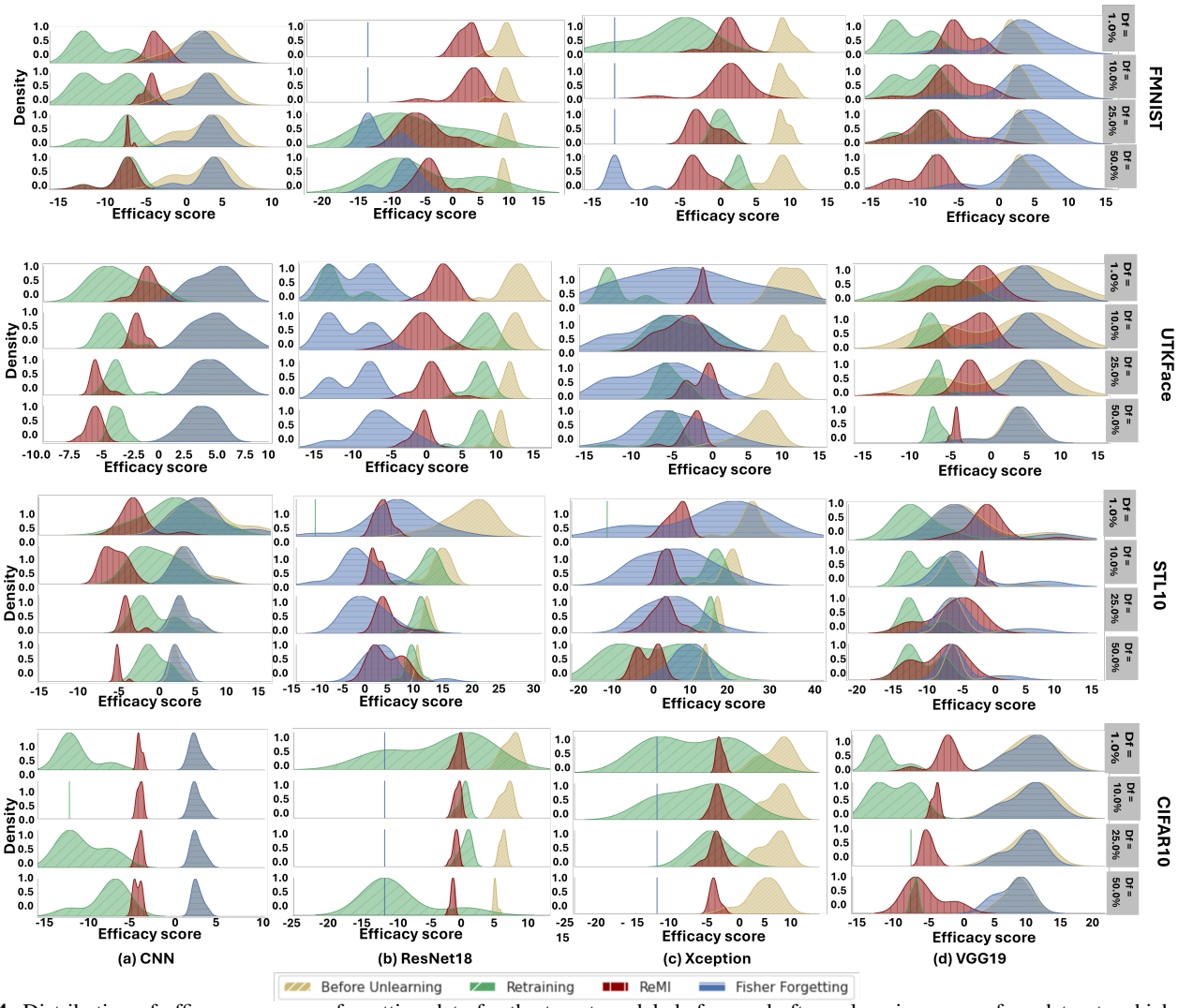


**Fig. 4:** Distribution of efficacy scores on forgetting data for the target models before and after unlearning across four datasets which shows how much information the model can leak. The results indicate that our unlearning method, ReMI, exposes less information compared to the original target model before unlearning. Additionally, the efficacy of the unlearning algorithms exhibits variations depending on the complexity of the deep learning architecture employed in the target model and the size of the forgetting dataset.

These results highlight the effectiveness of our unlearning algorithm in eliminating the private information of forgetting data from the target models without compromising the accuracy of the target classification models. The results of black-box MIAs on the unlearned data can be find in the Appendix (Figures 10–13).

### *D. Unlearning Evaluation: Leakage Analysis*

We also measured the unlearning efficacy score as another valuable metric for assessing the effectiveness of machine unlearning [10], [13]. The efficacy score quantifies the degree to which a model retains or discloses information by accounting principles from information theory and epistemic uncertainty [10]. A more robust unlearning algorithm is expected to lead to an unlearned model with a lower efficacy score, indicating higher uncertainty about the target data. We note that we calculate the efficacy score only for the forgetting data. We conducted unlearning experiments using the same model and data configurations with different random seeds to collect multiple efficacy scores for distribution analysis. This allows us to compare the efficacy score of ReMI against other unlearning algorithms.

We evaluate the effectiveness of ReMI by comparing its efficacy score with that of two alternative methods: Fisher Unlearning and Naive Retraining. The experiments for Fisher Unlearning and Naive Retraining were conducted under identical configurations as those used for our unlearning algorithm. These configurations included using four deep learning architectures and forgetting data of varying sizes (1.00%, 10.0%, 25.0%, and 50.0%) from four datasets. Figure 4 presents the distribution of efficacy scores obtained from the original target model and the unlearned models resulting from all three unlearning algorithms. The results indicate that in most cases, the three unlearning algorithms produced new models with lower efficacy scores than the original target model. This trend was consistent across various deep learning architectures regardless

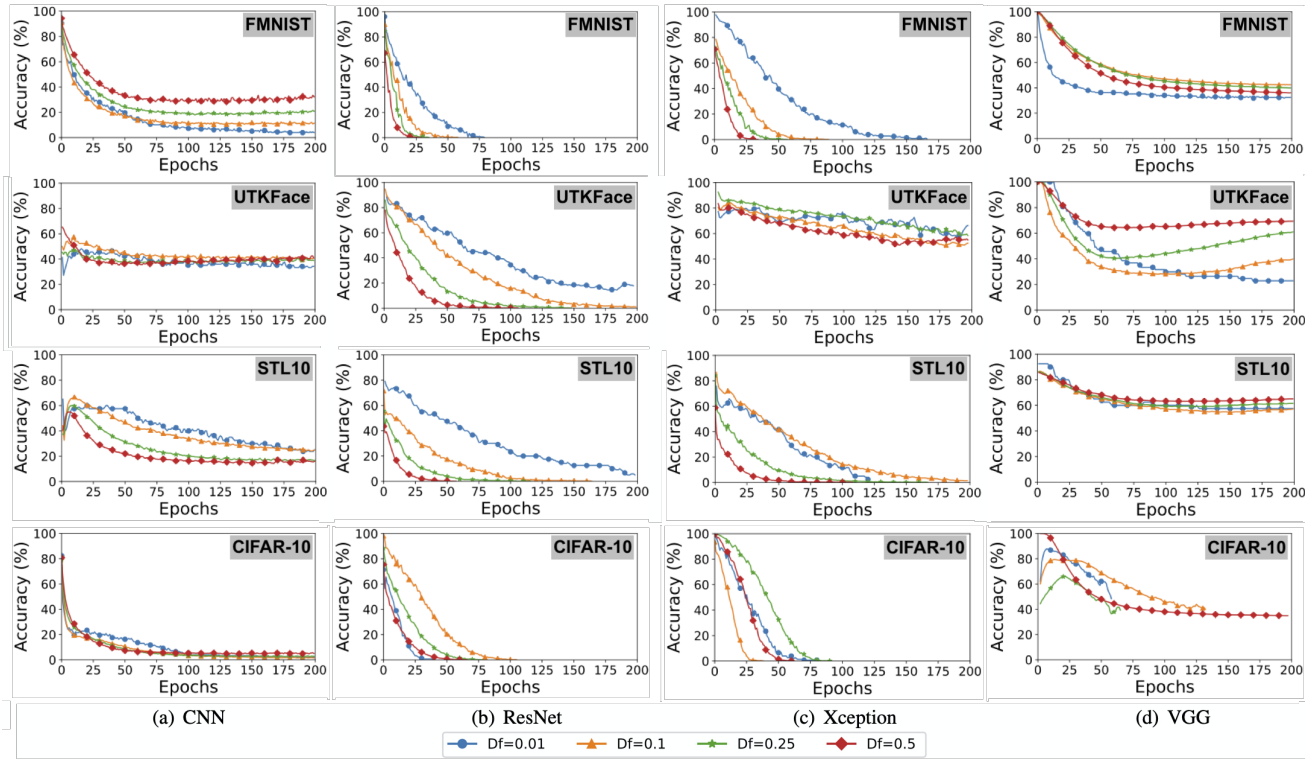


**Fig. 5:** Comparison of white-box MF accuracy during unlearning, highlighting the importance of the forgetting data size and the model's architecture on unlearning convergence. Selecting a smaller data to forget leads to faster convergence, particularly on CNN and VGG19.

of the size of the forgetting data. Notably, while the efficacy distribution of Fisher Unlearning sometimes overlapped with the efficacy distribution of the original model, as observed in cases like CNN on all four datasets and VGG19 on the FMNIST & CIFAR-10 datasets, the overall trend indicated a decrease in model efficacy.

Furthermore, the efficacy of the unlearning algorithms exhibited variations depending on the underlying target model architecture. For instance, both ReMI and Model Retraining generated new models with significantly lower efficacy scores across all forgetting datasets for a CNN-based target model. Our algorithm, in particular, demonstrated superior efficacy distributions for forgetting data sizes of 25% and 50%. However, Fisher unlearning outperformed the others by producing unlearning models with lower efficacy scores for larger forgetting datasets, specifically for the ResNet18-based target model. Nevertheless, ReMI algorithm still exhibited better unlearning efficacy than Model Retraining across various settings.

This analysis underscores the significant influence of both model complexity and forgetting dataset size on the performance of Model Retraining and Fisher Unlearning. For example, Model Retraining tends to yield less effective unlearning models as the model complexity increases, transitioning from simpler models like CNN to moderately complex models like ResNet. In contrast, Fisher Unlearning excels in generating better models with lower efficacy scores as the size of the forgetting dataset grows. Our unlearning algorithm, positioned between these two methods, is less affected by variations in model complexity and forgetting data sizes, offering a balanced performance across a range of scenarios.

### *E. ReMI Unlearning Process Insights*

Figure 5 provides insights into how the accuracy of the membership fingerprinting (MF) model on the forgetting dataset changes over various training epochs. The results consistently demonstrate a decrease in the MF accuracy during the unlearning process across different target models and forgetting data sizes. The speed of convergence depends on both the forgetting data size and the specific deep learning models used. For instance, ResNet and Xception models across all datasets exhibit faster convergence with bigger forgetting data sizes (25.0% and 50.0%), while smaller forgetting datasets (1.0% and 10.0%) require more epochs for convergence. A different pattern is observed with CNN and VGG19 models, where the unlearning process was longer despite the forgetting dataset.

Figure 6 provides an insightful representation of how the MF accuracy on the forgetting dataset changes across different training epochs when employing different types of MF (black-box and white-box) as privacy approximation functions ($\mathcal{G}(I(w))$) during the unlearning process when the forgetting data size is 25%. The other forgetting data size results can be found in the Appendix (Figures 14–16). The analysis illustrates that using the white-box MF model as a reference for minimizing the distribution of attack probabilities between in-sample and out-of-sample data proves to be more effective

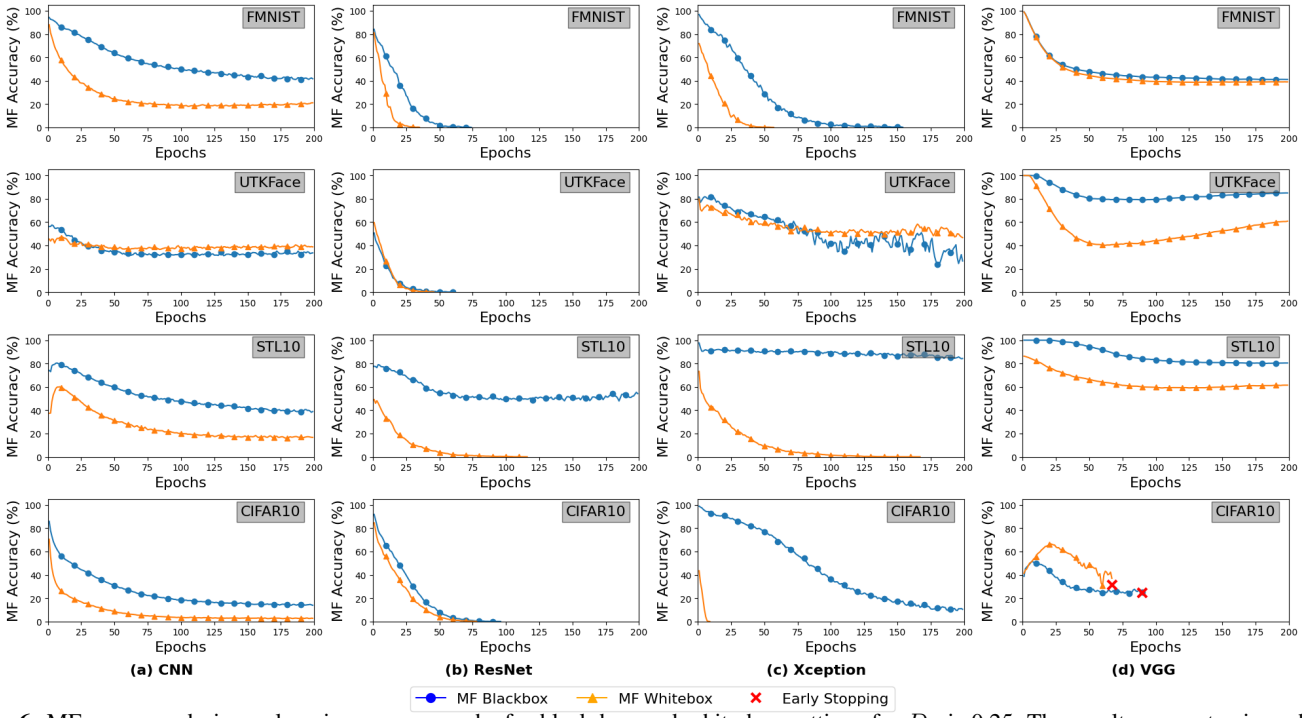


**Fig. 6:** MF accuracy during unlearning versus epochs for black-box and white-box settings for $D_f$ is 0.25. The results suggest using white-box MF to guide the unlearning process will have faster convergence, which consequently speeds up the unlearning process.

than relying on the black-box MF model. This approach results in a faster convergence of the unlearning loss and lower data leakage for the forgetting data when compared to the utilization of the black-box model. This observation can be partially attributed to the set of white-box features, which include gradients derived from the target model, contributing to enhanced unlearning performance. On the other hand, employing the black-box MF model to guide the unlearning process demonstrates less success, particularly evident when applying CNN and ResNet18 on STL10 data. This analysis underscores the need for more effective privacy approximation functions to enhance unlearning performance.

### *F. Unlearning Latency*

Finally, we conducted an evaluation of the unlearning latency of our algorithm in comparison to Model Retraining. Table V presents the results of our latency analysis based on the experiments we conducted. The evaluation involves measuring the running time of the unlearning process, which begins with the loading of a pre-trained target model and the specified forgetting data and ends when the unlearning process is completed. As demonstrated in Table V, the ReMI unlearning algorithm exhibits an average completion time of approximately 268 seconds across all architectures for the UTKFace dataset. This is nearly four times faster than Model Retraining, which requires approximately 933 seconds on average. Notably, the most time-consuming step in the ReMI unlearning process is the calculation of attack probabilities for loss optimization ($\mathcal{L}_{\mathcal{G}}(w)$), which consumes approximately 196 seconds on average. This analysis underscores the lower latency of our unlearning algorithm. We did not compare with Fisher unlearning as it has shown to be an order of magnitude slower than retraining [14].

**TABLE V:** Unlearning latency comparison of Retraining and ReMI Unlearning on UTKFace dataset. Time is measured in seconds. ReMI Unlearning includes total unlearning time and time for attack Loss calculation. The results include ReMI's speed up over retraining.

| | $D_f$ Ratio | Retraining | ReMI Unlearning | | Speed up |
|---|---|---|---|---|---|
| | | | Unlearning | $\mathcal{L}_{\mathcal{G}}(w)$ calculation | |
| CNN | 0.01 | 812.321 | 194.523 | 99.99 | 4.1x |
| | 0.1 | 850.325 | 270.449 | 215.573 | 3.1x |
| | 0.25 | 864.031 | 68.613 | 57.570 | 12.5x |
| | 0.5 | 808.766 | 204.579 | 176.096 | 3.9x |
| ResNet | 0.01 | 1063.228 | 11.463 | 7.896 | 92.76x |
| | 0.1 | 1034.444 | 12.555 | 9.127 | 82.3x |
| | 0.25 | 950.442 | 163.488 | 123.163 | 5.81x |
| | 0.5 | 943.834 | 168.435 | 130.271 | 5.6x |
| Xception | 0.01 | 1060.723 | 119.905 | 85.543 | 8.84x |
| | 0.1 | 787.714 | 312.653 | 231.046 | 2.51x |
| | 0.25 | 995.286 | 1123.721 | 849.638 | 0.88x |
| | 0.5 | 930.395 | 468.84 | 358.547 | 1.98x |
| VGG19 | 0.01 | 1044.088 | 153.981 | 106.279 | 6.78x |
| | 0.1 | 1051.894 | 100.856 | 70.592 | 10.42x |
| | 0.25 | 912.485 | 375.815 | 269.564 | 2.42x |
| | 0.5 | 833.932 | 540.613 | 351.665 | 1.54x |
| **Avg. Time** | | **933.994** | **268.155** | **196.410** | **3.48x** |

### *G. Cross-Method Unlearning Efficacy Analysis*

We further assess the impact of our algorithm on private information exposure under different membership fingerprinting scenarios. Specifically, we evaluate the accuracy of a black-box MF against models unlearned using a white-box MF as a reference, and vice versa. This bidirectional evaluation aims to uncover insights into the effectiveness of our algorithm

**TABLE VI:** Cross-method Unlearning Efficacy Evaluation

| $D_f$ | Unlearning Source | Evaluation | Attack Accuracy (%) | |
|---|---|---|---|---|
| | | | Before | After |
| 0.01 | White-box | Black-box | 100 | 40 |
| | Black-box | White-box | 92.5 | 87.5 |
| 0.1 | White-box | Black-box | 100 | 62.5 |
| | Black-box | White-box | 92.5 | 70 |
| 0.25 | White-box | Black-box | 100 | 42.5 |
| | Black-box | White-box | 92.5 | 100 |
| 0.5 | White-box | Black-box | 100 | 40 |
| | Black-box | White-box | 92.5 | 92.5 |

in mitigating privacy risks and understanding how different fingerprinting strategies interact with one another. Table VI presents a summary of the results from our comprehensive bidirectional membership fingerprinting evaluations across various forgetting dataset sizes. The successful unlearning process, guided by a reference MF model, should effectively mitigate privacy risks across different mechanisms, indicating the removal of private information associated with the training data. As presented in Table VI, when the target model undergoes unlearning by incorporating white-box MF probabilities into the loss function optimization, the resulting new model exhibits significantly reduced attack accuracy when subjected to independent black-box fingerprinting. This observation holds true across unlearned models trained with forgetting data of varying sizes. For instance, the fingerprinting accuracy of the black-box MF diminishes from 100% to 40%, 62.5%, 42.5%, and 40% when applied to forgetting datasets with sizes of 1%, 10%, 25%, and 50%, respectively, after the unlearning is applied based on white-box MF attack. In contrast, the target models that have undergone unlearning using a black-box MF model remain vulnerable to white-box MF.

## REFERENCES


[1] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[2] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5810–5818.

[3] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 215–223.

[4] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[6] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[8] Y. Liu, R. Wen, X. He, A. Salem, Z. Zhang, M. Backes, E. De Cristofaro, M. Fritz, and Y. Zhang, "{ML-Doctor}: Holistic risk assessment of inference attacks against machine learning models," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 4525–4542.

[9] A. Thudi, G. Deza, V. Chandrasekaran, and N. Papernot, "Unrolling sgd: Understanding factors influencing machine unlearning," in *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2022, pp. 303–319.

[10] A. Becker and T. Liebig, "Evaluating machine unlearning via epistemic uncertainty," *arXiv preprint arXiv:2208.10836*, 2022.

[11] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig *et al.*, "Adversarial robustness toolbox v1. 0.0," *arXiv preprint arXiv:1807.01069*, 2018.

[12] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," *Advances in neural information processing systems*, vol. 31, 2018.

[13] T. T. Nguyen, T. T. Huynh, P. L. Nguyen, A. W.-C. Liew, H. Yin, and Q. V. H. Nguyen, "A survey of machine unlearning," *arXiv preprint arXiv:2209.02299*, 2022.

[14] J. Foster, S. Schoepf, and A. Brintrup, "Fast machine unlearning without retraining through selective synaptic dampening," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 11, 2024, pp. 12 043–12 051.


## APPENDIX

## SUPPLEMENTARY RESULTS

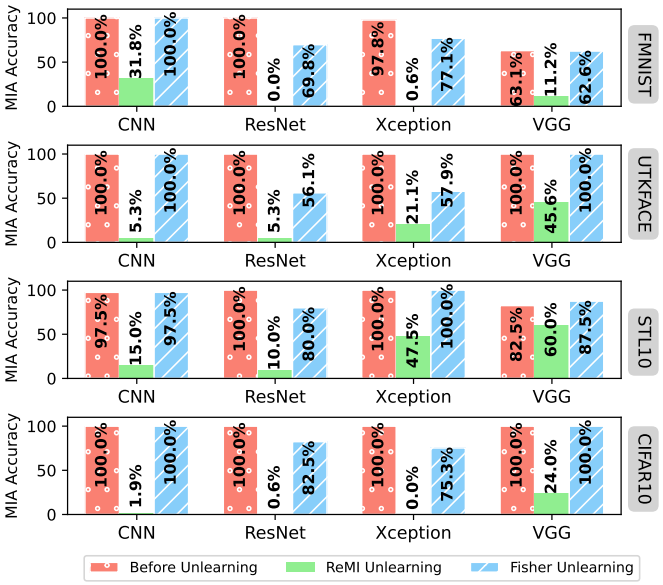


**Fig. 9:** MIA attack before unlearning on target model and after unlearning with Fisher and ReMI when the ratio of $D_f$ is 0.01

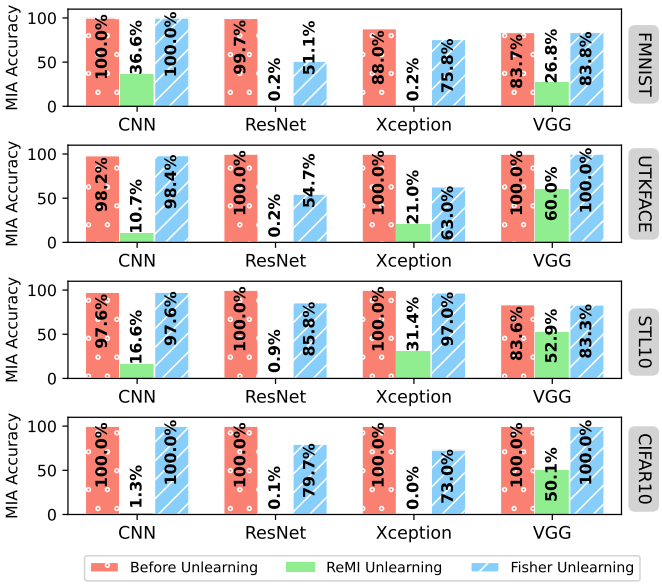


**Fig. 8:** MIA attack before unlearning on target model and after unlearning with Fisher and ReMI when the ratio of $D_f$ is 0.1

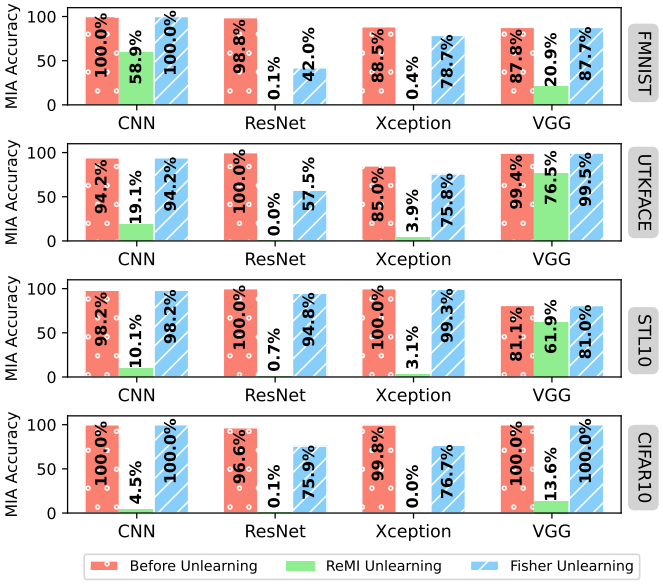


**Fig. 7:** MIA attack before unlearning on target model and after unlearning with Fisher and ReMI when the ratio of $D_f$ is 0.5

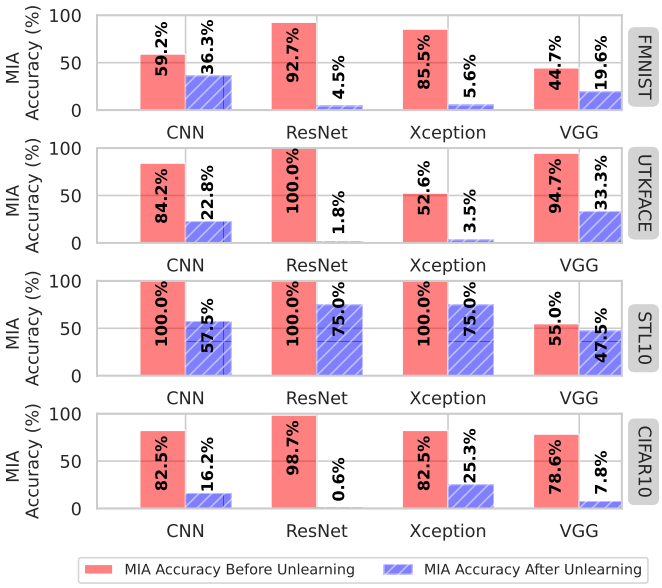


**Fig. 10:** Blackbox MIA accuracy on the forget dataset, a subset of the target model's training dataset, before and after unlearning when the ratio of $D_f$ is 0.01. It is evident that ReMI unlearning effectively reduces the information leakage of the forgetting data.

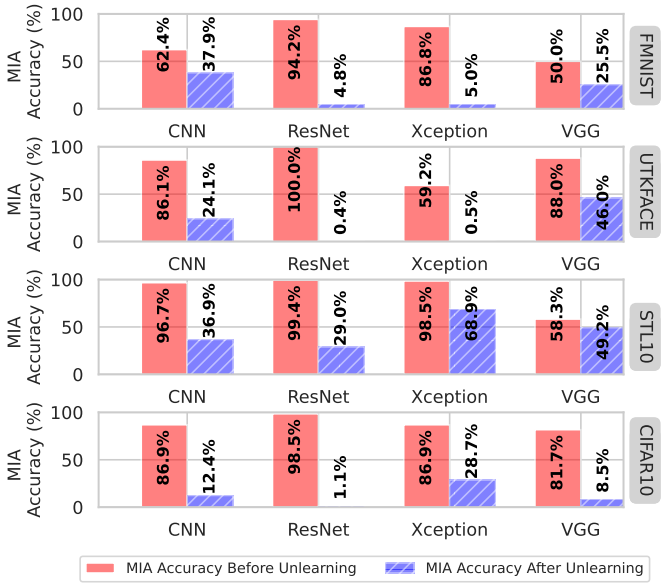


**Fig. 11:** Blackbox MIA accuracy on the forget dataset, a subset of the target model's training dataset, before and after unlearning when the ratio of $D_f$ is 0.1. It is evident that ReMI unlearning effectively reduces the information leakage of the forgetting data.

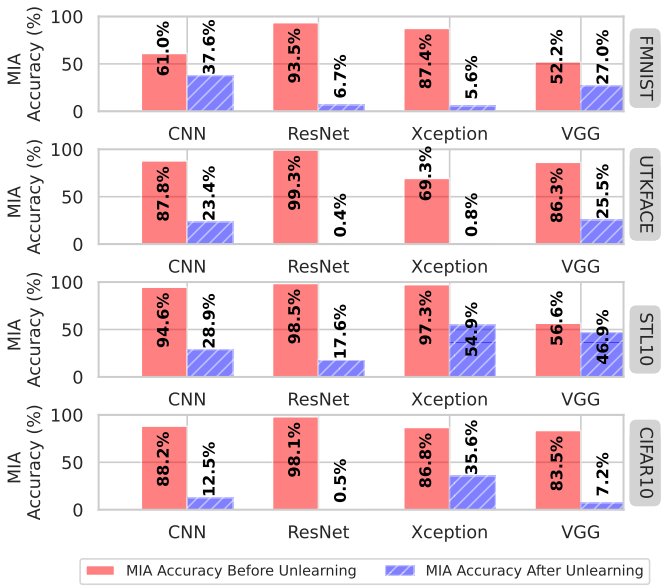


**Fig. 12:** Blackbox MIA accuracy on the forget dataset, a subset of the target model's training dataset, before and after unlearning when the ratio of $D_f$ is 0.25. It is evident that ReMI unlearning effectively reduces the information leakage of the forgetting data.

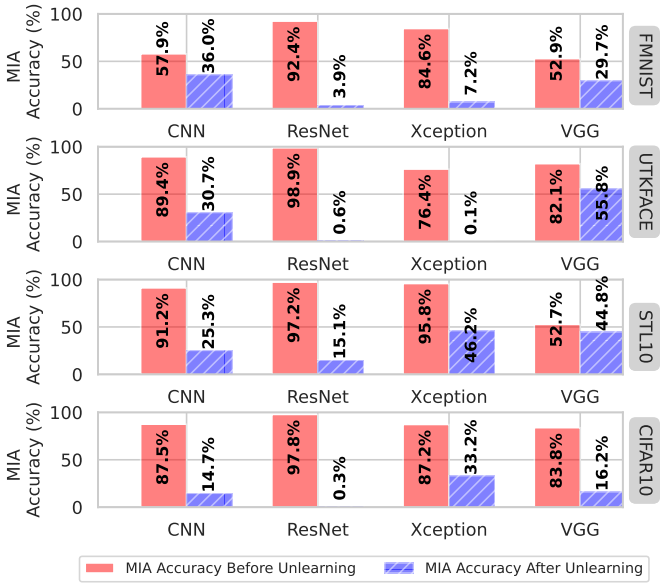


**Fig. 13:** Blackbox MIA accuracy on the forget dataset, a subset of the target model's training dataset, before and after unlearning when the ratio of $D_f$ is 0.5. It is evident that ReMI unlearning effectively reduces the information leakage of the forgetting data.

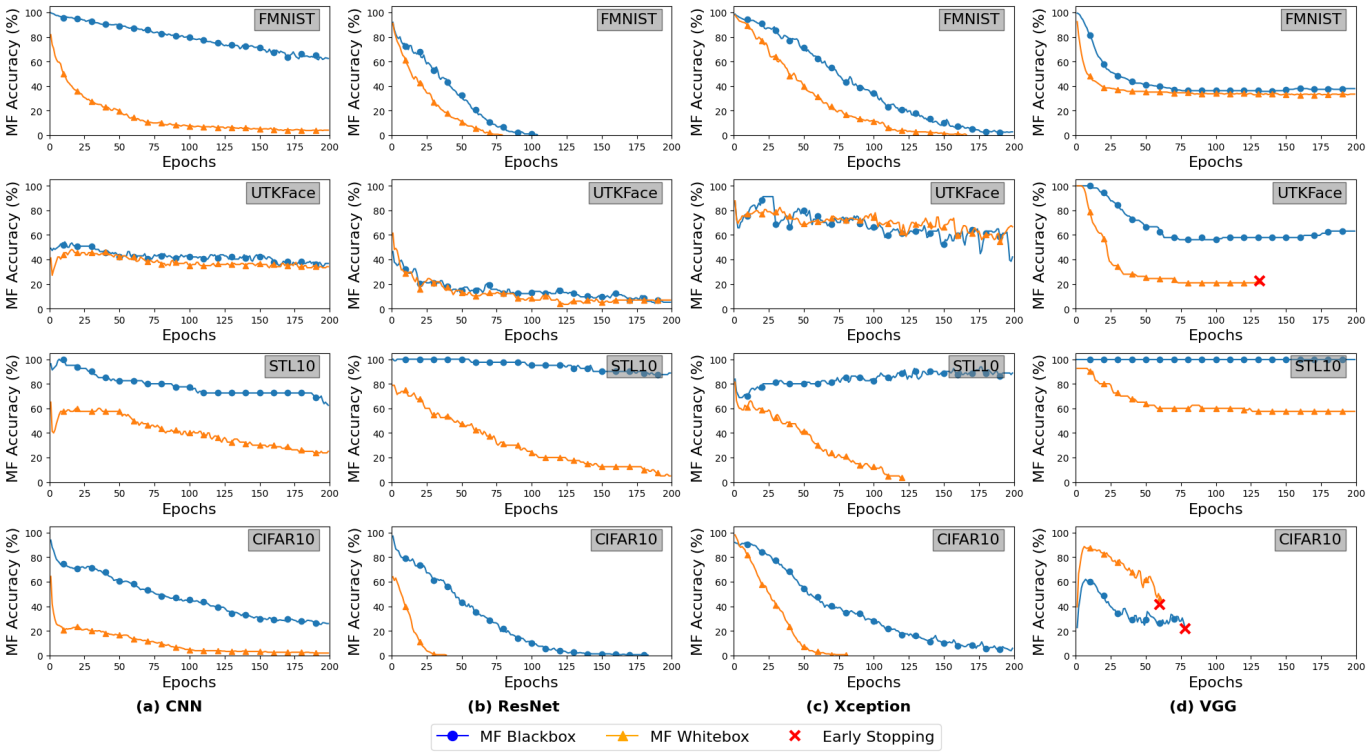


**Fig. 14:** MF accuracy during unlearning versus epochs for black-box and white-box settings for $D_f$ is 0.01. The results suggest using white-box MF to guide the unlearning process will have faster convergence, which consequently speeds up the unlearning process.

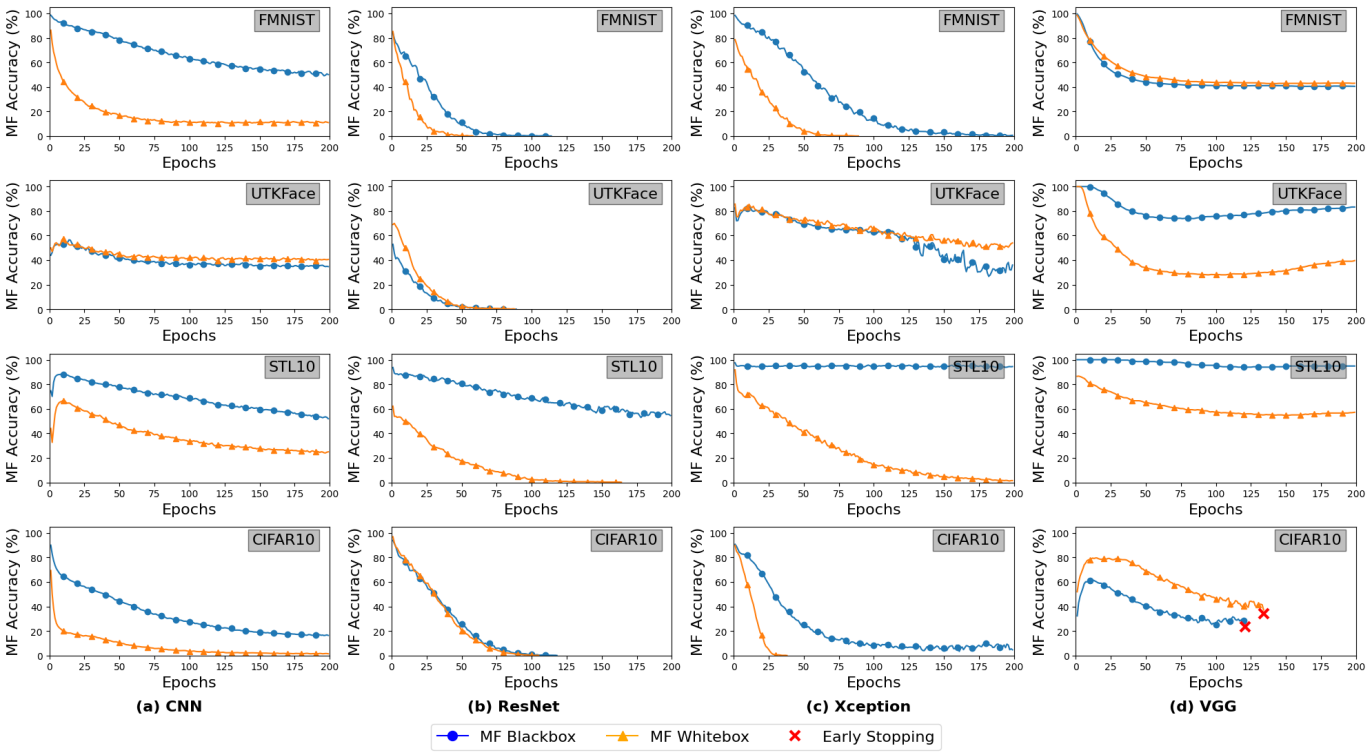


**Fig. 15:** MF accuracy during unlearning versus epochs for black-box and white-box settings for $D_f$ is 0.1. The results suggest using white-box MF to guide the unlearning process will have faster convergence, which consequently speeds up the unlearning process.

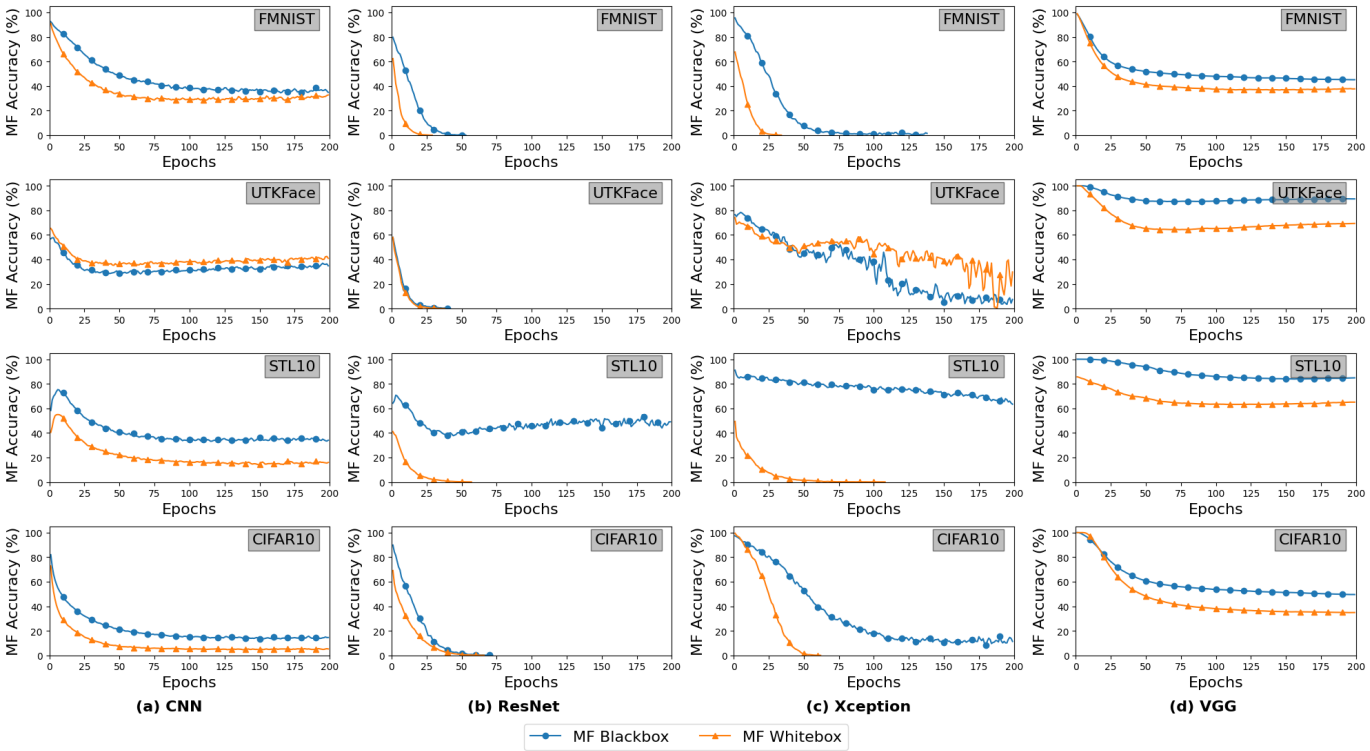


**Fig. 16:** MF accuracy during unlearning versus epochs for black-box and white-box settings for $D_f$ is 0.5. The results suggest using white-box MF to guide the unlearning process will have faster convergence, which consequently speeds up the unlearning process.