Appendix

In this Appendix, Appendix A provides experiment details and deferred experiment figures and tables; Appendix B provides the deferred proofs and the complete algorithm description. In addition, Appendix B also describes how to generalize *MUter* from successive *single datapoint* unlearning to successive *batch of datapoints* removal. Our source code can be found in Supplementary Material.

A. Experiment Details and Deferred Experiment Figures and Tables

A.1. Experiment Details

Our experiments are based on the Pytorch platform and run on RTX 3090. For the pretraining the Wide ResNet model on Downsampled ImageNet, we run on four GPU devices; For all other experiments, we run on a single GPU device.

We summarize the hyperparameters used for our adversarial training/finetuning in Table 3. For Neural Network Model, we conduct experiments on Wide ResNet 28-10 model using the datasets Lacuna-10 and CIFAR-10. We first perform adversarial pretraining on Lacuna-100 and Downsampled ImageNet datasets, both by SGD with momentum for the outer-loops. For the former, we fix the learning rate to 0.1, and then train for 80 epochs with momentum 0.9 and weight decay 0.0005. For the latter, we use the pretrained model in [29] as the pretrained model here, which has a similar training process. Then, we freeze all but the last layer to adversarially finetune the model on Lacuna-10 and CIFAR-10 datasets.

We summarize the hyperparameters required by *MUter* in Table 4. In the neural network model experiments, [32] points out that under the non-convex setting, the Hessian matrix on parameters ω^* sometimes will not be positive definite. We follow [32] to add a damping term λ on the diagonal with $\lambda = 0.0001$.

Dataset	Model type	Learning rate	(Tuning) Epoch	FGSM	PGD		
				epsilon	epsilon	alpha	steps
MNIST-b	Logistic	0.01	100	0.25	0.25	0.03	15
	Ridge	0.01	100	0.25	0.25	0.03	15
Covtype	Logistic	0.1	15	4/255	4/255	0.004	7
	Ridge	0.1	15	4/255	4/255	0.004	7
Lacuna-10	Neural Network	0.01	20	8/255	8/255	2/255	10
CIFAR-10	Neural Network	0.001	10	8/255	8/255	2/255	10

Table 3: Adversarial training/finetuning parameters.

A.2. Datasets

Linear Model. We perform experiments on MNIST-b and Covtype. MNIST-b is taken from the MNIST dataset, which consists of $28 * 28$ grayscale images from digist '0' to digit '9'. We select the digit '1' and digit '7' to form the binary subset MNIST-b. Covtype with 54 attributes is used to classify the main tree species in the Roosevelt National Forest wilderness area, where use the binary classification version from LIBSVM.

Dataset	Model type	Neumann Series order k	Conjugate Gradient iterations C
MNIST-b	Logistic	3	10
	Ridge	3	10
Covtype	Logistic	3	20
	Ridge	20	20
Lacuna-10	Neural Network	100	10
CIFAR-10	Neural Network	100	20

Table 4: *MUter* parameters.

Neural Network Model. We introduce Lacuna-100 and downsampled ImageNet as core datasets, and conduct experiments on target datasets Lacuna-10 and CIFAR-10. CIFAR-10 consists of $32 * 32$ color pictures, covering different animals and machines in 10 categories. The Downsampled ImageNet is derived from the 1000-class ImageNet dataset, which resize the image to $32 * 32$. Lacuna-10/Lacuna-100 comes from [22]. We use the same data processing method to select 10/100 celebrities (no intersection) from VGGFace2 [7], and each celebrity randomly selects at least 500 pictures. Then each celebrity divides 100 pictures to form the test set, and the remaining images to form the training set. Finally, we resize the images to $32 * 32$.

To facilitate the constrained adversarial perturbation for adversarial training, all the above datasets are scaled to $[0, 1]$ (for image data, we use Totensor to transform, and for Covtype, we choose the version scaled to $[0, 1]$ in LIBSVM.). We summarize the dimensions, classes, and quantity information of the above datasets in Table 5.

Dataset	Domension	Classes	Train data	Test Data
MNIST-b	784	2	11,982	1,198
Covtype	54	2	522,910	58,102
Lacuna-10	3072	10	4,374	1,000
CIFAR-10	3072	10	50,000	10,000

Table 5: Datasets Statistics

A.3. Deferred Experiment Figures and Tables

In this section, we report additional experiment results, where Subsection A.3.1 reports additional experiment results under the PGD setting, and Subsection A.3.2 reports additional experiment results under the FGSM setting.

A.3.1 Deferred Experiment Figures under PGD Setting

We summarize the experiment results under Ridge Model with PGD in Figure 6.

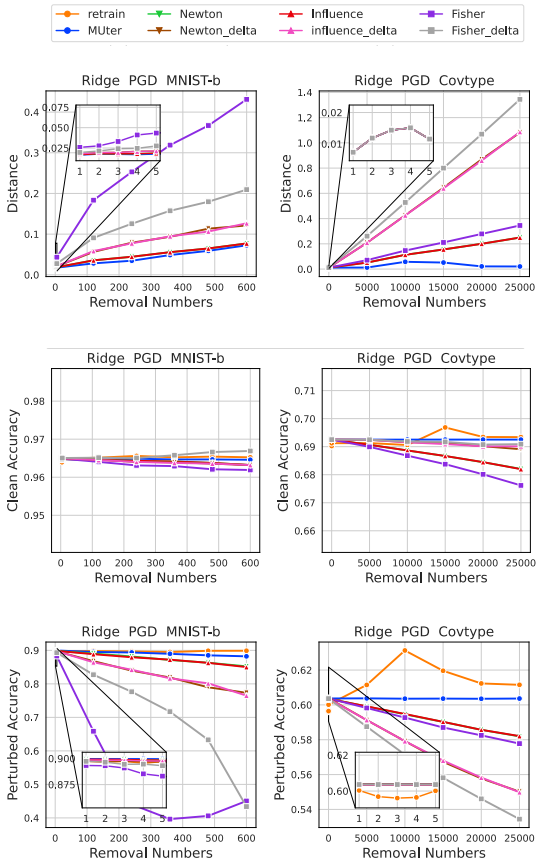

Figure 6: Evaluation results on Ridge Regression Model with PGD: Effectiveness (left column), Accuracy (middle column), and Robustness (right column) on datasets MNIST (top) and Covtype (bottom). Large plots have greater removal numbers: $1\%, 2\%, \cdots, 5\%$; Small plots inside the large plots have fewer removal numbers: $1, 2, \cdots, 5$.

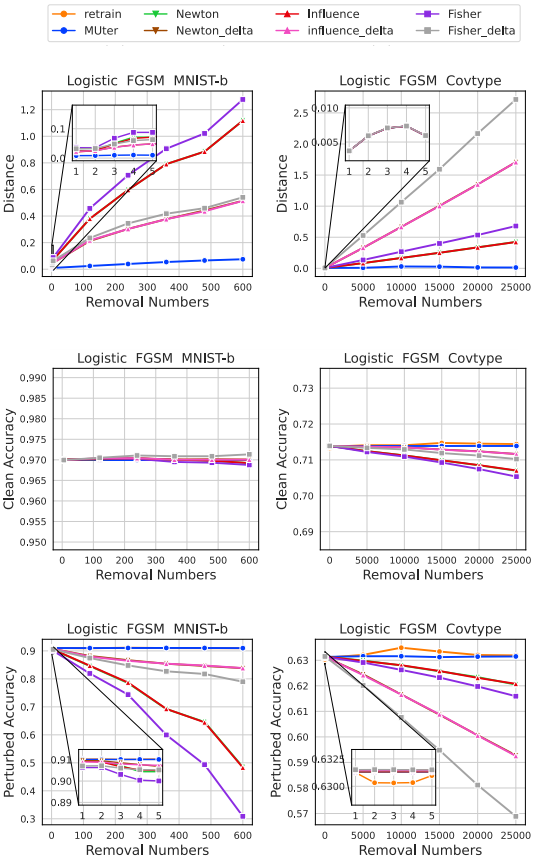

Figure 7: Evaluation results on Logistic Regression Model with FGSM: Effectiveness (left column), Accuracy (middle column), and Robustness (right column) on datasets MNIST (top) and Covtype (bottom). Large plots have greater removal numbers: $1\%, 2\%, \cdots, 5\%$; Small plots inside the large plots have fewer removal numbers: $1, 2, \cdots, 5$.

A.3.2 Deferred Experiment Figures and Tables under FGSM Setting

In this part, we change the way of generating perturbations from PGD to FGSM. With the same experimental settings under PGD conditions, we report experiment results both on Linear Model and Neural Network Model.

Results with Linear Model. We report the effectiveness, accuracy and robustness metrics of logistic model and ridge model in Figure 7 and Figure 8, respectively. In Table 6, we summarize the efficiency comparison for linear model under FGSM setting.

Model Type	Removal Number	MNIST-b				Covtype			
		Fisher	F-delta	*MUter*	Retrain	Fisher	F-delta	*MUter*	Retrain
LR	1	0.002	0.002	0.008	14.6	0.002	0.002	0.007	75
	5	0.010	0.012	0.045	14.5	0.008	0.009	0.033	76
	10	0.020	0.022	0.089	14.2	0.018	0.019	0.065	77
	1%	0.242	0.249	0.978	14.3	9.178	11.278	33.974	77
RR	1	0.002	0.002	0.008	14.6	0.002	0.002	0.007	78
	5	0.009	0.012	0.047	14.5	0.009	0.009	0.034	78
	10	0.018	0.020	0.091	14.3	0.018	0.021	0.063	79
	1%	0.239	0.244	0.993	14.5	9.797	11.043	33.505	79

Table 6: Efficiency results with Logistic Regression Model (top) and Ridge Regression Model (bottom) under FGSM: The unlearning time (in seconds) of **Fisher**, **Fisher-delta**, *MUter* and **Retrain** under varying removal numbers: 1, 5, 10, 1% (120 for MNSIT-b, 5000 for Covtype).

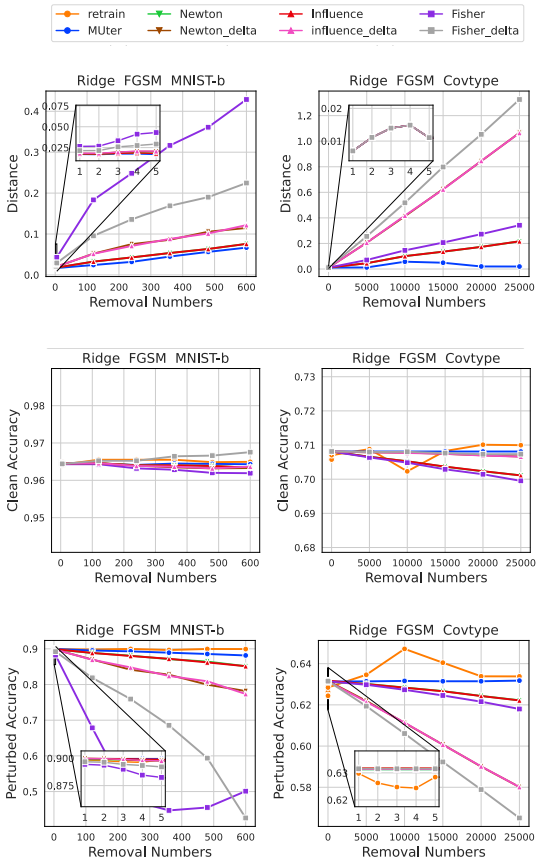

Figure 8: Evaluation results on Ridge Regression Model with FGSM: Effectiveness (left column), Accuracy (middle column), and Robustness (right column) on datasets MNIST (top) and Covtype (bottom). Large plots have greater removal numbers: $1\%, 2\%, \cdots, 5\%$; Small plots inside the large plots have fewer removal numbers: $1, 2, \cdots, 5$.

Result with Neural Network Model. We report the effectiveness, accuracy, and robustness metrics of the neural network model in Figure 9. In Table 7, we summarize the efficiency comparison for the neural network model under the FGSM setting.

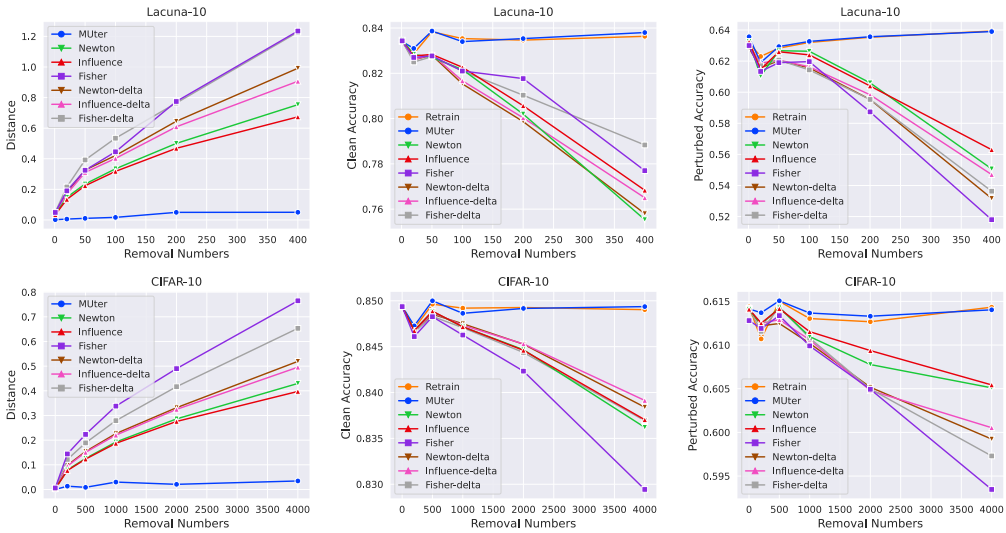

Figure 9: Evaluation results on Neural Network with FGSM: Effectiveness (left column), Accuracy (middle column), and Robustness (right column) on datasets Lacuna-10 (top) and CIFAR-10 (bottom), under removal numbers $(1, 0.4\%, 1\%, 2\%, 4\%, 8\%)$.

Removal	**Lacuna-10**				**CIFAR-10**			
Number	Fisher	F-delta	*MUter*	Retrain	Fisher	F-delta	*MUter*	Retrain
1	1.51	1.57	3.96	150	1.43	1.56	3.72	799
5	7.81	8.08	20.72	151	7.71	8.11	18.23	794
10	15.46	15.94	39.82	150	15.42	15.87	36.84	790

Table 7: Efficiency results with Neural Network Model under FGSM: The unlearning time (in seconds) of **Fisher**, **Fisher-delta**, *MUter* and **Retrain** under varying removal numbers: 1, 5, 10.

B. Deferred Proofs and Algorithm Description

In this section, Subsection B.1 provides the omitted proofs for the derivation of the ATM unlearning update (i.e., Lemma 1 and Theorem 1), Subsection B.2 provides the omitted proof for the derivation of the successive unlearning setting (i.e., Corollary 1), Subsection B.3 provides the deferred Lemma for Schur complement for completeness and the omitted proof for Theorem 2, Subsection B.4 provides the complete algorithm description for *MUter* , and Subsection B.5 how to generalize *MUter* from successive *single datapoint* unlearning to successive *batch of datapoints* removal.

B.1. Proofs for Derivation of ATM Unlearning Update

B.1.1 Proof of Lemma 1

Proof. By Taylor expansion around both $\boldsymbol{\omega}^*$ and $\delta_i(\boldsymbol{\omega}^*)$, we have

$$\begin{aligned}
&\frac{1}{n-1}\sum_{i=1,i\neq i^\dagger}^{n}\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^u, \mathbf{x}_i+\delta_i(\boldsymbol{\omega}^u))\\
=&\frac{1}{n-1}\sum_{i=1,i\neq i^\dagger}^{n}\Big[\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))\\
&+\partial_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u-\boldsymbol{\omega}^*)+O(\|\boldsymbol{\omega}^u-\boldsymbol{\omega}^*\|_2^2)\\
&+\partial_{\boldsymbol{\omega}\boldsymbol{\delta}} l(\boldsymbol{\omega}^*, \mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))(\delta_i(\boldsymbol{\omega}^u)-\delta_i(\boldsymbol{\omega}^*))+O(\|\delta_i(\boldsymbol{\omega}^u)-\delta_i(\boldsymbol{\omega}^*)\|_2^2)\Big].
\end{aligned}\tag{17}$$

By furthering neglecting the higher-order Taylor expansion terms $O(\|\boldsymbol{\omega}^u-\boldsymbol{\omega}^*\|_2^2)$ and $O(\|\delta_i(\boldsymbol{\omega}^u)-\delta_i(\boldsymbol{\omega}^*)\|_2^2)$, we have the following relationship,

$$\begin{aligned}
&\frac{1}{n-1}\sum_{i=1,i\neq i^\dagger}^{n}\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^u, \mathbf{x}_i+\delta_i(\boldsymbol{\omega}^u))\\
\approx&\frac{1}{n-1}\sum_{i=1,i\neq i^\dagger}^{n}\Big[\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))\\
&+\partial_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u-\boldsymbol{\omega}^*)\\
&+\partial_{\boldsymbol{\omega}\boldsymbol{\delta}} l(\boldsymbol{\omega}^*, \mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))(\delta_i(\boldsymbol{\omega}^u)-\delta_i(\boldsymbol{\omega}^*))\Big],
\end{aligned}\tag{18}$$

which proves Lemma 1. □

B.1.2 Proof of Theorem 1

Proof. By Lemma 1, we begin with

$$\begin{aligned}
&\frac{1}{n-1}\sum_{i=1,i\neq i^\dagger}^{n}\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^u, \mathbf{x}_i+\delta_i(\boldsymbol{\omega}^u))\\
\approx&\frac{1}{n-1}\sum_{i=1,i\neq i^\dagger}^{n}\Big[\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))\\
&+\partial_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u-\boldsymbol{\omega}^*)+\partial_{\boldsymbol{\omega}\boldsymbol{\delta}} l(\boldsymbol{\omega}^*, \mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))(\delta_i(\boldsymbol{\omega}^u)-\delta_i(\boldsymbol{\omega}^*))\Big].
\end{aligned}\tag{19}$$

For the first line on the right hand side of eq.(19), we have

$$\begin{aligned}
&\frac{1}{n-1}\sum_{i=1,i\neq i^\dagger}^{n}\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))\\
=&\frac{1}{n-1}\sum_{i=1}^{n}\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))-\frac{1}{n-1}\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_{i^\dagger}+\delta_{i^\dagger}(\boldsymbol{\omega}^*))\\
=&0-\frac{1}{n-1}\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_{i^\dagger}+\delta_{i^\dagger}(\boldsymbol{\omega}^*)\\
=&-\frac{1}{n-1}\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_{i^\dagger}+\delta_{i^\dagger}(\boldsymbol{\omega}^*)),
\end{aligned}\tag{20}$$

where the second equality is obtained as follows. Since $\boldsymbol{\omega}^*$ is the optimum of the following ATM,

$$\boldsymbol{\omega}^*=\operatorname*{argmin}_{\boldsymbol{\omega}}\max_{\boldsymbol{\delta}_i\in\mathcal{B}(\mathbf{x}_i,r)}\frac{1}{n}\sum_{i=1}^{n} l(\boldsymbol{\omega},\mathbf{x}_i+\delta(\boldsymbol{\omega}^*)),\tag{21}$$

it satisfies the following by Danskin's Theorem,

$$\frac{1}{n}\sum_{i=1}^{n}\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))=0,\tag{22}$$

which is the second equality relation in eq.(20).

For the second line on the right hand side of eq.(19), we further expand $(\delta_i(\boldsymbol{\omega}^u)-\delta_i(\boldsymbol{\omega}^*))$ by the implicit function theorem. That is, for all $i\in\{1,\cdots,n\}$, we have

$$\delta_i(\boldsymbol{\omega}^u)-\delta_i(\boldsymbol{\omega}^*)=-\partial_{\boldsymbol{\delta}\boldsymbol{\delta}}^{-1} l(\boldsymbol{\omega}^*,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))\cdot\partial_{\boldsymbol{\delta}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))\cdot(\boldsymbol{\omega}^u-\boldsymbol{\omega}^*),\tag{23}$$

which gives

$$\begin{aligned}
&\frac{1}{n-1}\sum_{i=1,i\neq i^\dagger}^{n}\Big[\partial_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u-\boldsymbol{\omega}^*)+\partial_{\boldsymbol{\omega}\boldsymbol{\delta}} l(\boldsymbol{\omega}^*,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))(\delta_i(\boldsymbol{\omega}^u)-\delta_i(\boldsymbol{\omega}^*))\Big]\\
=&\frac{1}{n-1}\sum_{i=1,i\neq i^\dagger}^{n}\Big[\partial_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u-\boldsymbol{\omega}^*)\\
&-\partial_{\boldsymbol{\omega}\boldsymbol{\delta}} l(\boldsymbol{\omega}^*,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))\partial_{\boldsymbol{\delta}\boldsymbol{\delta}}^{-1} l(\boldsymbol{\omega}^*,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))\partial_{\boldsymbol{\delta}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u-\boldsymbol{\omega}^*)\Big]\\
=&\frac{1}{n-1}\sum_{i=1,i\neq i^\dagger}^{n}\mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u-\boldsymbol{\omega}^*).
\end{aligned}\tag{24}$$

By substituting eq.(20) and eq.(24) into eq.(19), we have

$$\begin{aligned}
&\frac{1}{n-1}\sum_{i=1,i\neq i^\dagger}^{n}\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^u,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^u))\\
\approx&-\frac{1}{n-1}\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_{i^\dagger}+\delta_{i^\dagger}(\boldsymbol{\omega}^*))+\frac{1}{n-1}\sum_{i=1,i\neq i^\dagger}^{n}\mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u-\boldsymbol{\omega}^*).
\end{aligned}\tag{25}$$

Thus, $\boldsymbol{\omega}^u=\boldsymbol{\omega}^*+\mathfrak{U}(\boldsymbol{\omega}^*,i^\dagger)=\boldsymbol{\omega}^*+\Big[\sum_{i=1,i\neq i^\dagger}^{n}\mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))\Big]^{-1}\cdot\Big[\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_{i^\dagger}+\delta_{i^\dagger}(\boldsymbol{\omega}^*))\Big]$ is the the solution to the following linear system,

$$-\frac{1}{n-1}\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_{i^\dagger}+\delta_{i^\dagger}(\boldsymbol{\omega}^*))+\frac{1}{n-1}\sum_{i=1,i\neq i^\dagger}^{n}\mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u-\boldsymbol{\omega}^*)=0,\tag{26}$$

which gives $\frac{1}{n-1}\sum_{i=1,i\neq i^\dagger}^{n}\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^u,\mathbf{x}_i+\delta_i(\boldsymbol{\omega}^u))\approx 0$ according to eq.(25). As a result, we have proved that $\boldsymbol{\omega}^u$ satisfies the approximate unlearning criteria for ATM. □

B.2. Proofs for derivation of Successive Unlearning Setting

B.2.1 Proof of Corollary 1

Proof. Denote the data that have been deleted by $\mathcal{U}_r := \{i_1^\dagger, i_2^\dagger, \ldots, i_r^\dagger\}$. Similar to the ATM unlearning standard in eq.(3), we have the retraining-from-scratch model parameter after the $r+1$-th unlearning by $\boldsymbol{\omega}^*_{-\mathcal{U}_r \cup i^\dagger}$, which has the following definition,

$$\boldsymbol{\omega}^*_{-\mathcal{U}_r \cup i^\dagger} = \underset{\boldsymbol{\omega}}{\operatorname{argmin}} \frac{1}{n-r-1} \sum_{i=1, i \neq \mathcal{U}_r \cup i^\dagger}^{n} \max_{\boldsymbol{\delta}_i \in \mathcal{B}(\mathbf{x}_i, r)} l(\boldsymbol{\omega}, \mathbf{x}_i + \boldsymbol{\delta}_i). \tag{27}$$

The unlearning criteria for ATM at the $r+1$-th unlearning is as follows,

$$\sum_{i=1, i \neq \mathcal{U}_r \cup i^\dagger}^{n} \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*_{-\mathcal{U}_r \cup i^\dagger}, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*_{-\mathcal{U}_r \cup i^\dagger})) \approx 0. \tag{28}$$

Our aim is to show that $\boldsymbol{\omega}^u_{r+1}$ approximately satisfies the above criteria,

$$\sum_{i=1, i \neq \mathcal{U}_r \cup i^\dagger}^{n} \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^u_{r+1}, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^u_{r+1})) \approx 0, \tag{29}$$

which is also the approximate unlearning criteria for ATM in eq.(6).

The proof is similar to Theorem 1, as follows,

$$\begin{aligned}
&\sum_{i=1, i \neq \mathcal{U}_r \cup i^\dagger}^{n} \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^u_{r+1}, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^u_{r+1})) \\
=&\Big[\sum_{i=1}^{n} \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^u_{r+1}, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^u_{r+1}))\Big] - \Big[\sum_{i=i_1^\dagger}^{i_r^\dagger} \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^u_{r+1}, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^u_{r+1}))\Big] - \Big[\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^u_{r+1}, \mathbf{x}_{i^\dagger} + \delta_{i^\dagger}(\boldsymbol{\omega}^u_{r+1}))\Big] \\
\approx&\Big[\Big(\sum_{i=1}^{n} \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*)) + \partial_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u_{r+1} - \boldsymbol{\omega}^*) + \partial_{\boldsymbol{\omega}\boldsymbol{\delta}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*))(\delta_i(\boldsymbol{\omega}^u_{r+1}) - \delta_i(\boldsymbol{\omega}^*))\Big)\Big] \\
&- \Big[\Big(\sum_{i=i_1^\dagger}^{i_r^\dagger} \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*)) + \partial_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u_{r+1} - \boldsymbol{\omega}^*) + \partial_{\boldsymbol{\omega}\boldsymbol{\delta}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*))(\delta_i(\boldsymbol{\omega}^u_{r+1}) - \delta_i(\boldsymbol{\omega}^*))\Big)\Big] \\
&- \Big[\Big(\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_{i^\dagger} + \delta_{i^\dagger}(\boldsymbol{\omega}^*)) + \partial_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_{i^\dagger} + \delta_{i^\dagger}(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u - \boldsymbol{\omega}^*) + \partial_{\boldsymbol{\omega}\boldsymbol{\delta}} l(\boldsymbol{\omega}^*, \mathbf{x}_{i^\dagger} + \delta_{i^\dagger}(\boldsymbol{\omega}^*))(\delta_{i^\dagger}(\boldsymbol{\omega}^u_{r+1}) - \delta_{i^\dagger}(\boldsymbol{\omega}^*))\Big)\Big] \\
=&\Big[\sum_{i=1}^{n} \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*)) + \mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u_{r+1} - \boldsymbol{\omega}^*)\Big] \\
&- \Big[\sum_{i=i_1^\dagger}^{i_r^\dagger} \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*)) + \mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u_{r+1} - \boldsymbol{\omega}^*)\Big] \\
&- \Big[\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_{i^\dagger} + \delta_{i^\dagger}(\boldsymbol{\omega}^*)) + \mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_{i^\dagger} + \delta_{i^\dagger}(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u_{r+1} - \boldsymbol{\omega}^*)\Big] \\
=&\Big[\sum_{i=1}^{n} \mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u_{r+1} - \boldsymbol{\omega}^*)\Big] \\
&- \Big[\sum_{i=i_1^\dagger}^{i_r^\dagger} \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*)) + \mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u_{r+1} - \boldsymbol{\omega}^*)\Big] \\
&- \Big[\nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_{i^\dagger} + \delta_{i^\dagger}(\boldsymbol{\omega}^*)) + \mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_{i^\dagger} + \delta_{i^\dagger}(\boldsymbol{\omega}^*))(\boldsymbol{\omega}^u_{r+1} - \boldsymbol{\omega}^*)\Big],
\end{aligned} \tag{30}$$

where the approximation equality is by the Taylor expansion and neglecting the higher-order terms, the second equality is by implicit function theorem and the definition of the total Hessian, and the third equality is by $\sum_{i=1}^{n} \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*)) = 0$ by eq.(22).

Thus, $\boldsymbol{\omega}_{r+1}^u = \boldsymbol{\omega}^* + \boldsymbol{\mathcal{U}}_r(\boldsymbol{\omega}^*, i^\dagger)$ is the the solution to the following linear system,

$$\begin{aligned} &- \sum_{i=i_1^\dagger}^{i_r^\dagger} \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*)) - \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_{i^\dagger} + \delta_{i^\dagger}(\boldsymbol{\omega}^*)) \\ &+ \Big[\sum_{i=1}^{n} \mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*)) - \sum_{i=i_1^\dagger}^{i_r^\dagger} \mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*)) - \mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_{i^\dagger} + \delta_{i^\dagger}(\boldsymbol{\omega}^*)) \Big] (\boldsymbol{\omega}^u - \boldsymbol{\omega}^*) = 0, \end{aligned} \tag{31}$$

which gives $\sum_{i=1, i \neq \mathcal{U}_r \cup i^\dagger}^{n} \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}_{r+1}^u, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}_{r+1}^u)) \approx 0$. As a result, we have proved that $\boldsymbol{\omega}^u$ satisfies the approximate unlearning criteria for ATM in eq.(29) that is consistent with eq.(6). □

B.3. Proof for Schur Complement Conversion

B.3.1 Additional Lemma of Schur Complement

Lemma 2. *(Schur Complement Conversion) Let* $\mathbf{S} = \mathbf{H}_{11} - \mathbf{H}_{12}\mathbf{H}_{22}^{-1}\mathbf{H}_{21}$. *If* $\mathbf{H}_{22}$ *and* $\mathbf{S}$ *are invertible, then* $\mathbf{H}$ *is invertible and the following relation holds,*

$$\mathbf{H}^{-1} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{S}^{-1} & -\mathbf{S}^{-1}\mathbf{H}_{12}\mathbf{H}_{22}^{-1} \\ -\mathbf{H}_{22}\mathbf{H}_{21}\mathbf{S}^{-1} & \mathbf{H}_{22}^{-1}\mathbf{H}_{21}\mathbf{S}^{-1}\mathbf{H}_{12}\mathbf{H}_{22}^{-1} \end{bmatrix}. \tag{32}$$

B.3.2 Proof of Theorem 2

Proof. Let

$$\mathbf{S} = \Big[\mathcal{M}_r[\mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}}] - \partial_{\boldsymbol{\omega}\boldsymbol{\omega}} l_{i^\dagger}^* \Big] - \Big[- \partial_{\boldsymbol{\omega}\boldsymbol{\delta}} \partial_{\boldsymbol{\delta}\boldsymbol{\delta}}^{-1} \partial_{\boldsymbol{\delta}\boldsymbol{\omega}} l_{i^\dagger}^* \Big], \tag{33}$$

which is the Schur complement of the following block matrix,

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} = \begin{bmatrix} \mathcal{M}_r[\mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}}] - \partial_{\boldsymbol{\omega}\boldsymbol{\omega}} l_{i^\dagger}^* & \partial_{\boldsymbol{\omega}\boldsymbol{\delta}} l_{i^\dagger}^* \\ \partial_{\boldsymbol{\delta}\boldsymbol{\omega}} l_{i^\dagger}^* & -\partial_{\boldsymbol{\delta}\boldsymbol{\delta}} l_{i^\dagger}^* \end{bmatrix}. \tag{34}$$

Then, based on Lemma 2, we have

$$\mathbf{S}^{-1}\mathbf{g} = \Delta\boldsymbol{\omega} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix}^{-1} \cdot \begin{bmatrix} \mathbf{g} \\ \mathbf{0} \end{bmatrix}, \tag{35}$$

which can be cast as the solution of the following linear system,

$$\begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \cdot \begin{bmatrix} \Delta\boldsymbol{\omega} \\ \Delta\boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \mathbf{0} \end{bmatrix}. \tag{36}$$

By substituting eq.(34) in, we have

$$\begin{bmatrix} \mathcal{M}_r[\mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}}] - \partial_{\boldsymbol{\omega}\boldsymbol{\omega}} l_{i^\dagger}^* & \partial_{\boldsymbol{\omega}\boldsymbol{\delta}} l_{i^\dagger}^* \\ \partial_{\boldsymbol{\delta}\boldsymbol{\omega}} l_{i^\dagger}^* & -\partial_{\boldsymbol{\delta}\boldsymbol{\delta}} l_{i^\dagger}^* \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\omega} \\ \Delta\boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathcal{M}_r[\nabla_{\boldsymbol{\omega}}] + \nabla_{\boldsymbol{\omega}} l_{i^\dagger}^* \\ \mathbf{0} \end{bmatrix}, \tag{37}$$

which proves Theorem 2. □

B.4. The Complete Algorithm Description

We present the complete algorithm description for *MUter* in Algorithm 1.

Algorithm 1 The Complete Algorithm Description for *MUter*

Input: Training dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, loss l, adversarial perturbation constraint $\mathcal{B}(\mathbf{x}_i, r)$, adversarial training finishing status $\{\boldsymbol{\omega}^*, \delta_1(\boldsymbol{\omega}^*), \cdots, \delta_n(\boldsymbol{\omega}^*)\}$, a sequence of indices to be removed $\mathcal{U} = \{i_1^\dagger, i_2^\dagger, \ldots, \}$

Stage I. Pre-Unlearning:

Compute memory $\mathcal{M}_0[\mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}}] = \sum_{i=1}^n \widetilde{\mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \delta_i(\boldsymbol{\omega}^*))$ by eq.(16);

Initialize $\mathcal{M}_0[\nabla_{\boldsymbol{\omega}}] = \mathbf{0}$ and $\mathcal{M}_0[\boldsymbol{\omega}^*] = \boldsymbol{\omega}^*$;

for $i_{r+1}^\dagger = i^\dagger$ **do**

  Stage II. Unlearning:

  Re-compute the adversarial perturbation $\delta_{i^\dagger}(\boldsymbol{\omega}^*)$;

  Compute gradient $\mathbf{g} = \nabla_{\boldsymbol{\omega}} l_{i^\dagger}^*$;

  Apply conjugate gradient (with C iterations) to solve the least square problem of the linear system

$$\begin{bmatrix} \mathcal{M}_r[\mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}}] - \partial_{\boldsymbol{\omega}\boldsymbol{\omega}} l_{i^\dagger}^* & \partial_{\boldsymbol{\omega}\boldsymbol{\delta}} l_{i^\dagger}^* \\ \partial_{\boldsymbol{\delta}\boldsymbol{\omega}} l_{i^\dagger}^* & -\partial_{\boldsymbol{\delta}\boldsymbol{\delta}} l_{i^\dagger}^* \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\omega} \\ \Delta\boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} \mathcal{M}_r[\nabla_{\boldsymbol{\omega}}] + \nabla_{\boldsymbol{\omega}} l_{i^\dagger}^* \\ \mathbf{0} \end{bmatrix},$$

  Obtain $\mathfrak{U}_r(\boldsymbol{\omega}^*, i_{r+1}^\dagger) = \Delta\boldsymbol{\omega}$, the model paraeter after the $r+1$-th unlearning: $\boldsymbol{\omega}_{r+1}^u = \boldsymbol{\omega}^* + \mathfrak{U}_r(\boldsymbol{\omega}^*, i_{r+1}^\dagger)$;

  Stage III. Post-Unlearning:

  $\mathbf{m}_{i^\dagger} = \partial_{\boldsymbol{\omega}\boldsymbol{\omega}} l_{i^\dagger}^* - \partial_{\boldsymbol{\omega}\boldsymbol{\delta}} l_{i^\dagger}^* \big[\sum_{j=0}^k (\mathbf{I} - \partial_{\boldsymbol{\delta}\boldsymbol{\delta}} l_{i^\dagger}^*)^j\big] \partial_{\boldsymbol{\delta}\boldsymbol{\omega}} l_{i^\dagger}^*$ by eq.(16);

  Update the memory $\mathcal{M}_{r+1}[\mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}}] = \mathcal{M}_r[\mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}}] - \mathbf{m}_{i^\dagger}$ and $\mathcal{M}_{r+1}[\nabla_{\boldsymbol{\omega}}] = \mathcal{M}_r[\nabla_{\boldsymbol{\omega}}] + \mathbf{g}$;

end for

B.5. Extension to Successive Batch Unlearning

In this subsection, we show that our method can be generalized to the successive batch unlearning setting.

Successive Batch Unlearning Setting. Denote the index sets of datapoints that have already been forgotten at timestamp r by $\{\mathcal{U}_1^\dagger, \cdots, \mathcal{U}_r^\dagger\}$, where each $\mathcal{U}_r^\dagger$ contains a set of data indices. Let the set of datapoints to be forgotten at timestamp $r+1$ by $\mathcal{U}_{r+1}^\dagger = \mathcal{U}^\dagger$. Corollary 2 below extends Corollary 1 to support successive batch unlearning for ATM.

Corollary 2. *Considering the successive batch unlearning setting, let the machine unlearning update at the $r+1$-th timestamp $\mathfrak{U}_r(\boldsymbol{\omega}^*, \mathcal{U}^\dagger)$ take the following form*

$$\begin{aligned} \mathfrak{U}_r(\boldsymbol{\omega}^*, i_{r+1}^\dagger) := \Bigg\{ & \overbrace{\Big[\sum_{i=1}^n \mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \boldsymbol{\delta}_i^*)\Big]}^{\mathcal{M}_0[\mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}}]} - \\ & \overbrace{\Big[\sum_{j=1}^r \sum_{i \in \mathcal{U}_j^\dagger} \mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \boldsymbol{\delta}_i^*)\Big]}^{\text{Part of } \mathcal{M}_r[\mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}}]} - \Big[\sum_{i^\dagger \in \mathcal{U}^\dagger} \mathtt{D}_{\boldsymbol{\omega}\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_{i^\dagger} + \boldsymbol{\delta}_{i^\dagger}^*)\Big] \Bigg\}^{-1} \\ & \cdot \overbrace{\Big[\sum_{j=1}^r \sum_{i \in \mathcal{U}_j^\dagger} \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_i + \boldsymbol{\delta}_i^*) + \sum_{i^\dagger \in \mathcal{U}^\dagger} \nabla_{\boldsymbol{\omega}} l(\boldsymbol{\omega}^*, \mathbf{x}_{i^\dagger} + \boldsymbol{\delta}_{i^\dagger}^*)\Big]}^{\mathcal{M}_r[\nabla_{\boldsymbol{\omega}}]}. \end{aligned} \tag{38}$$

Then, the unlearning model with updated parameters $\boldsymbol{\omega}_{r+1}^u := \boldsymbol{\omega}^* + \mathfrak{U}_r(\boldsymbol{\omega}^*, \mathcal{U}^\dagger)$ satisfies the approximate unlearning criteria for the adversarial training model in eq.(6).

The proof of Corollary 2 is similar to the proof of Corollary 1. Based on Corollary 2, *MUter* can be similarly designed to support the successive batch unlearning setting.