

Learning the Unlearnable: Adversarial Augmentations Suppress Unlearnable Example Attacks

Tianrui Qin^{* a,b}, Xitong Gao^{* a}, Juanjuan Zhao^a, Kejiang Ye^a, Cheng-Zhong Xu^c
^a Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, China.
^b University of Chinese Academy of Sciences, China.
^c University of Macau, Macau S.A.R., China.

{tr.qin, xt.gao, jj.zhao, kj.ye}@siat.ac.cn czxu@um.edu.mo

Abstract

Unlearnable example attacks are data poisoning techniques that can be used to safeguard public data against unauthorized training of deep learning models. These methods add stealthy perturbations to the original image, thereby making it difficult for deep learning models to learn from these training data effectively. Current research suggests that adversarial training can, to a certain degree, mitigate the impact of unlearnable example attacks, while common data augmentation methods are not effective against such poisons. Adversarial training, however, demands considerable computational resources and can result in non-trivial accuracy loss. In this paper, we introduce the UEraser method variants, which outperforms current defenses against different types of state-of-the-art unlearnable example attacks through a combination of effective data augmentation policies and loss-maximizing adversarial augmentations. In stark contrast to the current SOTA adversarial training methods, UEraser uses adversarial augmentations, which extends beyond the confines of ℓ_p perturbation budget assumed by current unlearning attacks and defenses. UEraser suppresses the unlearning effect with error-maximizing data augmentations, thus restoring trained model accuracies. Interestingly, UEraser-Lite, a fast variant without adversarial augmentations, is also highly effective in preserving clean accuracies. On various unlearnable example attacks, it achieves results that are comparable to those obtained during clean training. We also demonstrate the efficacy of UEraser against possible adaptive attacks. Our code is open source¹ and available to the deep learning community.

1. Introduction

Deep learning has achieved great success in fields such as computer vision [10] and natural language processing [4],

and the development of various fields now relies on large-scale datasets. While these datasets have undoubtedly contributed significantly to the progress of deep learning, the collection of unauthorized private data for training these models now presents an emerging concern. Recently, numerous poisoning methods [7, 11, 17, 19, 21] have been proposed to add imperceptible perturbations to images. These perturbations can form “shortcuts” [8, 11] in the training data to prevent training and thus make the data unlearnable in order to preserve privacy. It is commonly perceived that the only effective defense against unlearnable examples are adversarial training algorithms [7, 11, 19]. Popular data augmentation methods such as CutOut [5], MixUp [25], and AutoAugment [3], however, have all been demonstrated to be ineffective defenses.

Current methods of unlearnable attacks involves the specification of an ℓ_p perturbation budget, where $p \in \{2, \infty\}$ in general. Essentially, they constrain the added perturbation to a small ϵ -ball of ℓ_p -distance from the source image, in order to ensure stealthiness of these attacks. Adversarial training defenses [7, 14] represent a defense mechanism that seeks to counteract the bounded perturbations from such unlearnable attacks. However, large defensive perturbations comes with significant accuracy degradations. This prompts the inquiry of the existence of effective defense mechanisms that leverage threat models that are outside the purview of attackers. Specifically, *can we devise effective adversarial policies for training models that extend beyond the confines of the ℓ_p perturbation budgets?*

In this paper, we thus propose *UEraser*, which performs error-maximizing data augmentation, to defense against unlearning poisons. *UEraser* challenges the preconception that data augmentation is not an effective defense against unlearning poisons. *UEraser* expands the perturbation distance far beyond traditional adversarial training, as data augmentation policies do not confine themselves to the ℓ_p perturbation constraints. It can therefore effectively disrupt “unlearning

^{*}Equal contribution. Correspondence to Xitong Gao.

¹<https://github.com/lafeat/ueraser>.

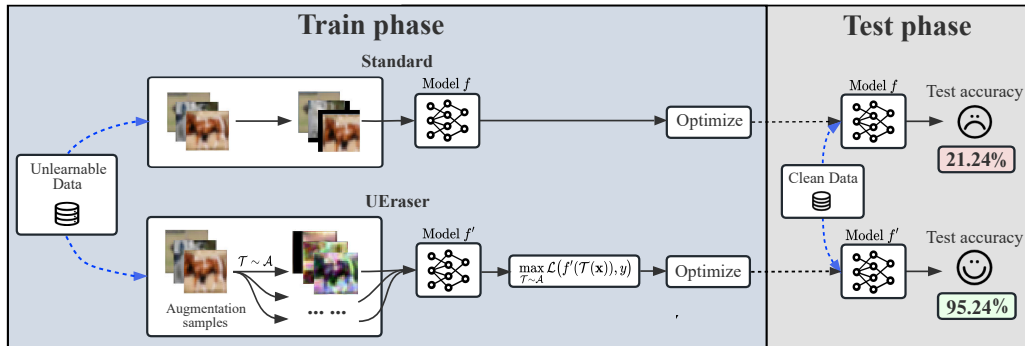


Figure 1: A high-level overview of *UEraser* for countering unlearning poisons. Note that *UEraser* recovers the clean accuracy of unlearnable examples by data augmentations. The reported results are for EM [11] attack.

shortcuts” formed by attacks within narrow ℓ_p constraints. Yet, the augmentations employed by *UEraser* are natural and realistic transformations extensively utilized by existing works to improve the models’ ability to generalize. This, in turn, helps in avoiding accuracy loss due to perturbations used by adversarial training that could potentially be out-of-distribution. Finally, traditional adversarial training is not effective in mitigating unlearning poisons produced by adaptive attacks [7], while *UEraser* is highly resilient against adaptive attacks with significantly lower accuracy reduction.

In summary, our main contributions are as follows:

- It extends adversarial training beyond the confines of the ℓ_p perturbation budgets commonly imposed by attackers into data augmentation policies.
- We propose *UEraser*, which introduces an effective adversarial augmentation to wipe out unlearning perturbations. It defends against the unlearnable attacks by maximizing the error of the augmented samples.
- *UEraser* is highly effective in suppressing the unlearning effect on state-of-the-art (SOTA) unlearning attacks, outperforming existing SOTA defense methods. It lays a fresh foundation for future competitions among unlearnable example attack and defense strategies.

Unlearnable example attacks bear great significance, not just from the standpoint of privacy preservation, but also as a form of data poisoning attack. It is thus of great significance to highlight the shortcomings of current attack methods. Perhaps most surprisingly, even a well-known unlearnable attack such as EM [11] is unable to impede the effectiveness of *UEraser*. By training a ResNet-18 model from scratch using exclusively CIFAR-10 unlearnable data produced with EM (with an ℓ_∞ budget of $8/255$), *UEraser* achieves exceptional accuracy of 95.24% on the clean test set, which closely matches the accuracy achievable by standard training on a clean training set. This suggests that existing unlearning perturbations are tragically inadequate

in making data unlearnable, even with adaptive attacks that employs *UEraser*. By understanding their weaknesses, we can anticipate how malicious actors may attempt to exploit them, and prepare stronger safeguards against such threats. We hope *UEraser* can help facilitate the advancement of research in these attacks and defenses.

2. Related Work

Adversarial examples and adversarial training. Adversarial examples deceive machine learning models by adding adversarial perturbations, often imperceptible to human, to source images, leading to incorrect classification results [9, 18]. White-box adversarial attacks [18] maximize the loss of a source image with gradient descent on the defending model to add adversarial perturbations onto an image to maximize its loss on the model. Effective methods to gain adversarial robustness usually involve adversarial training [14], which leverages adversarial examples to train models. Adversarial training algorithms thus solve the min-max problem of minimizing the loss function for most adversarial examples within a perturbation budget, typically bounded in ℓ_p . Recent years have thus observed an arms race between adversarial attack strategies and defense mechanisms [1, 2, 22, 23].

Unlearnable examples. Unlearnable examples attacks are a type of data poisoning methods with bounded perturbation that aims to make learning from such examples difficult. Unlike conventional data poisoning methods, unlearnable examples methods usually require adding imperceptible perturbations to all examples [7, 11, 17, 19, 21]. NTGA [24] simulates the training dynamics of a generalized deep neural network using a Gaussian process and leverages this surrogate to find better local optima with improved transferability. Error-minimizing (EM) [11] poison generates imperceptible perturbations with a min-min objective, minimizing the errors of training examples on a trained model. Unlike EM, Adversarial Poisoning (TAP) [6] considers the adversarial sample generation and uses the error maximization

process to generate adversarial samples. Hypocritical Perturbations (HYPO) [19] instead uses a pretrained surrogate rather than the above min-min optimization. As the above method cannot defend against adversarial training, Robust Error-Minimization (REM) [7] uses an adversarially-trained model as an adaptively attack to generate stronger unlearnable examples. Yu *et al.* [21] generate linearly separable perturbations (LSP) for unlearnable examples. Autoregressive poisoning (AR) [17] prescribes perturbation that can generalize to different datasets and architectures. One pixel shortcut (OPS) [20] is a targeted availability poisoning attack that perturbs only one pixel of an image, generating an effective attack against ℓ_p -bounded adversarial training.

3. Preliminaries

Attacker. We assume the attacker has access to the original data they want to make unlearnable, but cannot alter the training process [12]. Typically, the attacker attempts to make the data unlearnable by adding perturbations to the images to prevent trainers from using them to learn a classifier that generalize well to the original data distribution. Formally, suppose we have a dataset consisting of original clean examples $\mathcal{D}_{\text{clean}} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ drawn from a distribution \mathcal{S} , where $\mathbf{x}_i \in \mathcal{X}$ is an input image and $y_i \in \mathcal{Y}$ is its label. The attacker thus aims to construct a set of sample-specific unlearning perturbations $\delta = \{\delta_{\mathbf{x}} | \mathbf{x} \in \mathcal{X}\}$, in order to make the model $f_{\theta}: \mathcal{X} \rightarrow \mathcal{Y}$ trained on the *unlearnable examples* set $\mathcal{D}_{\text{ue}}(\delta) = \{(\mathbf{x} + \delta_{\mathbf{x}}, y) | (\mathbf{x}, y) \in \mathcal{D}_{\text{clean}}\}$ perform poorly on a test set $\mathcal{D}_{\text{test}}$ sampled from \mathcal{S} :

$$\begin{aligned} & \max_{\delta} \mathbb{E}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_{\text{test}}} [\mathcal{L}(f_{\theta^*}(\delta)(\mathbf{x}_i), y_i)], \\ \text{s.t. } & \theta^*(\delta) = \operatorname{argmin}_{\theta} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}_{\text{ue}}(\delta)} \mathcal{L}(f_{\theta}(\hat{\mathbf{x}}_i), y_i), \end{aligned} \quad (1)$$

where \mathcal{L} is the loss function, typically the softmax cross-entropy loss. For each image, the noise δ_i is bounded by $\|\delta_i\|_p \leq \epsilon$, where ϵ is a small perturbation budget such that it may not affect the intended utility of the image, and $\|\cdot\|_p$ denotes the ℓ_p norm. Table 1 provides samples generated by unlearnable example attacks and their corresponding perturbations (amplified with normalization).

Defender. The goal of the defender is to ensure that the trained model learns from the poisoned training data, allowing the model to be generalized to the original clean data distribution $\mathcal{D}_{\text{clean}}$. The attacker assumes full control of its training process. In our context, we thus assume that the attacker’s policy is to perform poison removal on the image, in order to ensure the trained model generalizes even when trained on poisoned data \mathcal{D}_{ue} . It has been shown in [7, 19, 11] that Adversarial training [14] is effective against unlearnable examples, which optimizes the following objective:

$$\operatorname{argmin}_{\theta} \mathbb{E}_{(\hat{\mathbf{x}}, y) \sim \mathcal{D}_{\text{ue}}} \left[\max_{\|\delta_{\text{adv}}\|_p \leq \epsilon} \mathcal{L}(f_{\theta}(\hat{\mathbf{x}} + \delta_{\text{adv}}), y) \right]. \quad (2)$$

Specifically for each image $\hat{\mathbf{x}} \in \mathcal{D}_{\text{ue}}$, it finds an adversarial perturbation δ_{adv} that maximizes the classifier loss. It then performs gradient descent on the maximal loss to optimize for the model θ . A model trained on the unlearnable set \mathcal{D}_{ue} in this manner thus gain robustness to perturbations in the input, and can generalize to clean images.

4. Adversarial Augmentations

Adversarial training can be viewed as a practical data augmentation policy, which presents an interesting perspective as it allows the model to choose its own policy in the form of ℓ_p -bounded perturbations adaptively. However, it poses a considerable challenge due to its use of large defensive perturbations, often resulting in reduced accuracy. This begs the question of whether new defense mechanisms can leverage *unseen* threat models that unlearnable attacks may be unable to account for.



















Inspired by this, we introduce *UEraser*, which performs adversarial augmentations polices that preserves to the semantic information of the images rather than adding ℓ_p -bounded adversarial noise. Our objective is a bi-level optimization, where the inner level samples image transformation policies $\mathcal{T}(\cdot)$ from a set of all possible augmentations \mathcal{A} , in order to maximize the loss, and the outer level performs model training with adversarial polices:

$$\operatorname{argmin}_{\theta} \mathbb{E}_{(\mathbf{x}, y) \sim \mathcal{D}_{\text{ue}}} [\max_{\mathcal{T} \sim \mathcal{A}} \mathcal{L}(f_{\theta}(\mathcal{T}(\mathbf{x})), y)]. \quad (3)$$

Intuitively, *UEraser* finds the most “adversarial” augmentation policies for the current images, and use that to train the model in order to prevent unlearnable “shortcuts” from emerging during model training. Compared to adversarial training methods that confine the defensive perturbations within a small ϵ -ball of ℓ_p distance, here we adopt a different approach that allows for a more aggressive distortion. Moreover, augmentation policies also effectively preserve the original semantics in the image. By maximizing the adversarial loss in this manner, the model can thus avoid training from the unlearning “shortcuts” and instead learn from the original features.

To generate augmentation policies with high-level of distortions, we select PlasmaTransform [16], and TrivialAugment [15], two modern suites of data augmentation policies, and ChannelShuffle in sequence, to form a strong pipeline of data augmentations polices. PlasmaTransform performs image distortion with fractal-based transformations. TrivialAugment provide a suite of natural augmentations which shows great generalization abilities that can train models with SOTA accuracies. Finally, ChannelShuffle swaps the color channels randomly, this is added to further increase the aggressiveness of adversarial augmentation policies. Interestingly, using this pipeline without the error-maximization augmentation sampling can also significantly reduce the effect of unlearning perturbations.

Table 1: The visualization of unlearned examples and perturbations of eight poisoning methods on CIFAR-10.

Poisons	EM [11]	REM [7]	HYPO [19]	NTGA [24]	TAP [6]	LSP [21]	AR [17]	OPS [20]	
Clean									
Perturbations									
Type	$\ell_\infty, \epsilon = 8/255$					$\ell_2, \epsilon = 1.30$		$\ell_2, \epsilon = 1.00$	$\ell_0, \epsilon = 1$

Finally, we provide an algorithmic overview of *UEraser* in Algorithm 1. It accepts as input the poisoned training dataset \mathcal{D}_{uc} , batch size B , randomly initialized model f_θ , number of training epochs N , number of error-maximizing augmentation epochs W , learning rate α , number of repeated sampling K , and a suite of augmentation policies \mathcal{A} . For each sampled mini-batch \mathbf{x} , \mathbf{y} of data points from the dataset, it applies K different random augmentation policies for each image in \mathbf{x} , and compute the corresponding loss values for all augmented images. It then selects for each image in \mathbf{x} , the maximum loss produced by its K augmented variants. The algorithm then uses the averaged loss across the same mini-batch to perform gradient descent on the model parameters. At last, the algorithm returns the trained model parameters θ after completing the training process.

From Algorithm 1, we can know the training of *UEraser* is affected by two hyperparameters, namely, the numbers of repeated augmentation samples K per image and the epochs of error-maximizing augmentations W . We denote the approach that uses only combinations of data augmentations as *UEraser-Lite*. It requires only 1 augmentation sample per training image. *UEraser* and *UEraser-Max*, on the other hand, use error-maximizing augmentations, with the only difference in the epochs of error-maximizing augmentations W . *UEraser-Max* utilizes the maximization augmentations throughout, whereas *UEraser* requires use in the early stages of training. In most cases, *UEraser* or *UEraser-Lite* can achieve similar results to *UEraser-Max*. Therefore, it is more practical to use *UEraser-Lite* or *UEraser* due to its faster training speed.

5. Experimental results

To demonstrate the effectiveness of *UEraser*, we selected 8 SOTA unlearnable example attacks, namely: Error-Minimization (EM) [11], Hypocritical Perturbations (HYPO) [19], NTGA [24], Robust Error-Minimization (REM) [7], Adversarial Poisons (TAP) [6], Linear-separable Synthetic Perturbations (LSP) [21], Autoregressive Poisoning (AR) [17], and One-pixel Shortcut (OPS) [20]. The experimental results of *UEraser* and defenses including adversarial training [14] and ISS [13] on the CIFAR-10 dataset are shown in Table 2. For adaptive attacks, we evaluated

Table 2: Clean test accuracies (%) of *UEraser* on CIFAR-10. All experiments are conducted on the ResNet-18. “U-Lite”, “U”, and “U-Max” respectively denote *UEraser-Lite*, *UEraser*, and *UEraser-Max*. “ST” and “AT” are standard and adversarial training respectively.

Methods	ST	U-Lite	U	U-Max	ISS [13]	AT
EM	21.24	90.78	93.38	95.24	92.27	83.02
REM	33.12	85.49	91.02	92.54	91.34	82.87
HYPO	72.12	85.67	87.59	88.67	84.77	85.49
NTGA	18.15	78.29	84.41	87.94	72.65	70.05
TAP	7.32	83.29	84.17	82.47	83.05	81.19
LSP	14.95	84.92	85.07	94.95	82.71	84.27
AR	12.04	87.12	88.64	89.82	84.67	84.16
OPS	15.20	68.50	73.22	81.84	77.81	11.08

Table 3: Adaptive poisoning with EM on CIFAR-10.

Methods	Standard	<i>UEraser-Lite</i>	<i>UEraser</i>	<i>UEraser-Max</i>
Baseline	21.21	90.78	93.38	95.24
+ <i>UEraser-Lite</i>	29.36	81.24	87.68	89.55
+ <i>UEraser-Max</i>	35.24	60.15	71.04	80.28

UEraser where the adversary leverages our *UEraser* variants, and Table 3 shows the results.

6. Conclusion

Using the intuition of disrupting the unlearning perturbation with perturbations beyond the ℓ_p budgets, we proposed a simple yet effective defense method called *UEraser*, which can mitigate unlearning poisons and restore clean accuracies. *UEraser* achieves robust defenses on unlearning poisons with simple data augmentations and adversarial augmentation policies. Similar to adversarial training, it employs error-maximizing augmentation to further eliminate the impact of unlearning poisons. Our experiments on state-of-the-art unlearnable example attacks demonstrate that *UEraser* outperforms existing countermeasures such as adversarial training and ISS. Our results suggest that existing unlearning perturbations are tragically inadequate in making data unlearnable. By understanding their weaknesses, we can anticipate how malicious actors may exploit them, and prepare safeguards against such threats. We hope *UEraser* can help facilitate future research in these attacks and defenses.

References

- [1] Francesco Croce, Maksym Andriushchenko, Vikash Sehwal, Edoardo DeBenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.
- [2] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.
- [3] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. AutoAugment: Learning augmentation strategies from data. In *CVPR*, 2019.
- [4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.
- [5] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv:1708.04552*, 2017.
- [6] Liam Fowl et al. Adversarial examples make strong poisons. In *NeurIPS*, 2021.
- [7] Shaopeng Fu, Fengxiang He, Yang Liu, Li Shen, and Dacheng Tao. Robust unlearnable examples: Protecting data privacy against adversarial learning. In *ICLR*, 2022.
- [8] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2020.
- [9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv:1412.6572*, 2014.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [11] Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *ICLR*, 2021.
- [12] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [13] Zhuoran Liu, Zhengyu Zhao, and Martha Larson. Image shortcut squeezing: Countering perturbative availability poisons with compression. In *ICML*, 2023.
- [14] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.
- [15] Samuel G Müller and Frank Hutter. TrivialAugment: Tuning-free yet state-of-the-art data augmentation. In *CVPR*, 2021.
- [16] Angelos Nicolaou, Vincent Christlein, Edgar Riba, Jian Shi, Georg Vogeler, and Mathias Seuret. Tormentor: Deterministic dynamic-path, data augmentations with fractals. In *CVPR Workshop*, 2022.
- [17] Pedro Sandoval-Segura, Vasu Singla, Jonas Geiping, Micah Goldblum, Tom Goldstein, and David W Jacobs. Autoregressive perturbations for data poisoning. In *NeurIPS*, 2022.
- [18] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv:1312.6199*, 2013.
- [19] Lue Tao, Lei Feng, Jinfeng Yi, Sheng-Jun Huang, and Songcan Chen. Better safe than sorry: Preventing delusive adversaries with adversarial training. In *NeurIPS*, 2021.
- [20] Shutong Wu et al. One-pixel shortcut: on the learning preference of deep neural networks. In *ICLR*, 2023.
- [21] Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Availability attacks create shortcuts. In *ACM SIGKDD*, 2022.
- [22] Yunrui Yu, Xitong Gao, and Cheng-Zhong Xu. LAFEAT: Piercing through adversarial defenses with latent features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5735–5745, June 2021.
- [23] Yunrui Yu, Xitong Gao, and Cheng zhong Xu. MORA: Improving ensemble robustness evaluation with model reweighing attack. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.
- [24] Chia-Hung Yuan and Shan-Hung Wu. Neural tangent generalization attacks. In *ICML*, 2021.
- [25] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

A. Statement

Availability poisoning aims to prevent unauthorized training on personal data. From this perspective, this type of data poisoning is usually considered as a defender, whereas implementing malicious training on protected data is considered as an attacker. However, data poisoning is widely recognized as a backdoor attack method. Thus in this paper, we define the poisoner as the attacker and the UEraser as the defender.

B. Algorithm

Algorithm 1 Training with *UEraser*.

```

1: function UERASER( $\mathcal{D}_{ue}, B, f_{\theta}, N, W, \alpha, K, \mathcal{A}$ )
2:   for  $n \in [1, \dots, N]$  do
3:     if  $n \geq W$  then  $K \leftarrow 1$  end if
4:     for  $(\mathbf{x}, \mathbf{y}) \sim \text{minibatch}(\mathcal{D}_{ue}, B)$  do
5:       for  $i \in [1, \dots, B]$  do
6:         for  $j \in [1, \dots, K]$  do
7:            $\text{aug} \sim \mathcal{A}$ 
8:            $\mathbf{L}_{ij} \leftarrow \mathcal{L}(f_{\theta}(\text{aug}(\mathbf{x}_i)), \mathbf{y}_i)$ 
9:         end for
10:         $\mathbf{L}_i^{\text{adv}} \leftarrow \max_{j \in [1, \dots, K]} \mathbf{L}_{ij}$ 
11:       end for
12:        $\theta \leftarrow \theta - \alpha \nabla_{\theta} \frac{1}{B} \sum_{i \in [1, \dots, B]} \mathbf{L}_i^{\text{adv}}$ 
13:     end for
14:   end for
15:   return  $\theta$ 
16: end function

```

C. Standard Augmentation

For baselines to compare against, we perform data augmentation via random flipping, and random cropping to 32×32 images on each image.

D. Unlearning Perturbation Budgets

The attacks, EM [11], REM [7], and HYPO [19], all have a permitted perturbation bound of $\ell_{\infty} = 8/255$ for each image. Additionally, the LSP [21] and AR [17] attacks permit $\ell_2 = 1.30$ and $\ell_2 = 1.00$, respectively.

E. Computational Overhead

E.1. Adversarial Training

For comparison, the baseline defenses against the attack methods on CIFAR-10 employ PGD adversarial training [14], following the evaluation of [7]. The adversarial training perturbation bounds used were $\ell_{\infty} = 8/255$ as baseline defenses.

Table 4: Computational overhead of *UEraser*.

Method	GPU hours (V100)
Standard	0.6
<i>UEraser-Lite</i>	1.0
<i>UEraser</i>	2.1
<i>UEraser-Max</i>	5.8

F. Attack and Defense Baselines

We use eight baseline attacks and two existing SOTA defenses for evaluation and comparisons in our experiments (Table 2). Each attack method is implemented from their respective official source code for a fair comparison. We adopt experimental setup identical to the original publications, and use perturbation budgets described in Appendix D. For defenses, we compare *UEraser* variants against the current SOTA techniques, image shortcut squeezing [13] and adversarial training [14]. The compared defenses (ISS and adversarial training) respectively follow the original source code and PGD adversarial training [14].