An Information Theoretic Approach to Machine Unlearning

Anonymous authors
Paper under double-blind review

Abstract

To comply with AI and data regulations, the need to forget private or copyrighted information from trained machine learning models is increasingly important. The key challenge in unlearning is forgetting the necessary data in a timely manner, while preserving model performance. In this work, we address the zero-shot unlearning scenario, whereby an unlearning algorithm must be able to remove data given only a trained model and the data to be forgotten. We explore unlearning from an information theoretic perspective, connecting the influence of a sample to the information gain a model receives by observing it. From this, we derive a simple but principled zero-shot unlearning method based on the geometry of the model. Our approach takes the form of minimising the gradient of a learned function with respect to a small neighbourhood around a target forget point. This induces a smoothing effect, causing forgetting by moving the boundary of the classifier. We explore the intuition behind why this approach can jointly unlearn forget samples while preserving general model performance through a series of low-dimensional experiments. We perform extensive empirical evaluation of our method over a range of contemporary benchmarks, verifying that our method is competitive with state-of-the-art performance under the strict constraints of zero-shot unlearning.

1 Introduction

Regulations such as the General Data Protection Regulation (GDPR) enshrine an individual's data autonomy rights, including the right to be forgotten. While deleting an entry from a database is relatively straightforward, removing the influence of that data from a trained model is a challenging open problem. The process of removal, referred to as *unlearning*, is difficult for several reasons. It is known that neural networks memorise instance-level information (Arpit et al., 2017; Zhang et al., 2021; Feldman, 2020), and it is practically intractable to ascribe parameter changes to a specific training sample post-hoc (Kurmanji et al., 2023). At its core, unlearning is a multi-objective optimization problem with three key desiderata. An effective unlearning algorithm must remove the influence of the selected subset of data, maintain model performance on retained data, and minimise computational cost. These goals are antagonistic since inducing forgetting inevitably disrupts the model's performance, thus balancing these objectives is key. Naïvely, one may achieve perfect forgetting by retraining a model on the training data sans the forget samples every time there is a forget request. However, this is prohibitively expensive thus violating the third desideratum.

Many of the unlearning methods proposed are effective, however they make strong assumptions about the problem setting that simplifies the task considerably. Primarily, existing methods typically assume access to all, or a subset of, the training data. This data is used in different ways, such as to fine-tune the model post-forgetting (Chundawat et al., 2023a; Graves et al., 2021), or to conduct parameter importance calculations after the initial training period (Foster et al., 2023). In reality, there are many reasons why this data could be unavailable, such as cost of storage, limited duration access to datasets, or an oversight in considering machine unlearning during model development. As such, Chundawat et al. (2023b) introduce a novel problem setting for unlearning, termed *zero-shot* (ZS) unlearning, whereby no data is available to the unlearning model. A limitation of this problem definition is that, without a concrete way to define the forget set, we are constrained to forgetting full class only. As such, we explore the zero-shot setting that permits an arbitrary forget set to be defined, but importantly only the data to be forgotten and the trained model are

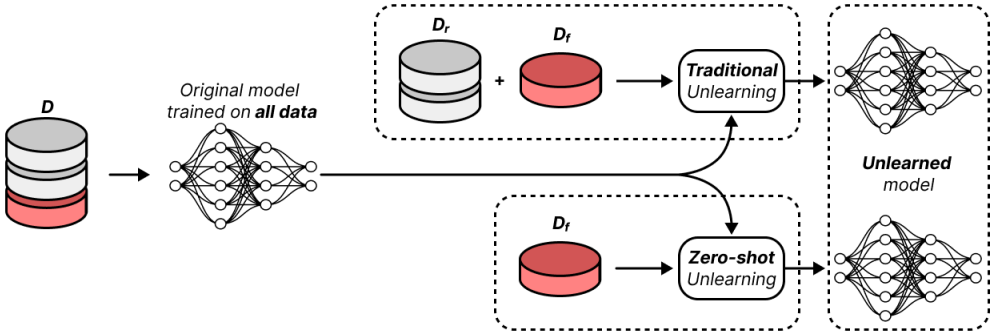

Figure 1: Visualization of the zero-shot unlearning scenario. Contrary to traditional unlearning there is no access to, or prior knowledge of, any data other than the forget set or the model at any point beyond its current state. These constraints make the problem considerably more challenging.

available (Figure 1). This is extremely challenging, since the remaining data is not available to protect model performance, and thus more delicate methods are required. Insightful treatment of the ZS scenario can be found in Chen et al. (2023), where unlearning is formulated as reconstructing a decision boundary that could be reasonably learnt by a model trained without the forget data, achieved through learning the nearest false label for each forget sample.

In this work, we approach ZS unlearning from an information theoretic perspective. Golatkar et al. (2020) consider information leakage when observing model weights, whereas we consider the information gained by a model by training on a given sample. Data points offer a classifier different amounts of information gain when included in the training data (Lindley, 1956; Houlsby et al., 2011). If a data point can be inferred from other training data, then it offers little information gain (Jeong and Qiu, 2018). In ZS unlearning, knowing the contribution a sample has made to a model is hard as we have access to only the model and the sample to be forgotten. We postulate that the information gain of a sample can serve as a good indicator for how much influence it has over a model. From this hypothesis, we derive a principled loss to induce forgetting based on how the geometry of the problem space changes with respect to the softmax classifier. If a sample offers little information gain then it may be inferred from other data, therefore the classifier's output should change minimally over similar samples. This minimal change can be effectively measured via the rate of change of the model with respect to the input, which should be low in the region surrounding low-information samples. In contrast, a high information gain sample cannot be inferred, and therefore one would expect they lie in high rate of change regions of space. We present Just in Time (JiT) unlearning, a novel ZS unlearning algorithm based on minimising the gradients of a classifier with respect to local neighbourhood around each forget sample. We show in low dimensional experiments that removing training samples from low-gradient regions yields minimal changes to the learned decision boundary, whereas removing samples from high gradient regions has a more pronounced effect. We therefore propose that, with reference to Feldman (2020), samples with low information gain may be predicted with more generalised knowledge and thus do not infringe on privacy. In contrast, samples with large information gain have significant impact on the learned classifier, are more likely memorised, and do infringe on privacy. We demonstrate empirically that following these principles, JiT causes the removal of influence from the forget set while preserving generalisation performance across the wider space.

Our primary contributions are as follows:

- To the best of our knowledge, JiT is the first unlearning algorithm to be directly informed by the information gain of a sample.
- We provide extensive empirical analysis of the geometry of JiT unlearning in low dimensions.
- We show our method is competitive with existing SOTA in the zero-shot domain.

2 Related Work

Information theory is concerned with the transmission, quantification, and storage of information (Shannon, 1948), and has seen widespread use in machine learning. Most relevant here is its use in determining the information gain of an experiment (Lindley, 1956). This notion has seen uses such as determining splits in decision trees (Quinlan, 1986), and active learning (Tong and Koller, 2001; Houlsby et al., 2011). Here we use this concept as a proxy to a training sample's influence on a learned function.

Machine unlearning was first introduced in Cao and Yang (2015), and a probabilistic perspective of unlearning was explored in Ginart et al. (2019); Sekhari et al. (2021); Gupta et al. (2021); Neel et al. (2021); Triantafillou et al. (2023). Here we focus on post-hoc unlearning methods that operate on models that are already trained. Methods exist that alter the initial training scheme (Bourtoule et al., 2021; Mehta et al., 2022; Shah et al., 2023), but these are considered out of scope as they do not satisfy the ZS problem constraints.

Current SOTA methods rely on accessing all or a subset of the original dataset that is *not* to be forgotten (i.e. the retain set), thus violating the ZS constraints. Bad Teacher unlearning (Chundawat et al., 2023a) and SCRUB (Kurmanji et al., 2023) leverage a student-teacher framework while Amnesiac unlearning (Graves et al., 2021) trains with randomised labels for forget data before fine-tuning on the retained data to repair the model. UNSIR (Tarun et al., 2023) learns an error-maximising noise to induce forgetting of the necessary data, before also employing a finetuning step. Warnecke et al. (2021) minimise the divergence in model output over a sample and its noisy perturbations and then finetune. A key limitation of these methods is that protecting model performance necessitates access to retain-set data for the entire duration of the model's lifetime. To address this, Golatkar et al. (2020) and Foster et al. (2023) propose methods that do not require fine-tuning or repair steps. Golatkar et al. (2020) derives an unlearning algorithm that minimises information gained about the training data when observing model weights. However, this scales quadratically with dataset size and often performs considerably worse than state-of-the-art (Tarun et al., 2023; Foster et al., 2023). Selective Synaptic Dampening is a scalable retrain-free approach, based on inducing forgetting by selectively dampening parameters that are disproportionately important to the forget-set (Foster et al., 2023). This requires access to the whole dataset at least once, to calculate the importance over the retained data. Chundawat et al. (2023b) introduces two methods to address ZS unlearning. The first method, an extension of Tarun et al. (2023), replaces the repair step with an error-minimising noise. The second utilises a generator network and an attention loss to distil knowledge from an expert teacher, with a band-pass filter preventing the flow of knowledge for specific classes. Both methods are slow, do not scale well to large problem spaces, and can only forget entire classes. Chen et al. (2023) present boundary shrinking and boundary expanding. Shrinking causes unlearning by training over the nearest false label for forget samples, found via a fast gradient sign attack (Goodfellow et al., 2014). While performant, shrinking scales poorly with model and input size. Boundary expanding is faster but less performant, remapping forget samples by training them to fit a new output neuron, before removing the neuron leaving the forget samples in high entropy states.

We note that many recent works have focused on generative models such as LLMs. Most evaluations in this space consider unlearning in the context of question answering (QA) rather than classification, for example, the TOFU benchmark focuses on QA about fictitious authors (Maini et al., 2024). The WMDP benchmark features four multiple choice questions about weapons of mass destruction (Li et al., 2024). The same work introduces Representation Misdirection Unlearning, which trains the LLM to produce random unit vectors as intermediate embeddings when prompted with hazardous queries, while simultaneously finetuning over a retain set to preserve wider knowledge. Negative preference optimization is an LLM unlearning mechanism that can be viewed as a stable formulation of gradient ascent over the forget data (Zhang et al., 2024). An important nuance of forgetting in LLMs is that forgetting needs to apply to a sequence of tokens, but the constituent tokens need to remain accessible to the model, thus making the domains fairly distinct, however Li et al. (2024) does show that SCRUB and SSD can generalize to LLMs reasonably. Despite LLMs receiving recent attention, classification remains a cornerstone of deep learning; its prevalance in both large-scale and sensitive problems in industry ranging from social-media to healthcare necessitates the continued research of unlearning in this domain.

Membership inference attacks (MIA) are a way of measuring information leakage of a machine learning model (Shokri et al., 2017). An auxiliary model (e.g. a logistic regression) is trained to infer whether a given

data point was included in a model's training data. MIAs are used as typical evaluation metrics in machine unlearning; if a MIA cannot recognise a forgotten sample as an element from the train set, then this presents empirical evidence that the sample has indeed been forgotten.

Randomized smoothing constructs an adversarially robust classifier by adding random noise to the input during training (Cohen et al., 2019), making the model less sensitive to small perturbations in the input space. This has clear connections to the unlearning method we present in this work, which induces forgetting through a post-hoc, localized variation of randomized smoothing.

3 Preliminaries

We introduce the notation for machine unlearning in a supervised classification task, consistent with the approach outlined in Chen et al. (2023). Consider some input space $\mathcal{X} \subset \mathbb{R}^d$ and some output label space $\mathcal{Y} \subset \mathbb{R}^c$, where d is the dimensionality of the input and c is the number of classes. We define a training dataset $\mathcal{D} = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N \subseteq \mathcal{X} \times \mathcal{Y}$, where $\boldsymbol{x}_i$ is a training input sample with the label $\boldsymbol{y}_i$. We denote a subset $\mathcal{D}_f \subseteq \mathcal{D}$ as the forget set, and $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$ as the retain set.

Let $f_\theta : \mathcal{X} \to \mathcal{Y}$ be a neural network, with parameters θ. We assume that f_θ is well trained and generalises to in distribution test samples well. The objective of ZS unlearning is, given only the model f_θ trained on $\mathcal{D}$, to remove the influence of $\mathcal{D}_f$ from the learned model such that the unlearnt model $f_{\theta'}$ is approximately equivalent to a model retrained on only $\mathcal{D}_r$ which we define as the optimal solution, f_{θ^*}. Since direct access to f_{θ^*} is, by definition of the unlearning problem, impossible, existing works construct approximate heuristics to induce forgetting and use a membership inference attack to measure forgetting. These attacks typically evaluate the difference in the output distributions of the model over train and test samples.

4 Proposed Method

In this section, we introduce our JiT unlearning method and provide intuition for its effectiveness by examining the geometry of a learned classifier and analyzing how the location of a forget sample in the input space can impact the decision boundary of a model retrained without it. This section considers the case where $|\mathcal{D}_f| = 1$, noting that larger subsets have an additive effect.

Consider the hypothetical of removing a single image of a black cat from a dataset comprised of 1 million black cats and 1 million white dogs. This likely has minimal effects on the learning process, since a well trained classifier should generalise and infer the class of the sample easily. If a sample may be removed from the training dataset without significant changes in the resultant model, we posit that an unlearning algorithm should also have minimal effect on the model when unlearning such a sample. As such, it is logical to design an unlearning algorithm that accounts for the information gain of a sample. However, directly measuring this quantity is difficult, especially in a ZS setting where there is no access to other data points. We therefore seek to derive a heuristic that can approximate how much information gain a sample may have offered the model, based on only that sample and the model itself. We begin by formally introducing the notion of a neighbourhood of a target sample:

Definition 4.1 (Neighbourhood of a sample)**.** Let $\mathcal{X} \subseteq \mathbb{R}^d$ be an input space of samples, and assume each sample $\mathbf{x} \in \mathcal{X}$ is associated with a class label $c(\mathbf{x}) \in \{1, 2, ..., K\}$. Fix a radius r > 0. For each $\boldsymbol{x} \in \mathcal{X}$, define its neighbourhood $\mathcal{B}_r(\mathbf{x})$ as:

$$\mathcal{B}_r(\mathbf{x}) = \{\mathbf{z} \in \mathcal{X} : ||\mathbf{z} - \mathbf{x}|| \leq r\}.$$

Then,

Definition 4.2 (Information of a sample)**.** For each $\mathbf{x} \in \mathcal{X}$, define:

$$\alpha(\mathbf{x}) = \frac{1}{|\mathcal{B}_r(\mathbf{x})} \sum_{\mathbf{z} \in \mathcal{B}_r(\mathbf{x}} \mathbb{I}\{c(\mathbf{z}) = c(\mathbf{x})\},$$

where $\mathbb{I}\{\cdot\}$ is the indicator function. In other words, $\alpha(\mathbf{x})$ is the propoertion of $\mathbf{x}$'s neighbours that share its class label. Choose a threshold $\tau \in [0, 1]$. We say that a sample $\mathbf{x} \in \mathcal{X}$ is low information if $\alpha(\mathbf{x}) \geq \tau$, and high information otherwise.

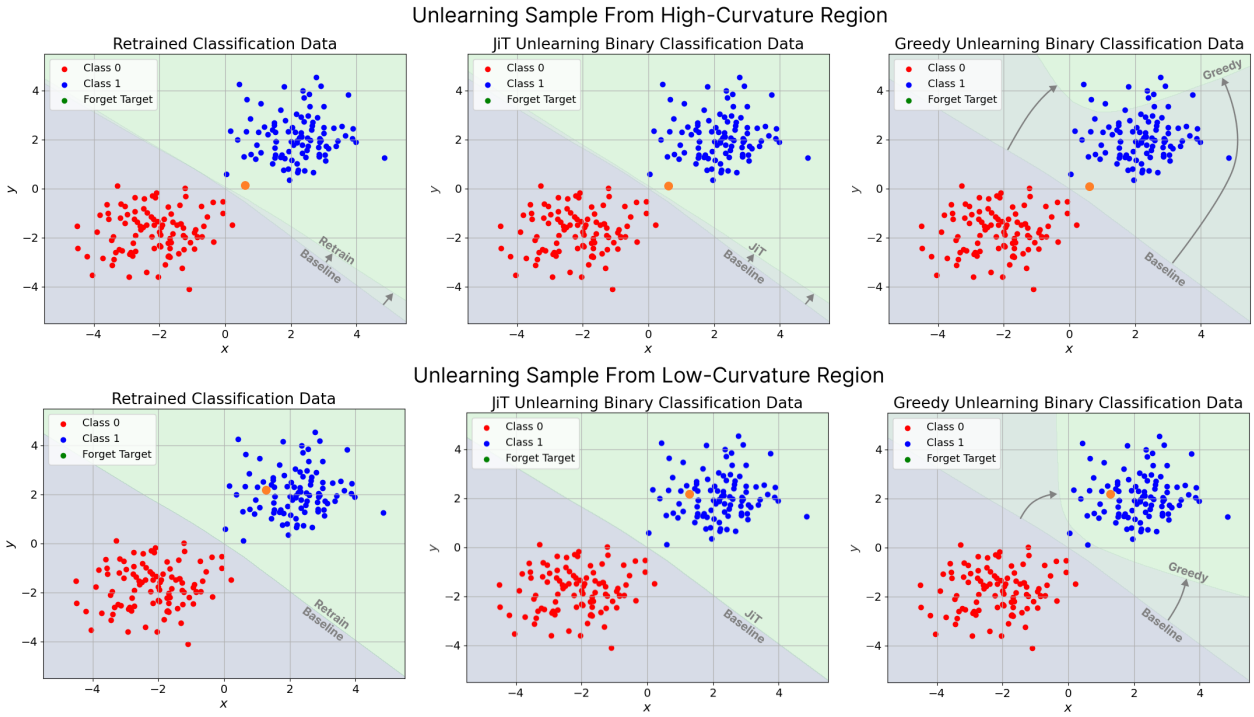

Figure 2: Demonstration of how the boundary of a classifier moves during unlearning. Retrained model is the gold standard. Removing a sample from a low-gradient region has almost no effect on the retrained model, whereas removing a sample from high gradient space has significant impact. In this low-dimensional setting, JiT successfully reconstructs the retrained boundary, whereas naively training to mislabel the forget sample completely destroys the trained model.

Plainly, we can say that a data point may be said to be low information if it can be inferred from its neighbourhood; and high information if it can not. Consider a low information training sample x_l, from definitions 4.1 and 4.2 we can expect that for some bound r, $f_\theta(\boldsymbol{x_l}) \approx f_\theta(\hat{\boldsymbol{x}}_{\boldsymbol{l}})\ \forall \hat{\boldsymbol{x}}_{\boldsymbol{l}} \in \mathcal{B}_r(\boldsymbol{x}_l)$. In other words, a model's predictions over a low information sample and a set of similar data should have similar output distributions. As such, the rate of change of the model in this space will be low. However, for a high information gain sample, this would not necessarily hold. From this we can describe an unlearning objective; if the classifier exhibits small output variation with respect to input perturbations around a forget sample (i.e., has small local gradients), then this sample likely has minimal influence on the decision boundary. Hence, we present a method based on minimising the gradient of the classifier with respect to input generated by perturbing points in the forget set. Since taking the gradient of the model with respect to the input is extremely expensive for larger problems, we instead construct a first order approximation to the gradient at the target via considering noisy perturbations within its neighbourhood. Formally, $\forall \boldsymbol{x} \in \mathcal{D}_f$, we seek to minimise the loss given below:

$$\ell = \mathbb{E}\left(\frac{\|f_\theta(\boldsymbol{x}) - f_\theta(\boldsymbol{x}+\boldsymbol{\xi})\|_2}{\|\boldsymbol{x} - (\boldsymbol{x}+\boldsymbol{\xi})\|_2}\right) \approx \frac{1}{N}\sum_{j=1}^{N}\left(\frac{\|f_\theta(\boldsymbol{x}) - f_\theta(\boldsymbol{x}+\boldsymbol{\xi}_j)\|_2}{\|\boldsymbol{\xi}_j\|_2}\right). \tag{1}$$

Where $\boldsymbol{\xi}$ is a noise vector of equivalent dimensionality to $\boldsymbol{x}$, and each component ξ_i of $\boldsymbol{\xi}$ is independently drawn from a Gaussian distribution such that $\xi_i \sim \mathcal{N}(0, \sigma^2)$. For samples that are highly influential, minimising this loss will smooth the local region and remove its influence from the model. For low-information samples that are generalised knowledge, the neighbourhood will be rather smooth resulting in minimal changes. A full algorithm for JiT is given in the appendix 10.1.

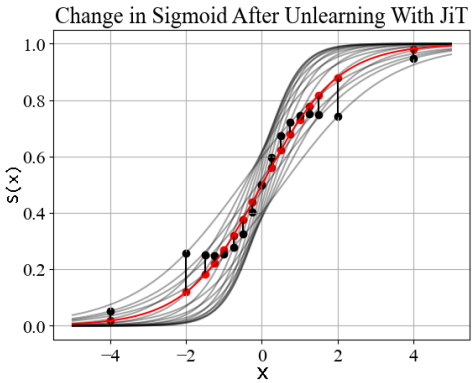

Figure 3: Change in sigmoid after unlearning with JiT. Red dots are unlearnt samples, black dots are the location on the new sigmoid post-JiT.

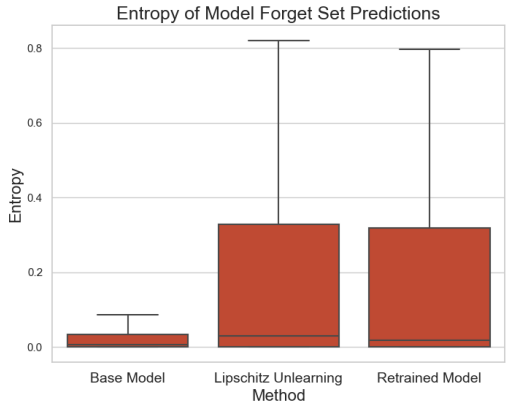

Figure 4: Entropy, $\mathcal{H}(x)$, of the $\mathcal{D}_f$ output distributions for full-class unlearning on CIFAR-10, showing JiT exhibits performance similar to the retrained model.

4.1 A Geometric Interpretation of JiT

We now present JiT from a geometric perspective, providing insight into why it causes forgetting and how it protects the wider function. Consider a simple 2D classification task, as visualised in figure 2. We pose a simple question: does it make sense to forget all regions of space equally? Naturally, the answer is no. Completely forgetting a sample from within a low-gradient region of space would necessitate the forgetting of almost the entire class, even if they are not part of the forget set. Furthermore, removing a sample from this region would not have significant ramifications on a model retrained from scratch, nor is it likely infringing on the privacy of an individual. Unlearning in this instance often requires minimal alterations to the model. In contrast, a sample that lies in a high-gradient region may not only have significant influence over the position of the learned boundary, but may have been misclassified had it not been included in the training data. Figure 2 shows that this intuition holds in low dimensions; when the forget sample is within the low-gradient region, a model retrained on $\mathcal{D}_r$ exhibits almost no change, whereas when the sample in $\mathcal{D}_f$ is in a high-gradient region, the boundary is shifted considerably. Enshrining such behaviour into a ZS unlearning algorithm is tantamount; as protection through fine-tuning or regularization is not possible, a ZS algorithm must be surgical in its forgetting methodology. Figure 2 shows that unlearning using JiT yields a classifier almost identical to the retrained model in this low-dimensional setting whereas greedily training over $\mathcal{D}_f$ with a false label causes complete destruction of the model.

The heuristic behind JiT's performance is based on the gradient field of the classifier. The crux of this rests upon the inherent non-linearity of neural networks. Typically well trained models exhibit sharp decision boundaries with a large rate of change, with flatter behaviour within the class (Fridovich-Keil et al., 2022). As such, given two unit noise vectors $\boldsymbol{\xi_i}, \boldsymbol{\xi_j}$, where $\boldsymbol{\xi_i}$ points towards the decision boundary and $\boldsymbol{\xi_j}$ points away, the gradient of the classifier between $\boldsymbol{x}$ and $\boldsymbol{x} + \boldsymbol{\xi_i}$ will be larger than for $\boldsymbol{x} + \boldsymbol{\xi_j}$. As such, minimising equation 1 for samples near a boundary will be biased towards moving the boundary *towards* $\boldsymbol{x}$. This has the consequence of increasing the uncertainty of the prediction and potentially changing the samples' predicted class. To further highlight this phenomena, figure 3 shows how unlearning forget samples (red dots) from a learned sigmoid function (red line) changes the learned function. Two things should be observed here: first, samples that lie in low gradient regions have relatively small changes and secondly, the updates to the function have the effect of pulling the the forget sample towards the centre of the sigmoid, which is the decision boundary. Unlearning in this way increases model uncertainty over forgotten samples, without destroying the wider function.

4.2 Entropy Similarity

In the previous section, we demonstrated how in low dimensions JiT can induce forgetting of a single sample in a way similar to retraining the model from scratch. Now, we demonstrate that the same loss can be used to forget arbitrary subsets $\mathcal{D}$ in higher dimensions, including full classes. Since visualising decision boundaries in high dimensions is challenging, we instead evaluate the entropy of the model output. We train a 2-layer CNN on the CIFAR-10 dataset, focusing specifically on the task of forgetting class 0. We compare the entropy of the unlearned model over class 0 with that of the original model, and a model retrained from scratch without class 0. Intuitively, low-entropy predictions indicate higher model confidence, and therefore we expect that the entropy of the model after JiT unlearning is applied will be higher, aligning closely with that of the retrained model.

Figure 4 shows the entropy of the forget-set output distributions for a CNN trained on CIFAR-10. Our unlearning approach increases the entropy over the forget set, reducing the divergence between it and that of a model retrained from scratch on $\mathcal{D}_r$. In fact, under a Wilcoxon signed-rank test (Woolson, 2007), we find there is no statistically significant difference between the model unlearned with JiT and the retrained model for $p = 0.10$. JiT and retraining both increase the entropy over the forget set, suggesting the resultant models behave in a similar way, possessing less knowledge of the forget samples compared to the baseline model. Alongside matching the entropy over the forget set, JiT preserve model performance, as the unlearned model drops only 2% accuracy (From 99% to 97%) on $\mathcal{D}_r$. Our algorithm demonstrates promising characteristics that are indicative of an effective unlearning algorithm.

5 Experimental Setup

5.1 Benchmarks

We implement the same benchmarks from Foster et al. (2023), which are similar to that of Chundawat et al. (2023a), Golatkar et al. (2020) and Kurmanji et al. (2023). We run experiments 10 times, reporting the mean and standard deviation of these performances. Where classes or sub-classes are forgotten, we show performance over the same class/sub-class as in Foster et al. (2023); performance on additional classes can be found in the appendix (10.3). We perform a hyper-parameter search across a single forget class/sub-class, then use these parameters for all classes. This is more realistic, as it cannot be known *a priori* what future forget sets may be presented to the method. The reported $\mathcal{D}_r$ accuracy refers to accuracy over a test set of samples from the classes in $\mathcal{D}_r$.

Unlearning scenarios: Typically the three unlearning scenarios are: i) Full-class forgetting, where a full class from the dataset must be unlearned, ii) Sub-class forgetting, where a related subset from a class (e.g. all rockets from class vehicle) is forgotten, and iii) Random forgetting, where a subset is sampled uniformly from the entire training distribution. We evaluate our method in all three scenarios.

Comparison methods: We compare JiT against the following methods: i) *Baseline* (BSLN): that has not been unlearnt, ii) *Retrain* (RTRN): trained on only the retain data, iii) *Finetune* (FNTN): , where the model is fine-tuned on $\mathcal{D}_r$ for 5 epochs, iv) *Selective Synaptic Dampening* SSD (Foster et al., 2023), v) GKT (Chundawat et al., 2023b), vi) EMMN (Chundawat et al., 2023b), vii) SCRUB (Kurmanji et al., 2023), viii) *Bad Teacher* (BT) (Chundawat et al., 2023a) ix) *Amnesiac* (AMN) (Graves et al., 2021), x) UNSIR (Tarun et al., 2023), xi) *Boundary Shrinking* (BDSH) (Chen et al., 2023). Since GKT, EMMN, and UNSIR are theoretically limited to forgetting just a full-class, these cannot be evaluated in sub-class or random scenarios. Finally, we note that due to VRAM constraints, we could not benchmark SCRUB on ViT.

Datasets: As with previous work, we benchmark JiT on a range of image classification benchmarks. We make use of the CIFAR suite (Krizhevsky and Hinton, 2010), and the Pins Facial Recognition dataset (Burak, 2020), which consists of $17,534$ images of 105 celebrity faces.

Models: We evaluate methods on Vision Transformer (ViT) (Dosovitskiy et al., 2021) and VGG11 (Simonyan and Zisserman, 2014), trained on an NVidia RTX 4090 using Stochastic Gradient Descent with an initial learning rate of 0.1, and the OneCycle learning rate scheduler (Smith and Topin, 2019). Additionally, we compare the performance of JiT to BDSH on a ViT-L ($\sim 300m$ parameters) trained on the ILSVRC Imagenet dataset to demonstrate our method can scale to larger problem spaces.

Evaluation metrics: We evaluate model performance according to four key metrics: i) $\mathcal{D}_r$ accuracy, ii)

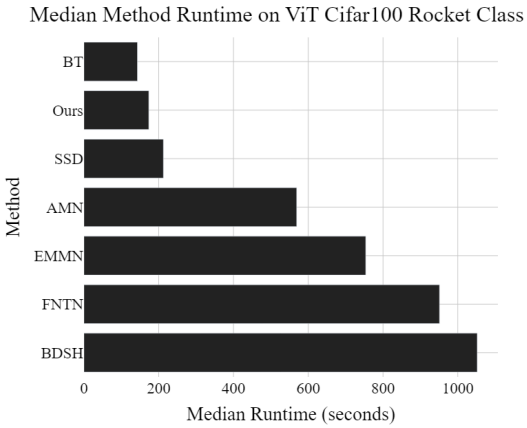

Figure 5: Median method runtime for ViT full-class forgetting on class rocket in seconds. For visual clarity we exclude GKT ($\sim$ 3000 seconds).

Table 1: VGG Full-class unlearning performance on PinsFaceRecognition class 1

Method	$\mathcal{D}_r$ Acc.	$\mathcal{D}_f$ Acc.	MIA	ZS
BSLN	94.0±0.0	93.9±0.0	13.82±0.0	×
RTRN	100.0±0.0	0.0±0.0	2.6±0.8	×
FNTN	97.6±0.7	36.9±9.9	4.3±2.7	×
AMN	99.7±0.1	0.0±0.0	1.4±1.33	×
SCRUB	98.8±0.0	97.1±0.0	8.8±0.76	×
SSD	55.8±0.0	0.0±0.0	4.0±0.0	×
BT	93.7±0.3	0.0±0.0	0.0±0.0	×
UNSIR	99.5±0.1	74.4±9.2	13.6±8.9	×
GKT	2.0±0.6	0.0±0.0	23.9±30.3	✓
EMMN	51.0±13.5	69.3±25.7	26.9±17.8	✓
BDSH	93.6±0.4	79.4±0.0	42.4±0.4	✓
Ours	91.4±0.1	1.9±0.2	4.7±0.5	✓

$\mathcal{D}_f$ accuracy, iii) MIA score, and iv) method runtime. For all metrics bar runtime, the objective is not to minimise/maximise them, but rather to be as close to the retrained model as possible. This is important, as performing worse than the retrained model implies insufficient performance, but as noted in Chundawat et al. (2023a), Foster et al. (2023) and Kurmanji et al. (2023), significant deviation from the retrained model (e.g. *over-forgetting*) may leak information about the fact a sample has been forgotten. To remain consistent with existing unlearning literature we use the same logistic regression MIA evaluation as Chundawat et al. (2023a) and Foster et al. (2023).

JiT hyper-parameters: We conduct a hyper-parameter search for η and σ using 250 runs of the TPE search from Optuna (Akiba et al., 2019), for each unlearning scenario. For VGG11, we use the following parameters: full-class unlearning uses $\eta = 0.0003, \sigma = 0.5$, sub-class and random both use $\eta = 0.0003, \sigma = 0.01$. For ViT, the selected parameters are: full-class $\eta = 1.5, \sigma = 0.8$, sub-class $\eta = 0.5, \sigma = 1.5$, and random $\eta = 0.01, \sigma = 0.5$. ViT and VGG use considerably different learning rates, since only a single epoch is used during the unlearning step. If minimising the runtime is a looser constraint, a smaller learning rate can be used for ViT with extra epochs of training. We stress that when selecting hyper-parameters, we selected values that yielded promising results, without rigorously fitting our results to the retrained model.

6 Results

6.1 Benchmark Evaluation

Compute Comparison. Figure 5 shows the runtime of JiT compared to other methods. JiT is very fast, especially in comparison to the other ZS methods, performing more than 5 times faster than Boundary Shrinking. Sufficiently short runtimes are an important desideratum of unlearning, one which JiT empirically satisfies. JiT has a computational complexity of $O(N|\mathcal{D}_f|)$, where N is the number of perturbed samples and $|\mathcal{D}_f|$ is the cardinality of the forget set. Requiring only $\mathcal{D}_f$ and processing each sample just once makes JiT efficient.

Full-class Unlearning. We begin by comparing full class performance to that of the existing ZS methods. As seen in Tables 2 (a), 2 (b), and 1, JiT demonstrates significantly superior performance over GKT and EMMN, and is competitive with Boundary Shrinking. The authors of Chundawat et al. (2023b) note the poor scalability of both EMMN and GKT, which is evident in our results. Failing to scale to large problems or models is a significant barrier, since the value of unlearning is found mostly in large models that are expensive to train, or large datasets that are expensive to store. JiT performance is competitive with Boundary Shrinking despite having a fraction of the compute cost and, even compared to non-ZS SOTA, JiT performs reasonably; dropping only 0.6% retain set performance compared to the baseline on ViT and

outperforming both UNSIR and SSD on the MIA. Our performance also holds for VGG and, when using the same hyper-parameters for the face dataset, JiT generalizes well, outperforming SCRUB, SSD, and UNSIR.

Sub-class Unlearning. Tables 3 (a) and 3 (b) show the performance of JiT on sub-class unlearning for ViT and VGG11. For both, JiT is able to comfortably unlearn. For ViT, it actually *over-forgets*, however this is typically an easier problem to correct, since more conservative values can always be selected. $\mathcal{D}_r$ performance also drops slightly more than usual, which a more conservative parameter set could also correct. For VGG11, however, the method is comfortably amongst the SOTA, outperforming methods that are granted access to the retained data. For ViT, JiT better minimises $\mathcal{D}_f$ compared to BDSH.

Random Unlearning. Tables 4 (a) and 4 (b) show method performance when forgetting 100 samples uniformly distributed across the training set. JiT is able to comfortably rival existing ZS and non-ZS SOTA methods; despite slight over-forgetting, $\mathcal{D}_r$ accuracy is almost unchanged ($\sim 1\%$ for both models). Table 5 (b) shows both ZS methods also generalize to random unlearning on ResNet18, albeit with a larger drop in retain set accuracy.

Additional Ablations. Table 5 (a) validates our method on a larger scale problem, with JiT achieving SOTA performance for ZS methods. As larger pretrained models can be robust to noise, we found larger perturbations were required to induce forgetting. To keep the input in-domain, we apply normalization to the noised image via: $\frac{(x+\xi)}{\sqrt{(1+\sigma^2)}}$. Finally, we take the 50 samples from class Rocket for which the MIA model outputs the highest probability. In other words, the 50 samples that are the "worst-case scenario" and which leak the most information about the training set. Table 6 shows the performance of each method when attempting to unlearn this subset. JiT performs very well here, performing comparatively with all other methods implemented. Interestingly, both SCRUB and BDSH struggle with forgetting the worst-case samples.

7 Discussion

JiT is by far the fastest ZS unlearning method benchmarked, a critical characteristic for satisfying the unlearning task. JiT is competitive with state-of-the-art performance in the ZS unlearning domain, as well as competing with non-ZS methods in the sub-class and random unlearning tasks despite their easier task. The entropy experiments highlight that JiT is able to replicate the output entropy of a retrained model over a forget set, while preserving retain set performance. When compared to existing ZS methods, JiT can be considered a strong baseline. It is fast and performant, and performed acceptably across all benchmarks implemented. If time constraints are ignored, BDSH is more stable and less sensitive to hyper-parameter selection, on account of taking the true gradients of the model with respect to the input, rather than the

Table 2: (a) ViT Full-class unlearning performance on CIFAR-100 class Rocket. (b) VGG11 Full-class unlearning performance on CIFAR-100 class Rocket.

(a)

Method	$\mathcal{D}_r$ Acc.	$\mathcal{D}_f$ Acc.	MIA	ZS
BSLN	88.9 ± 0.0	94.7 ± 0.0	94.4 ± 0.0	×
RTRN	90.1 ± 0.0	0.0 ± 0.0	3.2 ± 0.5	×
FNTN	80.8 ± 1.4	0.6 ± 0.7	19.0 ± 8.7	×
AMN	87.9 ± 0.9	0.0 ± 0.0	1.4 ± 0.9	×
SSD	88.90 ± 0.0	0.0 ± 0.0	1.8 ± 0.0	×
BT	87.5 ± 0.5	4.2 ± 5.2	0.0 ± 0.1	×
UNSIR	88.5 ± 0.4	65.3 ± 9.1	29.1 ± 6.1	×
GKT	1.0±0.6	0.0±0.0	60.0±51.6	✓
EMMN	84.6±0.4	94.3±1.5	93.7±2.2	✓
BDSH	87.6±0.0	0.0±0.0	5.0±0.0	✓
Ours	87.5±0.0	51.9±2.13	4.3±0.38	✓

(b)

Method	$\mathcal{D}_r$ Acc.	$\mathcal{D}_f$ Acc.	MIA	ZS
BSLN	66.3±0.0	77.0±0.0	97.4±0.0	×
RTRN	63.2±0.5	0.0±0.0	10.4±1.1	×
FNTN	59.7±0.4	3.9±3.0	13.2±4.2	×
AMN	64.3±0.4	0.0±0.0	1.8±0.8	×
SCRUB	66.2±0.1	0.0±0.0	8.2±1.7	×
SSD	63.79±0.0	0.0±0.0	8.6±0.0	×
BT	65.5±0.2	0.1±0.3	0.0±0.1	×
UNSIR	64.6±0.4	42.9±14.3	40.7±12.1	×
GKT	2.3±0.2	0.0±0.0	56.2±20.0	✓
EMMN	26.9±7.7	24.3±23.7	58.2±14.5	✓
BDSH	66.2±0.1	13.0±0.0	2.9±0.1	✓
OURS	66.2±0.3	14.2±0.6	2.9±0.3	✓

Table 3: (a) VGG-16 Sub-class unlearning performance on CIFAR-20 sub-class Rocket. (b) ViT Sub-class unlearning performance onCIFAR-20 sub-class Rocket

(a)

Method	$\mathcal{D}_r$ Acc.	$\mathcal{D}_f$ Acc.	MIA	ZS
BSLN	75.3±0.0	79.0±0.0	83.1±0.0	×
RTRN	72.9±0.2	11.5±2.8	14.1±1.3	×
FNTN	65.5±0.7	6.2±3.7	22.3±5.5	×
AMN	73.8±0.2	2.4±2.4	3.0±0.9	×
SCRUB	62.4±28.4	10.1±22.48	16.7±21.7	×
SSD	75.0±0.0	4.2±0.0	11.0±0.0	×
BT	74.9±0.2	48.4±16.9	0.1±0.1	×
UNSIR	74.1±0.2	57.5±10.3	57.4±8.6	×
BDSH	74.4±0.0	17.535±0.0	12.9±0.1	✓
OURS	73.7±0.8	19.3±18.3	11.2±7.8	✓

(b)

Method	D_r Acc.	D_f Acc.	MIA	ZS
BSLN	95.7 ± 0.0	94.5 ± 0.0	80.4 ± 0.0	×
RTRN	94.6 ± 0.1	22.3 ± 8.3	3.4 ± 1.1	×
FNTN	85.7 ± 3.1	6.2 ± 6.0	16.0 ± 2.7	×
AMN	93.5 ± 0.2	0.8 ± 1.7	0.8 ± 0.3	×
SSD	95.1 ± 0.0	5.12 ± 0.0	5.4 ± 0.0	×
BT	93.6 ± 0.3	3.3 ± 2.9	0.0 ± 0.1	×
UNSIR	93.3 ± 0.4	74.9 ± 10.1	27.3 ± 13.8	×
BDSH	95.7±0.0	48.4±0.0	0.1±0.0	✓
Ours	92.2±0.0	0.0±0.0	14.66±8.8	✓

approximation we employ with JiT. However, in practice the poor time complexity of BDSH will likely prove prohibitive when trying to unlearn from internet-scale models, whereas JiT is amongst the fastest methods we benchmarked. Future work could explore the efficacy of using JiT with exact gradients, or a more specialised gradient approximation. JiT has the potential for positive societal impacts, aiding the preservation of individual privacy. However, due to a lack of certification, poor use of JiT could result in organizations believing they have removed the influence of an individual's data when they haven't.

8 Limitations

The appendix provides a sensitivity analysis (10.2). It demonstrates that JiT can be sensitive to hyper-parameter selection. This is a by-product of having no access to $\mathcal{D}_r$ to finetune, and also due to our gradient approximation method. Exact gradients are slower, but may prove more stable. Importantly, we note that better stability can be achieved through gradient clipping, though this came at the cost of method performance.

Since we minimise the gradient over each forget sample independently, we advise caution when using a model with batch normalization; since this changes the model's mapping of single input/output to batch input/output (Gulrajani et al., 2017). This limitation can be mitigated by selecting an alternative choice of model or normalization strategy (e.g. layer norms).

Table 4: (a) VGG11 Random unlearning performance for 100 samples from CIFAR-10. (b)ViT Random unlearning performance for 100 samples from CIFAR-10.

(a)

Method	$\mathcal{D}_r$ Acc.	$\mathcal{D}_f$ Acc.	MIA	ZS
BSLN	87.0±0.0	92.0±3.6	70.1±5.4	×
RTRN	87.7±0.2	91.0±2.5	78.9±3.5	×
FNTN	84.4±0.8	86.4±4.4	70.8±4.7	×
AMN	86.8±0.3	51.3±4.4	13.1±2.9	×
SCRUB	87.7±0.1	92.7±2.9	71.8±5.2	×
SSD	85.6±2.7	90.8±3.7	66.7±5.9	×
BT	86.9±0.2	82.5±4.9	40.8±6.3	×
BDSH	86.9±0.1	92.2±3.4	69.8±5.1	✓
OURS	86.3±0.3	88.7±3.9	64.2±5.2	✓

(b)

Method	$\mathcal{D}_r$ Acc.	$\mathcal{D}_f$ Acc.	MIA	ZS
BSLN	98.9 ± 0.0	100.0 ± 0.0	90.8 ± 3.5	×
RTRN	98.6 ± 0.1	98.8 ± 0.8	91.8 ± 1.8	×
FNTN	97.3 ± 0.3	97.2 ± 1.0	86.1 ± 2.1	×
AMN	97.6 ± 0.3	73.5 ± 5.1	10.4 ± 4.9	×
SSD	98.0 ± 1.6	98.1 ± 2.4	85.5 ± 0.1	×
BT	97.6 ± 0.4	86.7 ± 3.6	33.5 ± 5.6	×
BDSH	98.0±0.29	97.9±1.6	78.8±0.0	✓
Ours	98.0±0.3	98.0±1.5	78.8±4.0	✓

Table 5: (a) Zero-shot methods performance on a ViT-L trained on ILSVRC Imagenet. (b) ResNet18 random unlearning performance on CIFAR10.

(a)

Method	$\mathcal{D}_r$ Acc.	$\mathcal{D}_f$ Acc	MIA
BSLN	86.0	100.0	94.0
BDSH	85.9	60.0	0.0
OURS	83.6	10.0	0.0

(b)

Method	$\mathcal{D}_r$ Acc.	$\mathcal{D}_f$ Acc	MIA
BSLN	90.7	95.3	75.8
RTRN	91.5	94.1	74.2
FNTN	88.0	90.0	74.6
AMN	90.2	59.0	25.2
SSD	88.7	93.6	72.7
BT	90.2	90.0	49.3
BDSH	89.5	91.6	71.1
OURS	85.6	92.8	68.7

JiT, like most SOTA methods, is not certified. The impact of this will vary by application domain, but may preclude its use in especially sensitive areas. Finally, we note that our method is specifically tailored to classification tasks. While it is possible a variation of this approach could work for large generative models, we restrict our focus and our claims to larger classifiers and leave generative applications to future work.

9 Conclusion

Unlearning is an important, challenging problem. The ZS setting is amongst the hardest, requiring delicate treatment of the unlearning process to ensure model performance is protected. In this work, we approached this challenge from an information theoretic perspective, deriving an unlearning algorithm directly from the notion of minimising information gained from a sample. We demonstrate empirically the geometric insights behind why JiT can effectively tackle the ZS unlearning problem, alongside showing experimentally that JiT can reconstruct behaviour analagous to that of a model retrained from scratch. JiT achieves performance competitive with state-of-the-art ZS and non-ZS methods. We evaluate JiT on a range of benchmarks, demonstrating its efficacy in full-class, sub-class, and random unlearning, across multiple models. Future work is needed to establish a stronger theoretical relationship between forgetting and information theory, as well as exploring whether this can be formalized to provide guarantees on forgetting using information theoretic approaches.

Table 6: Method performance when evaluated on a VGG11 against the 50 worst-case samples in class Rocket for CIFAR100.

Method	$\mathcal{D}_r$ Acc.	$\mathcal{D}_f$ Acc	MIA
BSLN	66.6	77.0	100.0
RTRN	64.5	0.0	0.0
FNTN	59.7	0.0	6.0
AMN	65.0	0.0	4.0
SSD	65.9	8.0	0.0
BT	66.0	0.0	0.0
SCRUB	66.5	34.0	0.0
BDSH	66.7	65	100.0
OURS	64.5	0.0	2.0

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 141–159. IEEE, 2021.

Burak. Pinterest face recognition dataset. `kaggle.com/datasets/hereisburak/pins-face-recognition`, 2020. Accessed: 2023-08-09.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015.

Min Chen, Weizhuo Gao, Gaoyang Liu, Kai Peng, and Chen Wang. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7766–7775, 2023.

Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7210–7217, 2023a.

Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Zero-shot machine unlearning. *IEEE Transactions on Information Forensics and Security*, 2023b.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. *arXiv preprint arXiv:2308.07707*, 2023.

Sara Fridovich-Keil, Raphael Gontijo Lopes, and Rebecca Roelofs. Spectral bias in practice: The role of function frequency in generalization. *Advances in Neural Information Processing Systems*, 35:7368–7382, 2022.

Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32, 2019.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9304–9312, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11516–11524, 2021.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. *Advances in Neural Information Processing Systems*, 34:16319–16330, 2021.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Jenny E Jeong and Peng Qiu. Quantifying the relative importance of experimental data points in parameter estimation. *BMC systems biology*, 12:1–14, 2018.

Alex Krizhevsky and Geoff Hinton. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7):1–9, 2010.

Meghdad Kurmanji, Peter Triantafillou, and Eleni Triantafillou. Towards unbounded machine unlearning. *arXiv preprint arXiv:2302.09880*, 2023.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. *arXiv preprint arXiv:2403.03218*, 2024.

Dennis V Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

Ronak Mehta, Sourav Pal, Vikas Singh, and Sathya N Ravi. Deep unlearning via randomized conditionally independent hessians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10422–10431, 2022.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory*, pages 931–962. PMLR, 2021.

J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086, 2021.

Vedant Shah, Frederik Träuble, Ashish Malik, Hugo Larochelle, Michael Mozer, Sanjeev Arora, Yoshua Bengio, and Anirudh Goyal. Unlearning via sparse representations. *arXiv preprint arXiv:2311.15268*, 2023.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27 (3):379–423, 1948.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019.

Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

Eleni Triantafillou, Fabian Pedregosa, Jamie Hayes, Peter Kairouz, Isabelle Guyon, Meghdad Kurmanji, Gintare Karolina Dziugaite, Peter Triantafillou, Kairan Zhao, Lisheng Sun Hosoya, Julio C. S. Jacques Junior, Vincent Dumoulin, Ioannis Mitliagkas, Sergio Escalera, Jun Wan, Sohier Dane, Maggie Demkin, and Walter Reade. Neurips 2023 - machine unlearning, 2023. URL `https://kaggle.com/competitions/neurips-2023-machine-unlearning`.

Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. *arXiv preprint arXiv:2108.11577*, 2021.

Robert F Woolson. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, pages 1–3, 2007.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.

10 Appendix

10.1 Method Algorithm

Algorithm 1 JiT Unlearning

Input: The trained model $f_\theta(\,\cdot\,)$ and the forget set $\mathcal{S}$.
Parameter: η, σ, N
Output: $f_{\hat{\theta}}(\,\cdot\,) = \mathcal{U}_{\mathcal{S}}\left(f_\theta(\,\cdot\,)\right)$

```
Initialise optim(θ, lr = η)
for x in S do
    ℓ = 0
    for i in range(N) do
        x' = x + ξ for ξ ~ N(0, σ^2)
        k = ||f_θ(x) − f_θ(x')||_2 / ||ξ||_2
        ℓ = ℓ + k
    end for
end for
ℓ = ℓ/N
θ̂ ← optim{∇_θ ℓ}
return f_θ̂(·)
```

Figure 6: Pseudocode algorithm for JiT Unlearning

10.2 Sensitivity Analysis

Figures 7 and 8 show the sensitivity of key metrics to changes in the hyper-parameters η and σ. In general, the approach is robust to small perturbations of the learning rate, but naturally as it increases by orders

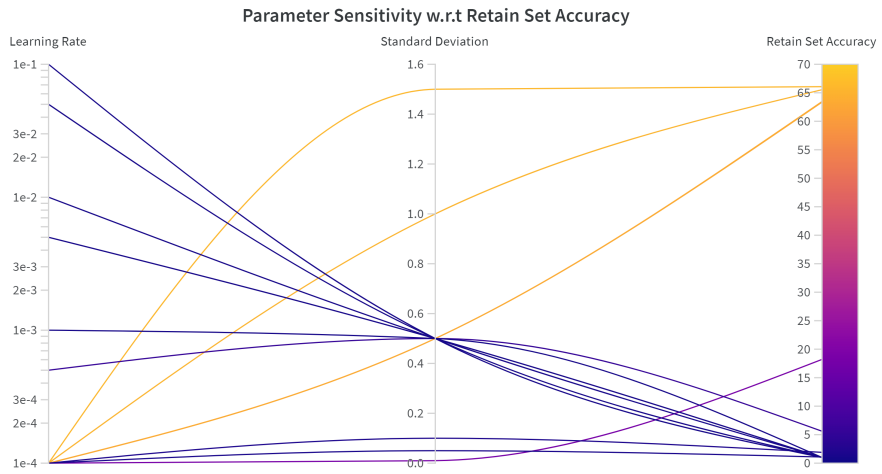

Figure 7: Plot of the $\mathcal{D}_r$ sensitivity to change in hyper-parameters for VGG11 full-class forgetting.

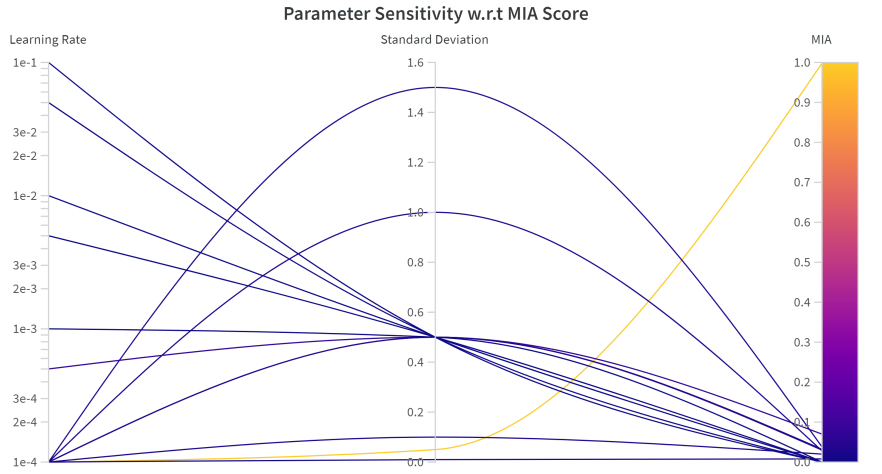

Figure 8: Plot of the MIA sensitivity to change in hyper-parameters for VGG11 full-class forgetting.

of magnitude, performance varies significantly. For σ, in general increasing the noise actually reduced the forgetting/increased $\mathcal{D}_r$ accuracy. This is because VGG was so sensitivity to noise, and so small values of σ did little to reduce the divergence of model output, and thus for $\sigma < 1$ the loss increases due to the division in the loss term. For models that are robust to additive noise, the relationship between σ and performance is often parabolic.

Choice of how many perturbed variants is simple. The more samples the more stable the forgetting, and the only trade-off is compute time.

10.3 Class Breakdown of Method Performance

Below are the forget-class breakdowns of each method's performance for full-class and sub-class unlearning. For each unlearning scenario, the same parameters are used across all classes; this can lead to significant variance in method performance.

Table 7: ViT CIFAR-100 full-class unlearning breakdown

Method	Forget Class	$\mathcal{D}_r$	$\mathcal{D}_f$	MIA	Method Runtime	ZS
Baseline	baby	88.9 ± 0.0	90.2 ± 0.0	75.6 ± 0.0	459.7 ± 83.0	×
Baseline	lamp	88.8 ± 0.0	97.2 ± 0.0	95.6 ± 0.0	504.1 ± 131.7	×
Baseline	mushroom	88.9 ± 0.0	94.9 ± 0.0	92.8 ± 0.0	474.6 ± 60.2	×
Baseline	rocket	88.9 ± 0.0	94.7 ± 0.0	94.4 ± 0.0	470.3 ± 85.7	×
Baseline	sea	88.9 ± 0.0	90.5 ± 0.0	80.4 ± 0.0	549.3 ± 81.8	×
Retrain	baby	90.3 ± 0.1	0.0 ± 0.0	21.5 ± 2.8	2029.1 ± 169.7	×
Retrain	lamp	90.1 ± 0.2	0.0 ± 0.0	2.3 ± 0.7	2115.0 ± 196.3	×
Retrain	mushroom	90.0 ± 0.2	0.0 ± 0.0	0.7 ± 0.4	1948.8 ± 123.1	×
Retrain	rocket	90.1 ± 0.1	0.0 ± 0.0	3.2 ± 0.5	1939.7 ± 109.7	×
Retrain	sea	90.3 ± 0.2	0.0 ± 0.0	8.4 ± 2.2	2176.1 ± 92.8	×
Finetune	baby	80.7 ± 1.4	0.0 ± 0.0	26.8 ± 12.7	1461.8 ± 96.2	×
Finetune	lamp	80.2 ± 1.5	0.4 ± 0.9	11.8 ± 4.3	1537.4 ± 133.2	×
Finetune	mushroom	81.1 ± 0.8	2.3 ± 2.4	7.1 ± 1.9	1421.7 ± 80.5	×
Finetune	rocket	80.8 ± 1.4	0.5 ± 0.7	19.0 ± 8.7	1404.9 ± 82.8	×
Finetune	sea	80.8 ± 1.4	0.0 ± 0.0	22.0 ± 7.1	1525.0 ± 147.3	×
Amnesiac	baby	88.4 ± 0.7	0.0 ± 0.0	1.8 ± 0.3	1132.0 ± 147.1	×
Amnesiac	lamp	88.4 ± 0.6	0.0 ± 0.0	2.7 ± 0.4	1146.5 ± 97.5	×
Amnesiac	mushroom	88.3 ± 0.7	0.0 ± 0.0	0.5 ± 0.2	1071.9 ± 98.1	×
Amnesiac	rocket	87.9 ± 0.9	0.0 ± 0.0	1.0 ± 0.6	1029.7 ± 70.8	×
Amnesiac	sea	88.3 ± 0.3	0.0 ± 0.0	0.8 ± 0.2	1176.2 ± 136.7	×
SSD	baby	88.59 ± 0.0	0.0 ± 0.0	0.60 ± 0.0	685.40 ± 97.35	×
SSD	lamp	89.06 ± 0.0	36.89 ± 0.0	0.40 ± 0.0	741.23 ± 170.57	×
SSD	mushroom	88.82 ± 0.0	0.0 ± 0.0	3.80 ± 0.0	650.93 ± 76.43	×
SSD	rocket	88.90 ± 0.0	0.0 ± 0.0	1.80 ± 0.0	655.64 ± 65.36	×
SSD	sea	87.95 ± 0.0	0.0 ± 0.0	3.20 ± 0.0	767.35 ± 177.77	×
Teacher	baby	87.5 ± 0.4	23.8 ± 22.5	0.0 ± 0.0	610.0 ± 98.1	×
Teacher	lamp	87.5 ± 0.4	25.2 ± 12.5	0.1 ± 0.2	668.6 ± 135.1	×
Teacher	mushroom	87.4 ± 0.4	12.8 ± 5.9	0.0 ± 0.1	602.9 ± 33.2	×
Teacher	rocket	87.5 ± 0.5	4.2 ± 5.2	0.0 ± 0.1	602.6 ± 63.2	×
Teacher	sea	87.7 ± 0.2	51.1 ± 17.4	0.0 ± 0.0	661.8 ± 117.1	×
UNSIR	baby	88.8 ± 0.4	2.0 ± 1.2	14.3 ± 6.1	954.6 ± 110.3	×
UNSIR	lamp	88.5 ± 0.4	70.9 ± 4.4	29.4 ± 4.8	1002.2 ± 135.1	×
UNSIR	mushroom	88.4 ± 0.6	83.9 ± 2.9	21.3 ± 2.7	891.8 ± 69.9	×
UNSIR	rocket	88.5 ± 0.4	65.3 ± 9.1	29.1 ± 6.1	868.8 ± 47.9	×
UNSIR	sea	88.8 ± 0.2	13.9 ± 6.2	9.1 ± 4.7	986.1 ± 149.6	×
GKT	baby	1.00 ± 0.07	0.00 ± 0.00	70.00 ± 48.30	1074.86 ± 791.54	✓
GKT	lamp	1.01 ± 0.06	0.00 ± 0.00	60.00 ± 51.64	1924.50 ± 1269.59	✓
GKT	mushroom	1.01 ± 0.06	0.00 ± 0.00	60.00 ± 51.64	1793.25 ± 1100.70	✓
GKT	rocket	1.00 ± 0.06	0.00 ± 0.00	60.00 ± 51.64	1943.99 ± 1227.35	✓
GKT	sea	1.01 ± 0.06	0.00 ± 0.00	70.00 ± 48.30	1788.32 ± 1150.91	✓
EMMN	baby	84.63 ± 0.30	87.86 ± 2.98	74.82 ± 6.54	1064.73 ± 202.59	✓
EMMN	lamp	84.99 ± 0.37	93.03 ± 2.04	87.58 ± 2.95	1100.77 ± 212.08	✓
EMMN	mushroom	84.65 ± 0.33	93.11 ± 1.84	91.14 ± 2.23	1068.89 ± 132.31	✓
EMMN	rocket	84.62 ± 0.40	94.33 ± 1.53	93.68 ± 2.16	1115.30 ± 179.26	✓
EMMN	sea	84.68 ± 0.33	89.47 ± 4.26	74.64 ± 7.44	1075.82 ± 216.62	✓
BDSH	baby	83.87 ± 0.0	0.0 ± 0.0	8.6 ± 0.0	1388.14 ± 310.5	✓
BDSH	lamp	86.94 ± 0.0	2.0 ± 0.0	28.44±0.0	1386.44 ± 304.9	✓
BDSH	mushroom	87.8 ± 0.0	0.0 ± 0.0	0.8 ± 0.0	1398.19 ± 305.3	✓
BDSH	rocket	87.62 ± 0.0	0.0 ± 0.0	5.0 ± 0.0	1396.85 ± 306.4	✓
BDSAH	sea	87.34 ± 0.0	3.0 ± 0.0	0.8 ± 0.0	1414,41 ± 146.2	✓
Ours	baby	87.16 ± 0.03	38.70 ± 0.95	0.40 ± 0.00	607.94 ± 130.32	✓
Ours	lamp	88.39 ± 0.01	93.90 ± 0.32	49.86 ± 2.09	596.99 ± 143.57	✓
Ours	mushroom	87.70 ± 0.03	77.90 ± 0.57	4.78 ± 0.37	656.20 ± 141.64	✓
Ours	rocket	87.46 ± 0.02	51.90 ± 2.13	4.26 ± 0.38	629.38 ± 116.30	✓
Ours	sea	83.78 ± 0.08	24.00 ± 1.15	16.32 ± 0.36	596.28 ± 151.89	✓

Table 8: ViT CIFAR-20 sub-class unlearning breakdown

Method	Forget Class	$\mathcal{D}_r$	$\mathcal{D}_f$	MIA	Method Runtime	ZS
Baseline	baby	95.7 ± 0.0	96.4 ± 0.0	91.6 ± 0.0	443.0 ± 48.7	×
Baseline	lamp	95.8 ± 0.0	89.6 ± 0.0	81.0 ± 0.0	419.7 ± 25.2	×
Baseline	mushroom	95.7 ± 0.0	97.0 ± 0.0	77.8 ± 0.0	525.7 ± 124.3	×
Baseline	rocket	95.7 ± 0.0	94.5 ± 0.0	80.4 ± 0.0	535.0 ± 75.3	×
Baseline	sea	95.7 ± 0.0	99.2 ± 0.0	88.4 ± 0.0	475.5 ± 120.3	×
Retrain	baby	94.5 ± 0.2	93.2 ± 1.1	77.4 ± 3.4	2148.7 ± 112.5	×
Retrain	lamp	94.7 ± 0.1	34.5 ± 8.6	5.6 ± 1.6	2149.3 ± 124.4	×
Retrain	mushroom	94.6 ± 0.1	26.6 ± 6.4	2.3 ± 0.5	2161.5 ± 81.6	×
Retrain	rocket	94.6 ± 0.1	22.3 ± 8.3	3.4 ± 1.1	2168.3 ± 118.4	×
Retrain	sea	94.6 ± 0.2	95.1 ± 0.8	66.0 ± 3.8	2142.0 ± 85.9	×
Finetune	baby	87.6 ± 0.8	85.4 ± 4.5	66.6 ± 7.1	1426.9 ± 72.7	×
Finetune	lamp	87.7 ± 0.5	16.9 ± 10.4	14.7 ± 3.9	1424.9 ± 71.4	×
Finetune	mushroom	87.4 ± 0.9	15.7 ± 12.1	9.2 ± 4.1	1442.4 ± 113.2	×
Finetune	rocket	85.7 ± 3.1	6.2 ± 6.0	16.0 ± 2.7	1436.6 ± 117.9	×
Finetune	sea	87.6 ± 1.6	89.2 ± 4.2	65.0 ± 12.9	1512.9 ± 129.8	×
Amnesiac	baby	93.3 ± 0.3	38.8 ± 7.4	0.9 ± 0.7	1025.0 ± 35.2	×
Amnesiac	lamp	93.7 ± 0.5	0.6 ± 1.5	2.0 ± 1.0	1009.0 ± 37.7	×
Amnesiac	mushroom	93.4 ± 0.5	0.2 ± 0.4	1.5 ± 0.5	1131.8 ± 137.6	×
Amnesiac	rocket	93.5 ± 0.2	0.8 ± 1.7	0.8 ± 0.3	1186.6 ± 107.5	×
Amnesiac	sea	93.3 ± 0.2	21.4 ± 8.5	0.4 ± 0.2	1070.4 ± 138.1	×
SSD	baby	95.54 ± 0.0	94.10 ± 0.0	77.20 ± 0.0	736.00 ± 12.01	×
SSD	lamp	95.54 ± 0.0	14.58 ± 0.0	3.2 ± 0.0	728.98 ± 73.07	×
SSD	mushroom	95.51 ± 0.0	6.68 ± 0.0	0.40 ± 0.0	718.83 ± 73.41	×
SSD	rocket	95.13 ± 0.0	5.12 ± 0.0	5.40 ± 0.0	699.33 ± 72.47	×
SSD	sea	95.57 ± 0.0	97.05 ± 0.0	82.20 ± 0.0	645.74 ± 53.38	×
Teacher	baby	93.0 ± 0.5	46.7 ± 17.9	0.0 ± 0.1	553.9 ± 65.9	×
Teacher	lamp	93.6 ± 0.7	8.2 ± 7.1	0.1 ± 0.2	558.1 ± 63.9	×
Teacher	mushroom	93.6 ± 0.4	13.0 ± 9.1	0.0 ± 0.0	620.1 ± 111.7	×
Teacher	rocket	93.6 ± 0.3	3.3 ± 2.9	0.0 ± 0.1	631.9 ± 115.2	×
Teacher	sea	93.6 ± 0.3	26.0 ± 14.0	0.2 ± 0.1	586.7 ± 89.7	×
UNSIR	baby	93.2 ± 0.3	94.5 ± 0.8	88.0 ± 3.1	871.1 ± 60.7	×
UNSIR	lamp	93.4 ± 0.5	76.5 ± 5.2	36.5 ± 11.7	899.8 ± 72.0	×
UNSIR	mushroom	93.1 ± 0.6	79.8 ± 7.6	19.0 ± 7.4	925.7 ± 117.5	×
UNSIR	rocket	93.3 ± 0.4	74.9 ± 10.1	27.3 ± 13.8	983.1 ± 143.2	×
UNSIR	sea	93.3 ± 0.3	94.3 ± 2.3	77.0 ± 7.2	1024.5 ± 144.8	×
BDSH	baby	95.36 ± 0.0	93.32 ± 0.0	18.8 ± 0.0	1163 ± 49.76	✓
BDSH	lamp	95.76 ± 0.0	89.58 ± 0.0	80.8 ± 0.0	1152.48 ± 53.9	✓
BDSH	mushroom	95.72 ± 0.0	88.37 ± 0.0	2.6 ± 0.0	1087.87 ± 200.2	✓
BDSH	rocket	95.66 ± 0.0	48.44 ± 0.0	1.4 ± 0.0	1087.15 ± 212.6	✓
BDSH	sea	95.09 ±0.0	78.91±0.0	4.6 ± 0.0	1101.91 ± 215.9	✓
Ours	baby	87.40 ± 0.02	0.00 ± 0.00	0.80 ± 0.00	532.20 ± 112.52	✓
Ours	lamp	90.20 ± 0.02	0.00 ± 0.00	22.86 ± 0.19	560.02 ± 58.54	✓
Ours	mushroom	93.73 ± 0.01	0.00 ± 0.00	1.40 ± 0.00	563.76 ± 69.07	✓
Ours	rocket	92.15 ± 0.01	0.00 ± 0.00	14.66 ± 0.10	535.14 ± 105.65	✓
Ours	sea	87.39 ± 0.02	0.00 ± 0.00	3.40 ± 0.00	542.19 ± 87.80	✓

Table 9: VGG11 CIFAR-100 class unlearning breakdown

Method	Forget Class	$\mathcal{D}_r$	$\mathcal{D}_f$	MIA	Method Runtime	ZS
Baseline	baby	66.88 ± 0.00	52.00 ± 0.00	88.20 ± 0.00	74.90 ± 0.60	×
Baseline	lamp	66.79 ± 0.00	61.00 ± 0.00	96.80 ± 0.00	75.10 ± 0.90	×
Baseline	mushroom	66.67 ± 0.00	73.00 ± 0.00	97.40 ± 0.00	74.10 ± 1.10	×
Baseline	rocket	66.63 ± 0.00	77.00 ± 0.00	97.40 ± 0.00	75.00 ± 1.00	×
Baseline	sea	66.65 ± 0.00	75.00 ± 0.00	82.60 ± 0.00	75.50 ± 1.10	×
Retrain	baby	63.53 ± 0.47	0.00 ± 0.00	12.20 ± 1.90	1100.50 ± 2.60	×
Retrain	lamp	63.42 ± 0.45	0.00 ± 0.00	9.40 ± 1.20	1099.40 ± 2.10	×
Retrain	mushroom	63.47 ± 0.25	0.00 ± 0.00	6.60 ± 1.00	1100.00 ± 2.10	×
Retrain	rocket	63.21 ± 0.54	0.00 ± 0.00	10.40 ± 1.10	1100.00 ± 3.00	×
Retrain	sea	63.28 ± 0.34	0.00 ± 0.00	18.40 ± 1.00	1100.30 ± 2.40	×
Finetune	baby	60.02 ± 0.64	0.00 ± 0.00	26.50 ± 5.10	102.80 ± 0.70	×
Finetune	lamp	59.97 ± 0.60	1.80 ± 1.20	18.40 ± 3.20	102.80 ± 1.50	×
Finetune	mushroom	60.36 ± 0.58	1.40 ± 1.30	12.90 ± 3.00	102.40 ± 1.20	×
Finetune	rocket	59.70 ± 0.43	3.90 ± 3.00	13.20 ± 4.20	103.20 ± 1.40	×
Finetune	sea	59.73 ± 0.61	0.00 ± 0.00	28.00 ± 7.10	103.40 ± 0.60	×
Amnesiac	baby	64.70 ± 0.36	0.00 ± 0.00	4.30 ± 1.10	99.50 ± 3.40	×
Amnesiac	lamp	64.58 ± 0.37	0.00 ± 0.00	4.80 ± 1.20	99.90 ± 3.00	×
Amnesiac	mushroom	64.49 ± 0.52	0.00 ± 0.00	3.10 ± 1.10	100.80 ± 3.30	×
Amnesiac	rocket	64.34 ± 0.38	0.00 ± 0.00	1.80 ± 0.80	99.70 ± 2.80	×
Amnesiac	sea	64.39 ± 0.40	0.00 ± 0.00	1.20 ± 0.30	100.20 ± 3.50	×
SCRUB	baby	67.05 ± 0.11	0.00 ± 0.00	9.30 ± 1.50	109.10 ± 2.60	×
SCRUB	lamp	66.81 ± 0.10	0.00 ± 0.00	8.70 ± 0.90	108.50 ± 3.50	×
SCRUB	mushroom	67.05 ± 0.08	0.00 ± 0.00	9.40 ± 0.50	107.60 ± 2.80	×
SCRUB	rocket	66.19 ± 0.14	0.00 ± 0.00	8.20 ± 1.70	108.70 ± 2.50	×
SCRUB	sea	66.71 ± 0.12	0.00 ± 0.00	6.30 ± 2.40	107.70 ± 2.70	×
SSD	baby	52.68 ± 0.01	0.00 ± 0.00	7.40 ± 0.00	81.30 ± 1.30	×
SSD	lamp	65.44 ± 0.00	0.00 ± 0.00	5.80 ± 0.00	81.60 ± 0.80	×
SSD	mushroom	62.19 ± 0.00	0.00 ± 0.00	14.60 ± 0.00	81.20 ± 1.00	×
SSD	rocket	63.79 ± 0.01	0.00 ± 0.00	8.60 ± 0.00	81.50 ± 0.90	×
SSD	sea	32.75 ± 0.01	0.00 ± 0.00	7.00 ± 0.00	81.30 ± 1.20	×
Teacher	baby	66.05 ± 0.33	1.00 ± 1.10	0.10 ± 0.10	79.60 ± 0.80	×
Teacher	lamp	65.96 ± 0.19	0.40 ± 1.00	0.00 ± 0.00	79.10 ± 1.30	×
Teacher	mushroom	65.67 ± 0.27	0.90 ± 1.50	0.10 ± 0.10	79.60 ± 1.40	×
Teacher	rocket	65.51 ± 0.24	0.10 ± 0.30	0.00 ± 0.10	79.10 ± 1.00	×
Teacher	sea	65.57 ± 0.24	5.20 ± 4.60	0.00 ± 0.00	86.70 ± 24.00	×
UNSIR	baby	64.88 ± 0.43	3.80 ± 2.30	29.40 ± 11.00	111.10 ± 2.00	×
UNSIR	lamp	64.72 ± 0.35	25.90 ± 6.10	17.00 ± 5.90	111.60 ± 2.50	×
UNSIR	mushroom	64.79 ± 0.27	21.10 ± 10.70	11.30 ± 5.20	111.30 ± 2.90	×
UNSIR	rocket	64.58 ± 0.39	42.90 ± 14.30	40.70 ± 12.10	112.10 ± 3.50	×
UNSIR	sea	64.51 ± 0.33	13.90 ± 7.50	22.90 ± 4.50	111.30 ± 2.90	×
GKT	baby	2.31 ± 0.27	0.00 ± 0.00	47.90 ± 26.50	634.30 ± 9.20	✓
GKT	lamp	2.42 ± 0.37	0.00 ± 0.00	45.30 ± 23.70	630.80 ± 8.60	✓
GKT	mushroom	2.23 ± 0.29	0.00 ± 0.00	47.40 ± 11.80	628.90 ± 6.00	✓
GKT	rocket	2.31 ± 0.24	0.00 ± 0.00	56.20 ± 20.00	629.00 ± 6.90	✓
GKT	sea	2.42 ± 0.39	0.00 ± 0.00	60.50 ± 28.60	632.00 ± 5.60	✓
EMMN	baby	30.11 ± 8.56	7.80 ± 9.20	54.00 ± 11.20	274.70 ± 4.40	✓
EMMN	lamp	33.02 ± 9.02	21.50 ± 17.00	54.70 ± 14.60	276.00 ± 4.60	✓
EMMN	mushroom	31.87 ± 11.81	13.50 ± 12.40	53.40 ± 12.80	275.40 ± 4.60	✓
EMMN	rocket	26.91 ± 7.74	24.30 ± 23.70	58.20 ± 14.50	274.80 ± 3.90	✓
EMMN	sea	30.28 ± 8.81	33.30 ± 21.40	69.10 ± 16.00	275.40 ± 4.70	✓
BDSH	baby	66.88 ± 0.0	16.9 ± 0.3	4.3 ± 0.0	85.16 ± 1.1	✓
BDSH	lamp	66.31 ± 0.0	16.7 ± 0.5	14.68 ± 0.0	85.4 ± 2.2	✓
BDSH	mushroom	66.83 ± 0.0	21.0 ± 0.0	12.64 ± 0.0	85.27 ± 1.8	✓
BDSH	rocket	66,17 ± 0.0	13.0 ± 0.0	2.9 ± 0.0	85.01 ± 1.6	✓
Ours	baby	67.06 ± 0.04	11.10 ± 1.40	9.90 ± 0.30	78.70 ± 2.40	✓

Table 10: VGG11 face unlearning class breakdown

Method	Forget Class	$\mathcal{D}_r$	$\mathcal{D}_f$	MIA	Method Runtime	ZS
Baseline	1	93.95 ± 0.00	93.88 ± 0.00	13.82 ± 0.00	72.93 ± 0.52	×
Baseline	10	93.99 ± 0.00	95.90 ± 0.00	11.48 ± 0.00	72.92 ± 0.84	×
Baseline	20	94.06 ± 0.00	84.48 ± 0.00	65.19 ± 0.00	72.82 ± 0.98	×
Baseline	30	93.93 ± 0.00	97.46 ± 0.00	11.17 ± 0.00	72.64 ± 0.85	×
Baseline	40	94.01 ± 0.00	92.31 ± 0.00	15.38 ± 0.00	72.66 ± 0.62	×
Retrain	1	100.00 ± 0.00	0.00 ± 0.00	2.63 ± 0.76	491.93 ± 2.02	×
Retrain	10	100.00 ± 0.00	0.00 ± 0.00	0.74 ± 0.47	494.31 ± 1.83	×
Retrain	20	100.00 ± 0.00	0.00 ± 0.00	2.49 ± 1.02	492.70 ± 2.20	×
Retrain	30	100.00 ± 0.00	0.00 ± 0.00	5.98 ± 1.15	491.86 ± 2.70	×
Retrain	40	100.00 ± 0.00	0.00 ± 0.00	4.79 ± 1.52	494.67 ± 2.37	×
Finetune	1	97.61 ± 0.67	36.87 ± 9.86	4.28 ± 2.69	84.67 ± 0.88	×
Finetune	10	95.58 ± 2.93	43.77 ± 21.26	8.61 ± 6.74	84.50 ± 0.51	×
Finetune	20	98.08 ± 0.48	25.10 ± 7.26	5.41 ± 1.84	84.76 ± 0.74	×
Finetune	30	97.92 ± 0.82	23.78 ± 9.66	6.48 ± 3.74	84.81 ± 0.89	×
Finetune	40	97.65 ± 0.57	16.58 ± 7.11	6.32 ± 3.23	84.51 ± 0.77	×
Amnesiac	1	99.72 ± 0.13	0.00 ± 0.00	1.38 ± 1.33	82.19 ± 1.10	×
Amnesiac	10	99.67 ± 0.14	0.00 ± 0.00	0.98 ± 0.75	82.19 ± 0.61	×
Amnesiac	20	99.70 ± 0.12	0.00 ± 0.00	1.49 ± 0.64	82.14 ± 0.87	×
Amnesiac	30	99.67 ± 0.10	0.00 ± 0.00	1.12 ± 0.65	82.14 ± 0.74	×
Amnesiac	40	99.67 ± 0.17	0.00 ± 0.00	1.28 ± 1.16	81.92 ± 0.70	×
SCRUB	1	98.83 ± 0.02	97.14 ± 0.00	8.75 ± 0.76	85.48 ± 0.85	×
SCRUB	10	98.85 ± 0.01	97.54 ± 0.00	10.49 ± 0.35	85.50 ± 0.79	×
SCRUB	20	98.83 ± 0.03	98.00 ± 0.32	74.20 ± 0.74	85.45 ± 1.13	×
SCRUB	30	98.88 ± 0.03	95.89 ± 0.00	81.84 ± 0.92	85.57 ± 1.03	×
SCRUB	40	98.86 ± 0.02	96.50 ± 0.49	78.03 ± 1.14	84.88 ± 0.75	×
SSD	1	55.77 ± 0.01	0.00 ± 0.00	3.95 ± 0.00	92.03 ± 42.05	×
SSD	10	73.68 ± 0.01	0.00 ± 0.00	2.46 ± 0.00	86.66 ± 27.23	×
SSD	20	0.82 ± 0.00	0.00 ± 0.00	100.00 ± 0.00	78.18 ± 1.46	×
SSD	30	85.96 ± 0.00	0.00 ± 0.00	10.06 ± 0.00	82.12 ± 9.97	×
SSD	40	47.43 ± 0.01	0.00 ± 0.00	4.27 ± 0.00	78.22 ± 1.07	×
Teacher	1	93.70 ± 0.31	0.00 ± 0.00	0.00 ± 0.00	75.02 ± 0.73	×
Teacher	10	93.72 ± 0.27	0.00 ± 0.00	0.00 ± 0.00	74.64 ± 0.83	×
Teacher	20	93.90 ± 0.26	0.00 ± 0.00	0.28 ± 0.39	74.74 ± 0.84	×
Teacher	30	93.46 ± 0.34	0.08 ± 0.16	0.06 ± 0.18	74.53 ± 0.72	×
Teacher	40	93.80 ± 0.24	0.43 ± 0.92	0.09 ± 0.27	75.04 ± 0.78	×
UNSIR	1	99.49 ± 0.12	74.43 ± 9.17	13.62 ± 8.87	87.32 ± 0.45	×
UNSIR	10	99.39 ± 0.34	87.13 ± 4.20	44.84 ± 9.23	87.66 ± 0.64	×
UNSIR	20	99.44 ± 0.14	54.92 ± 15.19	10.33 ± 4.72	86.78 ± 0.86	×
UNSIR	30	99.43 ± 0.14	65.72 ± 11.97	13.24 ± 5.21	87.76 ± 1.04	×
UNSIR	40	99.54 ± 0.11	48.97 ± 23.85	6.92 ± 6.98	87.18 ± 0.80	×
GKT	1	2.02 ± 0.59	0.00 ± 0.00	23.88 ± 30.25	441.46 ± 1.81	✓
GKT	10	2.01 ± 0.57	0.00 ± 0.00	53.44 ± 44.14	439.67 ± 3.07	✓
GKT	20	1.98 ± 0.59	0.00 ± 0.00	28.18 ± 34.79	441.43 ± 2.98	✓
GKT	30	2.15 ± 0.66	0.00 ± 0.00	28.38 ± 40.11	439.63 ± 3.38	✓
GKT	40	2.11 ± 0.63	0.00 ± 0.00	44.87 ± 49.71	441.20 ± 2.28	✓
EMMN	1	50.98 ± 13.53	69.30 ± 25.68	26.91 ± 17.76	283.65 ± 1.33	✓
EMMN	10	51.78 ± 14.34	65.00 ± 21.82	27.79 ± 26.23	283.64 ± 1.87	✓
EMMN	20	41.38 ± 12.94	26.22 ± 17.24	55.75 ± 13.54	282.91 ± 0.70	✓
EMMN	30	45.42 ± 18.22	63.02 ± 27.70	32.91 ± 21.73	283.73 ± 1.17	✓
EMMN	40	46.60 ± 16.99	26.50 ± 12.03	60.94 ± 7.43	282.83 ± 1.26	✓
BDSH	1	93.60 ± 0.0	79.43 ± 0.0	42.4 ± 0.0	91.24 ±1.7	✓
BDSH	10	94.0 ± 0.0	93.44 ± 0.0	79.51 ± 0.0	91.10 ± 0.7	✓
BDSH	20	93.68 ± 0.0	64.17 ± 0.0	32.93 ± 0.0	90.85 ± 1.8	✓
BDSH	30	94.11 ± 0.0	90.23 ± 0.0	65.9 ± 0.0	90.46 ± 3.5	✓
BDSH	40	94.01 ± 0.0	90.60 ± 0.0	73.5 ± 0.0	90.89 ± 1.5	✓

Table 11: VGG11 CIFAR-20 sub-class unlearning breakdown

Method	Forget Class	$\mathcal{D}_r$	$\mathcal{D}_f$	MIA	Method Runtime	ZS
Baseline	baby	75.21 ± 0.00	82.29 ± 0.00	74.80 ± 0.00	76.03 ± 1.48	×
Baseline	lamp	75.45 ± 0.00	56.86 ± 0.00	72.00 ± 0.00	75.29 ± 0.80	×
Baseline	mushroom	75.28 ± 0.00	75.61 ± 0.00	73.40 ± 0.00	75.73 ± 0.84	×
Baseline	rocket	75.26 ± 0.00	78.99 ± 0.00	83.00 ± 0.00	75.39 ± 1.05	×
Baseline	sea	75.09 ± 0.00	92.88 ± 0.00	90.60 ± 0.00	75.02 ± 1.19	×
Retrain	baby	72.60 ± 0.23	67.69 ± 3.00	55.36 ± 1.20	425.76 ± 1.39	×
Retrain	lamp	72.99 ± 0.23	18.01 ± 2.81	17.90 ± 2.03	425.73 ± 1.64	×
Retrain	mushroom	72.98 ± 0.33	8.78 ± 2.23	12.14 ± 1.28	425.62 ± 1.72	×
Retrain	rocket	72.87 ± 0.21	11.46 ± 2.80	14.06 ± 1.30	425.49 ± 1.86	×
Retrain	sea	72.61 ± 0.27	85.53 ± 2.59	66.76 ± 2.29	425.21 ± 1.70	×
Finetune	baby	65.45 ± 0.84	60.45 ± 6.82	53.20 ± 5.89	120.34 ± 1.07	×
Finetune	lamp	65.96 ± 1.55	17.53 ± 5.18	22.56 ± 4.10	121.19 ± 1.31	×
Finetune	mushroom	65.69 ± 1.06	6.13 ± 5.96	18.52 ± 4.15	121.28 ± 0.64	×
Finetune	rocket	65.51 ± 0.72	6.23 ± 3.69	22.26 ± 5.51	120.56 ± 0.87	×
Finetune	sea	64.95 ± 0.76	82.15 ± 4.64	70.98 ± 9.58	120.37 ± 1.12	×
Amnesiac	baby	73.70 ± 0.25	53.55 ± 6.44	4.96 ± 0.90	97.95 ± 1.11	×
Amnesiac	lamp	74.18 ± 0.24	9.20 ± 2.38	7.08 ± 1.62	97.62 ± 0.90	×
Amnesiac	mushroom	74.05 ± 0.24	2.52 ± 1.60	4.28 ± 0.94	97.81 ± 1.00	×
Amnesiac	rocket	73.80 ± 0.19	2.43 ± 2.43	2.98 ± 0.86	98.39 ± 0.85	×
Amnesiac	sea	73.65 ± 0.16	64.00 ± 17.71	1.22 ± 0.53	97.64 ± 1.23	×
SCRUB	baby	75.04 ± 0.26	76.99 ± 2.26	71.00 ± 2.92	130.78 ± 2.81	×
SCRUB	lamp	75.44 ± 0.16	41.22 ± 2.77	40.22 ± 6.66	130.45 ± 2.12	×
SCRUB	mushroom	75.40 ± 0.13	15.52 ± 7.61	11.82 ± 0.85	130.79 ± 2.64	×
SCRUB	rocket	62.44 ± 28.38	10.12 ± 22.48	16.71 ± 21.68	131.41 ± 3.75	×
SCRUB	sea	75.05 ± 0.12	91.18 ± 1.56	83.13 ± 3.26	130.69 ± 2.52	×
SSD	baby	71.97 ± 0.00	0.00 ± 0.00	8.80 ± 0.00	83.78 ± 1.04	×
SSD	lamp	73.98 ± 0.00	3.56 ± 0.00	9.00 ± 0.00	84.39 ± 1.09	×
SSD	mushroom	71.99 ± 0.00	0.00 ± 0.00	6.20 ± 0.00	84.03 ± 1.25	×
SSD	rocket	74.96 ± 0.00	4.17 ± 0.00	11.00 ± 0.00	83.91 ± 0.81	×
SSD	sea	74.80 ± 0.00	69.62 ± 0.00	56.20 ± 0.00	83.54 ± 1.42	×
Teacher	baby	74.87 ± 0.17	77.90 ± 1.25	0.38 ± 0.71	78.74 ± 0.59	×
Teacher	lamp	75.18 ± 0.11	35.55 ± 13.07	0.40 ± 0.38	79.47 ± 1.10	×
Teacher	mushroom	74.99 ± 0.13	27.47 ± 14.49	0.25 ± 0.09	90.58 ± 36.91	×
Teacher	rocket	74.86 ± 0.20	48.36 ± 16.87	0.07 ± 0.10	91.51 ± 38.05	×
Teacher	sea	74.71 ± 0.14	85.28 ± 2.58	0.08 ± 0.14	78.36 ± 0.90	×
UNSIR	baby	73.85 ± 0.20	77.66 ± 2.12	77.60 ± 2.52	110.77 ± 0.90	×
UNSIR	lamp	74.17 ± 0.27	44.52 ± 7.40	55.06 ± 6.21	110.70 ± 0.73	×
UNSIR	mushroom	74.03 ± 0.32	47.14 ± 4.00	41.70 ± 8.15	110.58 ± 1.31	×
UNSIR	rocket	74.08 ± 0.24	57.47 ± 10.25	57.44 ± 8.61	111.27 ± 0.66	×
UNSIR	sea	73.80 ± 0.17	90.75 ± 1.83	85.32 ± 3.83	109.55 ± 1.01	×
BDSH	baby	74.25 ± 0.0	59.12 ± 0.0	31.2 ± 0.0	86.05 ± 1.1	✓
BDSH	lamp	75.3 ± 0.0	36.46 ± 0.0	36.8 ± 0.0	85.24 ± 1.3	✓
BDSH	mushroom	75.06 ±0.0	36.63 ± 0.0	25.74 ± 0.0	86.47 ± 2.5	✓
BDSH	rocket	74.41 ± 0.0	17.54 ± 0.0	12.88 ± 0.0	85.91 ± 1.6	✓
BDSH	sea	72.88 ± 0.0	39.15 ± 0.0	11.84 ± 0.0	85.254 ± 1.2	✓
Ours	baby	73.57 ± 1.28	49.82 ± 19.43	33.32 ± 17.53	79.47 ± 3.63	✓
Ours	lamp	74.05 ± 1.34	32.46 ± 15.76	32.38 ± 19.76	79.73 ± 3.22	✓
Ours	mushroom	74.20 ± 0.69	33.67 ± 12.27	23.52 ± 12.84	79.14 ± 4.00	✓
Ours	rocket	73.68 ± 0.84	19.31 ± 18.31	11.20 ± 7.78	80.14 ± 3.27	✓
Ours	sea	73.07 ± 1.02	37.82 ± 22.10	17.29 ± 20.20	79.04 ± 3.25	✓