LoTUS: Large-Scale Machine Unlearning with a Taste of Uncertainty

Supplementary Material

7. Baselines

The **Gold Stadnard (Gold Std)** model is retrained entirely only on the retain set (D_r), achieving ideal unlearning –when access to the full training set is guaranteed– but at the cost of increased computational complexity. **Finetuning**: The pre-trained model is further trained only on the retain samples (D_r). **NegGrad+** [23]: The pre-trained model continues training on the full training set, but the gradient sign is reversed during backpropagation for the forget samples. **Random Labeling (RndLbl)** [17]: The pre-trained model continues training on the full training set, but the forget samples are randomly reassigned to incorrect classes. **Bad Teacher (BadT)** [9]: A knowledge distillation framework where the student model follows the pre-trained model for retain samples and a randomly initialized model for forget samples. **SCRUB** [23]: A knowledge distillation framework where student model selectively aligns with the pre-trained model by minimizing the KL divergence of their outputs on retain samples while maximizing it for forget samples. **SSD** [14]: Weights that are disproportionately important for forget samples are identified using the Fisher Information Matrix and subsequently dampened. **UNSIR** [34]: A noise matrix, generated based on the forget samples, is fed to the pre-trained model to maximize its error on these samples. **SalUn** [12]: A gradient-based approach that identifies weights to be unlearned and those to keep unchanged, followed by a downstream unlearning method such as Random Labeling. Finetuning, NegGrad+ and Random Labeling are considered simple yet widely used unlearning baselines, whereas the latter five are state-of-the-art approaches.

LoTUS can be integrated with SalUn, with SalUn used to obtain the weight saliency mask for pruning, and LoTUS applied for unlearning. This integration can enhance the unlearning effectiveness of LoTUS. For instance, on ResNet18 with TinyImageNet, it reduces the Avg Gap of LoTUS to 0.1250 (a 25.37% decrease) and the JSD to 0.55 (an 11.29% decrease). However, this comes at the cost of efficiency, with unlearning time increasing to 4.62 minutes (a 162% increase).

8. Reproducibility and Transparency

The code to reproduce the results presented in this paper is publicly available at https://github.com/cspartalis/LoTUS. In addition, all tables and figures have been documented in Jupyter notebooks to enhance transparency. We conducted the experiments using Python 3.11 and CUDA 12.1. For ImageNet1k experiments, we used an NVIDIA RTX A6000 48GB GPU. The remaining experiments were performed on an NVIDIA RTX 4080 16GB GPU. We also used an Intel i7-12700K CPU and 32GB RAM. The hyperparameters used for the baselines are listed in Tab. 5

Baseline	Learning Rate	Weight Decay	Optimizer
Finetune	1×10^{-3}	5×10^{-4}	SGD
Negrad+	1×10^{-3}	5×10^{-4}	SGD
RndLbl	1×10^{-3}	5×10^{-4}	SGD
BadT	1×10^{-4}	0	Adam
SCRUB	5×10^{-4}	5×10^{-4}	Adam
SSD	0.1	0	SGD
UNSIR	1×10^{-3}	0	SGD
SalUn	0.1	5×10^{-4}	SGD

Table 5. Hyperparameters used for baselines. For state-of-the-art methods, they are taken from their respective papers.

9. Extended Analysis on the Accuracy Metrics

Table 6 presents the accuracy scores that define the Avg Gap metric. Beyond outperforming state-of-the-art methods in terms of Avg Gap, LoTUS achieves the best scores in individual accuracy metrics, including MIA accuracy, and accuracy on the retain and test sets. Specifically, it consistently ranks either first or second in these metrics, with first place being the most frequent.

Regarding retention performance (*i.e.*, preserving the utility of the pre-trained model), LoTUS clearly outperforms state-of-the-art, as evidenced by its superior accuracy on the retain and test sets.

However, evaluating unlearning effectiveness, requires a more nuanced analysis. Although LoTUS consistently ranks among the top two methods in MIA accuracy, its accuracy on the forget set exceeds that of the gold standard model (*i.e.*, the model retrained solely on the retain set). This apparent discrepancy may lead to misleading evaluation, suggesting that LoTUS exhibits poor unlearning performance.

However, by incorporating the more sensitive JSD metric –a measure that captures distributional-level differences and provides a more robust evaluation, as detailed in Sec. 5– we conclude that LoTUS achieves effective unlearning. Given this, the increased accuracy on the forget set does not indicate poor unlearning, but rather suggests that LoTUS preserves the utility of the pre-trained model even for the forget samples. The fact that LoTUS achieves the best Avg Gap scores despite the disproportionate penalty imposed by the gap between the accuracy of the unlearned and gold standard models on forget samples

		Metric (↓)	Gold Std	Finetuning	NegGrad+ [23]	RndLbl [17]	Bad Teacher [9]	SCRUB [23]	SSD [14]	UNSIR [34]	SalUn [12]	**LoTUS**
Vision Transformer (ViT)	TinyImageNet	MIA Acc.	0.76$_{\pm0.00}$	0.78$_{\pm0.00}$(0.02)	0.83$_{\pm0.00}$(0.07)	0.50$_{\pm0.43}$(0.26)	0.67$_{\pm0.00}$(0.09)	0.79$_{\pm0.00}$(0.03)	0.79$_{\pm0.00}$(0.03)	0.80$_{\pm0.00}$(0.04)	0.67$_{\pm0.25}$(0.09)	0.76$_{\pm0.00}$**(0.00)**
		Forget Acc.	0.90$_{\pm0.00}$	0.93$_{\pm0.00}$(0.03)	0.97$_{\pm0.00}$(0.07)	0.61$_{\pm0.52}$(0.29)	0.84$_{\pm0.01}$(0.06)	0.96$_{\pm0.00}$(0.06)	0.96$_{\pm0.00}$(0.06)	0.92$_{\pm0.00}$**(0.02)**	0.82$_{\pm0.30}$(0.08)	0.96$_{\pm0.00}$(0.06)
		Retain Acc.	0.96$_{\pm0.00}$	0.98$_{\pm0.00}$(0.02)	0.98$_{\pm0.00}$(0.02)	0.64$_{\pm0.55}$(0.32)	0.87$_{\pm0.01}$(0.09)	0.96$_{\pm0.00}$**(0.00)**	0.96$_{\pm0.00}$**(0.00)**	0.94$_{\pm0.00}$(0.02)	0.86$_{\pm0.32}$(0.10)	0.96$_{\pm0.00}$**(0.00)**
		Test Acc.	0.90$_{\pm0.00}$	0.90$_{\pm0.00}$**(0.00)**	0.90$_{\pm0.00}$**(0.00)**	0.60$_{\pm0.51}$(0.30)	0.83$_{\pm0.01}$(0.07)	0.90$_{\pm0.00}$**(0.00)**	0.90$_{\pm0.00}$**(0.00)**	0.89$_{\pm0.00}$(0.01)	0.80$_{\pm0.30}$(0.10)	0.90$_{\pm0.00}$**(0.00)**
		Avg Gap	0.0000	0.0175	0.0400	0.2925	0.0775	0.0225	0.0225	0.0225	0.0925	**0.0150**
	CIFAR-100	MIA Acc.	0.72$_{\pm0.00}$	0.77$_{\pm0.00}$(0.05)	0.79$_{\pm0.02}$(0.07)	0.74$_{\pm0.01}$(0.02)	0.66$_{\pm0.01}$(0.06)	0.75$_{\pm0.00}$(0.03)	0.75$_{\pm0.01}$(0.03)	0.78$_{\pm0.01}$(0.06)	0.75$_{\pm0.01}$(0.03)	0.71$_{\pm0.02}$**(0.01)**
		Forget Acc.	0.92$_{\pm0.00}$	0.95$_{\pm0.01}$(0.03)	0.97$_{\pm0.02}$(0.05)	0.94$_{\pm0.00}$**(0.02)**	0.90$_{\pm0.01}$**(0.02)**	0.97$_{\pm0.00}$(0.05)	0.96$_{\pm0.01}$(0.04)	0.94$_{\pm0.00}$**(0.02)**	0.94$_{\pm0.01}$(0.02)	0.96$_{\pm0.01}$(0.04)
		Retain Acc.	0.96$_{\pm0.00}$	0.98$_{\pm0.00}$(0.02)	0.97$_{\pm0.02}$(0.01)	0.98$_{\pm0.00}$(0.02)	0.91$_{\pm0.00}$(0.05)	0.96$_{\pm0.00}$**(0.00)**	0.96$_{\pm0.00}$**(0.00)**	0.95$_{\pm0.01}$(0.01)	0.98$_{\pm0.01}$(0.02)	0.96$_{\pm0.00}$**(0.00)**
		Test Acc.	0.91$_{\pm0.01}$	0.92$_{\pm0.01}$(0.01)	0.91$_{\pm0.01}$**(0.00)**	0.92$_{\pm0.00}$(0.01)	0.89$_{\pm0.01}$(0.02)	0.91$_{\pm0.01}$**(0.00)**	0.91$_{\pm0.00}$**(0.00)**	0.90$_{\pm0.01}$(0.01)	0.92$_{\pm0.00}$(0.01)	0.91$_{\pm0.00}$**(0.00)**
		Avg Gap	0.0000	0.0275	0.0325	0.0175	0.0375	0.0200	0.0175	0.0250	0.0200	**0.0125**
	CIFAR-10	MIA Acc.	0.88$_{\pm0.00}$	0.90$_{\pm0.00}$(0.02)	0.91$_{\pm0.00}$(0.03)	0.84$_{\pm0.02}$(0.04)	0.81$_{\pm0.02}$(0.07)	0.88$_{\pm0.00}$**(0.00)**	0.89$_{\pm0.01}$(0.01)	0.90$_{\pm0.00}$(0.02)	0.84$_{\pm0.02}$(0.04)	0.87$_{\pm0.00}$(0.01)
		Forget Acc.	0.99$_{\pm0.00}$	0.99$_{\pm0.00}$**(0.00)**	1.00$_{\pm0.00}$(0.01)	0.99$_{\pm0.00}$**(0.00)**	0.96$_{\pm0.01}$(0.03)	1.00$_{\pm0.00}$(0.01)	1.00$_{\pm0.01}$(0.01)	0.99$_{\pm0.00}$**(0.00)**	0.99$_{\pm0.00}$**(0.00)**	1.00$_{\pm0.00}$(0.01)
		Retain Acc.	1.00$_{\pm0.00}$	1.00$_{\pm0.00}$**(0.00)**	1.00$_{\pm0.00}$**(0.00)**	1.00$_{\pm0.00}$**(0.00)**	0.97$_{\pm0.01}$(0.03)	1.00$_{\pm0.00}$**(0.00)**	1.00$_{\pm0.01}$**(0.00)**	0.99$_{\pm0.00}$(0.01)	1.00$_{\pm0.00}$**(0.00)**	1.00$_{\pm0.00}$**(0.00)**
		Test Acc.	0.98$_{\pm0.01}$	0.99$_{\pm0.01}$(0.01)	0.99$_{\pm0.01}$(0.01)	0.99$_{\pm0.00}$(0.01)	0.96$_{\pm0.01}$(0.02)	0.99$_{\pm0.01}$(0.01)	0.99$_{\pm0.01}$(0.01)	0.99$_{\pm0.00}$(0.01)	0.99$_{\pm0.01}$(0.01)	0.98$_{\pm0.01}$**(0.00)**
		Avg Gap	0.0000	0.0075	0.0125	0.0125	0.0375	**0.0050**	0.0075	0.0100	0.0125	**0.0050**
	MUFAC	MIA Acc.	0.57$_{\pm0.00}$	0.52$_{\pm0.08}$(0.05)	0.52$_{\pm0.07}$(0.05)	0.52$_{\pm0.10}$(0.05)	0.35$_{\pm0.05}$(0.22)	0.59$_{\pm0.01}$**(0.02)**	0.59$_{\pm0.01}$**(0.02)**	0.47$_{\pm0.08}$(0.10)	0.53$_{\pm0.12}$(0.04)	0.59$_{\pm0.01}$**(0.02)**
		Forget Acc.	0.57$_{\pm0.01}$	0.61$_{\pm0.01}$(0.04)	0.66$_{\pm0.02}$(0.09)	0.58$_{\pm0.01}$**(0.01)**	0.43$_{\pm0.06}$(0.14)	0.62$_{\pm0.01}$(0.05)	0.59$_{\pm0.04}$(0.02)	0.58$_{\pm0.01}$**(0.01)**	0.58$_{\pm0.01}$**(0.01)**	0.63$_{\pm0.00}$(0.06)
		Retain Acc.	0.66$_{\pm0.01}$	0.72$_{\pm0.01}$(0.06)	0.71$_{\pm0.01}$(0.05)	0.67$_{\pm0.02}$(0.01)	0.47$_{\pm0.07}$(0.19)	0.66$_{\pm0.01}$**(0.00)**	0.63$_{\pm0.04}$(0.03)	0.72$_{\pm0.01}$(0.06)	0.68$_{\pm0.01}$(0.02)	0.66$_{\pm0.01}$**(0.00)**
		Test Acc.	0.65$_{\pm0.01}$	0.66$_{\pm0.01}$(0.01)	0.65$_{\pm0.03}$**(0.00)**	0.64$_{\pm0.01}$(0.01)	0.50$_{\pm0.08}$(0.15)	0.66$_{\pm0.01}$(0.01)	0.64$_{\pm0.01}$(0.01)	0.63$_{\pm0.02}$(0.02)	0.64$_{\pm0.02}$(0.01)	0.65$_{\pm0.01}$**(0.00)**
		Avg Gap	0.0000	0.0400	0.0475	**0.0200**	0.1750	**0.0200**	**0.0200**	0.0475	**0.0200**	**0.0200**
ResNet18 (RN18)	TinyImageNet	MIA Acc.	0.30$_{\pm0.01}$	0.00$_{\pm0.00}$(0.30)	0.00$_{\pm0.00}$(0.30)	0.00$_{\pm0.00}$(0.30)	0.67$_{\pm0.52}$(0.37)	0.96$_{\pm0.01}$(0.66)	0.95$_{\pm0.01}$(0.65)	0.67$_{\pm0.58}$(0.37)	0.00$_{\pm0.00}$(0.30)	0.53$_{\pm0.01}$**(0.23)**
		Forget Acc.	0.58$_{\pm0.00}$	0.70$_{\pm0.02}$(0.12)	0.73$_{\pm0.02}$(0.15)	0.56$_{\pm0.02}$**(0.02)**	0.49$_{\pm0.04}$(0.09)	1.00$_{\pm0.00}$(0.42)	1.00$_{\pm0.00}$(0.42)	0.68$_{\pm0.03}$(0.10)	0.62$_{\pm0.01}$(0.04)	0.91$_{\pm0.01}$(0.33)
		Retain Acc.	1.00$_{\pm0.00}$	0.73$_{\pm0.02}$(0.27)	0.73$_{\pm0.02}$(0.27)	0.73$_{\pm0.02}$(0.27)	0.55$_{\pm0.04}$(0.45)	1.00$_{\pm0.00}$**(0.00)**	1.00$_{\pm0.00}$**(0.00)**	0.71$_{\pm0.02}$(0.29)	0.71$_{\pm0.02}$(0.29)	0.93$_{\pm0.01}$(0.07)
		Test Acc.	0.89$_{\pm0.01}$	0.40$_{\pm0.01}$(0.19)	0.41$_{\pm0.01}$(0.18)	0.41$_{\pm0.02}$(0.18)	0.36$_{\pm0.03}$(0.23)	0.60$_{\pm0.01}$**(0.01)**	0.60$_{\pm0.00}$**(0.01)**	0.40$_{\pm0.02}$(0.19)	0.41$_{\pm0.01}$(0.18)	0.55$_{\pm0.00}$(0.04)
		Avg Gap	0.0000	0.2200	0.2250	0.1925	0.2850	0.2725	0.2700	0.2375	0.2025	**0.1675**
	CIFAR-100	MIA Acc.	0.49$_{\pm0.01}$	0.00$_{\pm0.00}$(0.49)	0.00$_{\pm0.00}$(0.49)	0.00$_{\pm0.00}$(0.49)	0.33$_{\pm0.58}$(0.16)	0.78$_{\pm0.05}$(0.29)	0.59$_{\pm0.05}$(0.10)	0.00$_{\pm0.00}$(0.49)	0.00$_{\pm0.00}$(0.49)	0.28$_{\pm0.22}$**(0.21)**
		Forget Acc.	0.57$_{\pm0.02}$	0.40$_{\pm0.06}$(0.17)	0.41$_{\pm0.06}$(0.16)	0.31$_{\pm0.06}$(0.26)	0.27$_{\pm0.03}$(0.30)	0.93$_{\pm0.03}$(0.36)	0.50$_{\pm0.32}$**(0.07)**	0.40$_{\pm0.07}$(0.17)	0.38$_{\pm0.04}$(0.19)	0.81$_{\pm0.08}$(0.24)
		Retain Acc.	0.94$_{\pm0.03}$	0.41$_{\pm0.06}$(0.53)	0.41$_{\pm0.06}$(0.53)	0.37$_{\pm0.07}$(0.57)	0.28$_{\pm0.03}$(0.66)	0.93$_{\pm0.03}$**(0.01)**	0.50$_{\pm0.32}$(0.44)	0.41$_{\pm0.07}$(0.53)	0.41$_{\pm0.04}$(0.53)	0.92$_{\pm0.02}$**(0.02)**
		Test Acc.	0.60$_{\pm0.02}$	0.35$_{\pm0.05}$(0.25)	0.35$_{\pm0.05}$(0.25)	0.31$_{\pm0.06}$(0.29)	0.25$_{\pm0.03}$(0.35)	0.60$_{\pm0.02}$**(0.00)**	0.36$_{\pm0.20}$(0.24)	0.34$_{\pm0.04}$(0.26)	0.35$_{\pm0.03}$(0.25)	0.61$_{\pm0.01}$(0.01)
		Avg Gap	0.0000	0.3600	0.3575	0.4025	0.3675	0.1650	0.2125	0.3625	0.3650	**0.1200**
	CIFAR-10	MIA Acc.	0.76$_{\pm0.03}$	0.30$_{\pm0.26}$(0.46)	0.48$_{\pm0.50}$(0.28)	0.48$_{\pm0.50}$(0.28)	0.43$_{\pm0.37}$(0.33)	0.94$_{\pm0.01}$(0.18)	0.81$_{\pm0.11}$**(0.05)**	0.46$_{\pm0.03}$(0.30)	0.16$_{\pm0.28}$(0.60)	0.82$_{\pm0.10}$(0.06)
		Forget Acc.	0.91$_{\pm0.02}$	0.97$_{\pm0.01}$(0.06)	0.97$_{\pm0.01}$(0.06)	0.96$_{\pm0.01}$(0.05)	0.71$_{\pm0.18}$(0.20)	1.00$_{\pm0.00}$(0.09)	0.86$_{\pm0.16}$(0.05)	0.93$_{\pm0.01}$**(0.02)**	0.94$_{\pm0.02}$**(0.02)**	0.99$_{\pm0.00}$(0.08)
		Retain Acc.	0.99$_{\pm0.02}$	0.98$_{\pm0.01}$(0.01)	0.97$_{\pm0.01}$(0.02)	0.97$_{\pm0.01}$(0.02)	0.71$_{\pm0.18}$(0.28)	1.00$_{\pm0.00}$(0.01)	0.87$_{\pm0.16}$(0.12)	0.93$_{\pm0.01}$(0.06)	0.95$_{\pm0.02}$(0.04)	0.99$_{\pm0.00}$**(0.00)**
		Test Acc.	0.91$_{\pm0.02}$	0.89$_{\pm0.02}$(0.02)	0.88$_{\pm0.02}$(0.03)	0.89$_{\pm0.02}$(0.02)	0.66$_{\pm0.16}$(0.25)	0.93$_{\pm0.01}$(0.02)	0.80$_{\pm0.15}$(0.11)	0.86$_{\pm0.01}$(0.05)	0.86$_{\pm0.03}$(0.05)	0.91$_{\pm0.01}$**(0.00)**
		Avg Gap	0.0000	0.1375	0.0975	0.0925	0.2650	0.0750	0.0825	0.1075	0.1800	**0.0350**
	MUFAC	MIA Acc.	0.48$_{\pm0.04}$	0.54$_{\pm0.09}$(0.06)	0.53$_{\pm0.08}$**(0.05)**	0.33$_{\pm0.31}$(0.15)	0.34$_{\pm0.01}$(0.14)	0.70$_{\pm0.05}$(0.22)	0.70$_{\pm0.06}$(0.22)	0.40$_{\pm0.35}$(0.08)	0.53$_{\pm0.08}$**(0.05)**	0.53$_{\pm0.04}$**(0.05)**
		Forget Acc.	0.47$_{\pm0.04}$	0.64$_{\pm0.04}$(0.17)	0.68$_{\pm0.04}$(0.21)	0.66$_{\pm0.04}$(0.19)	0.53$_{\pm0.07}$**(0.06)**	0.88$_{\pm0.06}$(0.41)	0.87$_{\pm0.06}$(0.40)	0.71$_{\pm0.03}$(0.24)	0.63$_{\pm0.05}$(0.16)	0.86$_{\pm0.04}$(0.39)
		Retain Acc.	0.89$_{\pm0.04}$	0.64$_{\pm0.04}$(0.25)	0.66$_{\pm0.03}$(0.23)	0.80$_{\pm0.03}$(0.09)	0.76$_{\pm0.04}$(0.13)	0.89$_{\pm0.04}$**(0.00)**	0.89$_{\pm0.05}$**(0.00)**	0.73$_{\pm0.03}$(0.16)	0.67$_{\pm0.04}$(0.22)	0.85$_{\pm0.08}$(0.04)
		Test Acc.	0.56$_{\pm0.02}$	0.43$_{\pm0.01}$(0.13)	0.43$_{\pm0.01}$(0.13)	0.47$_{\pm0.02}$(0.09)	0.48$_{\pm0.03}$(0.08)	0.54$_{\pm0.03}$**(0.02)**	0.54$_{\pm0.03}$**(0.02)**	0.46$_{\pm0.01}$(0.10)	0.43$_{\pm0.02}$(0.13)	0.54$_{\pm0.05}$**(0.02)**
		Avg Gap	0.0000	0.1525	0.1550	0.1300	**0.1025**	0.1625	0.1600	0.1450	0.1400	0.1250

Table 6. **Accuracy Metrics used to compute Average (Avg) Gap**. Mean performance and standard deviation ($\mu \pm \sigma$) are reported across three trials with different forget and retain sets. Performance gaps relative to the Gold Standard are noted as (●), with smaller gaps indicating stronger performance. Avg Gap serves as a key indicator, summarizing performance across MIA, Forget, Retain, and Test Accuracy. LoTUS achieves state-of-the-art results in MIA, retain and test accuracies, ranking as the best in most cases and second-best in the remaining.

further reinforces its capacity to balance forgetting and retention, as evidenced by Avg Gap.

This also raises concerns about the widely used Avg Gap metric, as it may lead to misleading evaluation of unlearning. However, incorporating both Avg Gap and JSD metrics in the evaluation helps mitigate these concerns.

10. Detailed Comparison of RF-JSD and ZRF

The ZRF metric [9] assesses the unlearning effectiveness by computing the JSD score twice: once between the unlearned and a randomly initialized model, and again between the pre-trained and the same randomly initialized model. The latter serves as a reference point for the optimal value.

By contrast, RF-JSD simplifies the evaluation by requiring only a single JSD computation –between the unlearned model and the original model– where the optimal value is fixed at zero. This direct alignment with the JSD metric (which also has an optimal value fixed at zero) facilitates a more comprehensive evaluation of the unlearning effectiveness.

Beyond the obvious efficiency gain from RF-JSD not requiring inference on an additional randomly initialized model to obtain a reference score –unlike ZRF– its use of normalized class-wise mean distributions further enhances computational efficiency. Specifically, this reduces the complexity from $O(n_f \cdot n_u \cdot k)$ to $O\big((n_f + n_u) \cdot k\big)$, where n_f and n_u denote the number of samples in the forget and test sets, respectively, and k is the number of classes. This optimization significantly reduces the computational overhead, particularly for large datasets. In this analysis, we exclude the complexity of the feed-forward process, which remains unchanged.

Finally, Table 7 presents a detailed correlation between RF-JSD and JSD as measured by the Pearson correlation coefficient (PCC) for all benchmarks. PCC results exhibit a strong correlation between these two metrics, with RF-JSD offering the additional advantage of not requiring a retrained model (*i.e.*, gold standard).

	Dataset $\left(\frac{\text{num. of forget samples}}{\text{num. of training samples}} \times 100\%\right)$	PCC (↑)	p-value (↓)
ViT	CIFAR-100 (10%)	0.84	0.0043
	CIFAR-10 (10%)	0.92	0.0005
	MUFAC	0.93	0.0003
	CIFAR-100 (50%)	0.94	0.0001
	CIFAR-10 (50%)	0.99	0.0000
ResNet18	CIFAR-100 (10%)	0.97	0.0000
	CIFAR-10 (10%)	0.90	0.0011
	MUFAC	0.88	0.0018
	CIFAR-100 (50%)	0.91	0.0006
	CIFAR-10 (50%)	0.89	0.0013
	Mean ± Std	$0.92_{\pm 0.04}$	$0.0010_{\pm 0.0016}$

Table 7. **Retrain Free-JSD (RF-JSD) and JSD Correlation** measured with the Pearson correlation coefficient (PCC). A high PCC (closer to 1) indicates a strong correlation, while a low p-value reflects high confidence in the measurement. The table shows that RF-JSD strongly correlates with the well-established JSD metric across datasets and architectures, demonstrating its reliability as unlearning metric that is particularly useful when the gold standard model is not available (*e.g*., it is impractical due to high computational complexity or it is infeasible due to not access to the original training set) .

	Metric (↓)	Finetuning	NegGrad+	RndLbl	**LoTUS**
ViT C-100	Avg. Gap	0.0400	0.0600	0.0250	**0.0225**
	JSD ×1e4	$0.02_{\pm 0.00}$	$0.03_{\pm 0.01}$	$\mathbf{0.01}_{\pm \mathbf{0.01}}$	$\mathbf{0.01}_{\pm \mathbf{0.00}}$
	Time (min)	$\mathbf{6.34}_{\pm \mathbf{0.01}}$	$12.68_{\pm 0.02}$	$12.63_{\pm 0.02}$	$13.79_{\pm 0.02}$
ViT C-10	Avg. Gap	0.0125	0.0200	**0.0050**	**0.0050**
	JSD ×1e4	$\mathbf{0.00}_{\pm \mathbf{0.00}}$	$0.01_{\pm 0.00}$	$\mathbf{0.00}_{\pm \mathbf{0.00}}$	$\mathbf{0.00}_{\pm \mathbf{0.00}}$
	Time (min)	$\mathbf{6.48}_{\pm \mathbf{0.27}}$	$12.97_{\pm 0.50}$	$12.60_{\pm 0.03}$	$14.09_{\pm 0.53}$
RN18 C-100	Avg. Gap	0.3200	0.3150	0.3875	**0.1725**
	JSD ×1e4	$1.39_{\pm 0.10}$	$1.38_{\pm 0.08}$	$1.03_{\pm 0.23}$	$\mathbf{0.28}_{\pm \mathbf{0.00}}$
	Time (min)	$\mathbf{0.26}_{\pm \mathbf{0.01}}$	$0.52_{\pm 0.00}$	$0.48_{\pm 0.00}$	$0.57_{\pm 0.01}$
RN18 C-10	Avg. Gap	0.1100	0.1475	0.2100	**0.0650**
	JSD ×1e4	$0.31_{\pm 0.00}$	$0.31_{\pm 0.01}$	$0.73_{\pm 0.22}$	$\mathbf{0.09}_{\pm \mathbf{0.01}}$
	Time (min)	$\mathbf{0.26}_{\pm \mathbf{0.01}}$	$0.51_{\pm 0.02}$	$0.48_{\pm 0.00}$	$0.57_{\pm 0.00}$

Table 8. **Scaling up the Forget set to 50% of the training sets:** LoTUS outperforms basic unlearning methods in unlearning effectiveness, but not in efficiency.

11. Detailed Analysis on the Time Complexity

This section provides an in-depth analysis that demonstrates why LoTUS achieves superior efficiency compared to state-of-the-art approaches, as observed in Tabs. 1, 3 and 4 and discussed in Sec. 5. We define the time complexity of model updates in DNNs, generalized across architectures like ResNet18 and ViT, as follows:

$$O(E\cdot\frac{n_f+n_r}{B}\cdot N_p\cdot N_i) \tag{13}$$

where E represents the total number of epochs, n_f and n_r are the number of instances in D_f (forget set) and D_r (retain set) used during unlearning, respectively, B is the batch size, N_p is the total number of model parameters, and N_i is the input dimensionality. While this definition abstracts away architectural-specific details and optimizations, it provides a meaningful framework for comparing methods on shared benchmarks.

The main advantage of LoTUS over Finetuning, NegGrad+, Random Labeling, and SCRUB is that it requires significantly fewer instances n_r from the retain set D_r. Specifically, LoTUS can use only 30% of the instances in D_r to preserve the utility of the model. All other factors (E, n_f, B, N_p, N_i) are the same for all unlearning baselines in our benchmarks. As shown in Tab. 1, LoTUS achieves superior efficiency.

As the number of instances n_f in the forget set increases, the execution time of LoTUS increases, in alignment with Eq. (13). Thus, in the extreme scenario where 50% of the forget set is designated for unlearning, we observe that the efficiency of Finetuning, NegGrad+, and Random Labeling may exceed that of LoTUS, as shown in Tab. 3. In Tab. 8 we present the scores of these basic unlearning methods that are not presented in Tab. 3, and show that they may be better in terms of efficiency, but LoTUS remains the best in terms of effectiveness.

Next, we compare the time complexity of the auxiliary computations between LoTUS and other unlearning baselines that use equal or fewer samples from the retain set D_r:

LoTUS: $O(n_f\!+\!n_v)$, where n_v is the total number of instances in the validation set, for computing τ_d.

Bad Teacher [9]: $O\big((n_f\!+\!n_r)\!\cdot\!k\big)$, where k is the total number of classes, for calculating the $\mathcal{KL}$ divergences between the student and the teacher.

UNSIR [34]: $O(E_{noise}\!\cdot\!n_f\!\cdot\!N_i)$, where E_{noise} are the epochs for noise optimization, and N_i represents the total input dimensionality (product of channels, width and height of the images).

SSD [14]: $O(n_f\!\cdot\!N_p^2)$ for computing the Fisher Information Matrix.

In this analysis, we exempt the complexity of the feedforward process which is the same for all the unlearning methods in our benchmarks. Also, SalUn [12] introduces a computational overhead prior to unlearning due to the computations of the saliency mask for weight pruning. The complexity of this auxiliary computation contributes to the overall complexity of the downstream method used for unlearning (*e.g*., Random Labeling and LoTUS in our case). Among the unlearning methods, LoTUS is the only one with auxiliary computations of linear complexity.

12. Cleaning the MUFAC Dataset

We identified duplicates within the forget, retain, validation, and test splits of the MUFAC dataset. More critically,

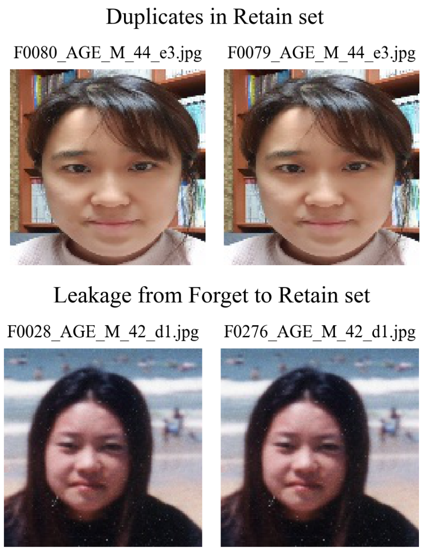


Figure 3. **Duplicates in MUFAC:** An example of a duplicate within the retain set (top) and a critical duplicate shared between the retain and forget set (bottom), which introduces information leakage.

we discovered instances of information leakage across these splits. To address this, we used image hashing to detect identical images with different filenames in these splits, as shown in Fig. 3.

After cleaning MUFAC, the retain set contains $5,513$ samples, and the forget set contains $1,062$ samples. We provide the code for identifying duplicate images and cleaning MUFAC in https://github.com/cspartalis/LoTUS.

Moreover, Figure 4 presents the class distribution of samples in the clean version of MUFAC, showing that the forget set and the unseen set (*i.e*., the validation set in our case) follow different class distributions. The strong performance of LoTUS in MUFAC further suggest that the assumption of distributional similarity between the forget and unseen sets, discussed in Sec. 3.2, can be relaxed.

13. Failure Analysis

Unlearning samples from MUFAC (the clean version) presents greater challenges for all unlearning methods, as reflected in significantly higher JSD scores in Tab. 1. In addition, MUFAC & ResNet18 is the only benchmark where LoTUS achieves the second-best Avg Gap rather than the best. To explore the particularities of this dataset, we investigated the orthogonality of the forget and retain sets (*i.e*., how much they differ). Figure 5 presents that the images in the forget and retain sets of MUFAC are more similar, making unlearning more challenging.

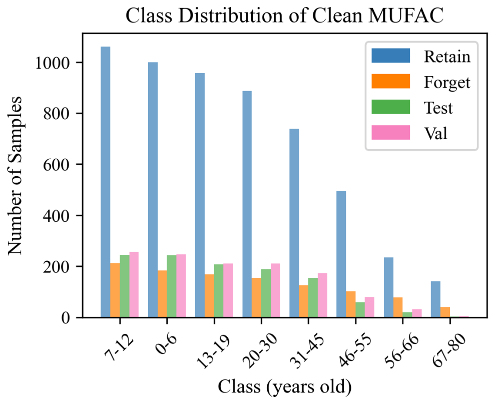


Figure 4. **Number of MUFAC Samples per Class & Split.** Unlike the balanced CIFAR-10/100 splits, MUFAC exhibits imbalanced class distributions of that varies across the retain, forget, test, and validation splits.

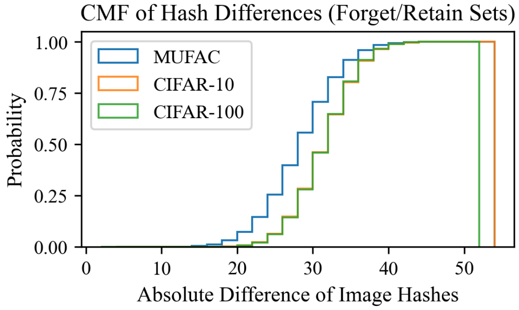
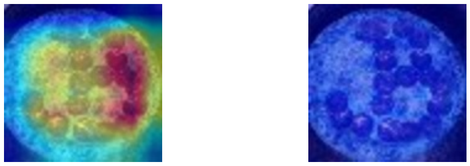


Figure 5. **Orthogonality of Forget/Retain Sets.** We measure the similarity between samples in the forget and retain sets using the absolute difference between their image hashes. MUFAC exhibits significantly higher similarity between forget and retain sets, complicating the unlearning process.



(a) prediction: *pizza*

(b) prediction: *potpie*

Figure 6. **Class Activation Maps and Model Predictions:** (a) before and (b) after class unlearning.

14. Class Unlearning with LoTUS

After retraining the model excluding a single *pizza* image from the training set, the model preserves *global information* that stems from the remaining *pizzas* in the training set, being able to correctly classify many of them (see Forget Acc. in Tab. 6). In instance-wise unleanring, LoTUS prevents performance degradation by preventing the elimination of *global information*. To do so, it uses

	Metric (↓)	Gold Std	Finetuning	NegGrad+	RndLbl	BadT	SCRUB	SSD	UNSIR	SalUn	**LoTUS**
TinyIN Pizza	Avg Gap	0.0000	0.2975	0.3250	$\underline{0.2925}$	0.3125	0.4200	0.3650	0.5075	$\underline{0.2925}$	**0.0925**
	JSD ×1e4	$0.00_{\pm 0.00}$	$94.96_{\pm 7.24}$	$86.36_{\pm 9.66}$	$92.27_{\pm 6.43}$	$72.62_{\pm 22.07}$	$73.10_{\pm 0.82}$	$\underline{34.96_{\pm 14.21}}$	$102.29_{\pm 9.33}$	$91.01_{\pm 8.59}$	$\mathbf{37.02}_{\pm \mathbf{18.68}}$
	Time (min.)	$42.15_{\pm 16.05}$	$3.23_{\pm 0.01}$	$3.24_{\pm 0.03}$	$3.27_{\pm 0.03}$	$1.59_{\pm 0.01}$	$4.05_{\pm 0.03}$	$3.19_{\pm 0.03}$	$\mathbf{1.01}_{\pm \mathbf{0.01}}$	$3.98_{\pm 0.01}$	$\underline{1.30_{\pm 0.02}}$
C-100 Beaver	Avg Gap	0.0000	$\underline{0.2825}$	0.3725	0.2925	0.3000	0.3225	0.4325	0.4050	$\underline{0.2850}$	**0.1200**
	JSD ×1e4	$0.00_{\pm 0.00}$	$101.48_{\pm 2.87}$	$108.50_{\pm 2.59}$	$102.66_{\pm 3.11}$	$78.65_{\pm 3.12}$	$64.09_{\pm 8.71}$	$\underline{45.19_{\pm 9.19}}$	$76.28_{\pm 6.88}$	$100.93_{\pm 2.44}$	$\mathbf{25.46}_{\pm \mathbf{1.41}}$
	Time (min.)	$4.00_{\pm 0.11}$	$0.43_{\pm 0.00}$	$0.44_{\pm 0.01}$	$0.45_{\pm 0.00}$	$0.26_{\pm 0.01}$	$0.55_{\pm 0.00}$	$0.83_{\pm 0.03}$	$\mathbf{0.20}_{\pm \mathbf{0.01}}$	$1.16_{\pm 0.01}$	$\underline{0.23_{\pm 0.01}}$

Table 9. **Class Unlearning** with ResNet18 models and the TinyImageNet (TinyIN) and CIFAR-100 (C-100) datasets. We highlight the **best** and second-best scores.

accuracy on labeled unseen *pizzas*, $\mathrm{Acc}(f_{\mathrm{orig}}, D_u)$ in Eq. (11) as an estimator of *global information*.

Framing class unlearning as sequential instance-wise unlearning applied to all class samples, *global information* is ultimately eliminated (see Class Activation Maps of *pizza* class in Fig. 6). Since there is no global information to estimate, we also do not need the unseen set. To adapt LoTUS to class unlearning, we set as objective the accuracy on the forget set to become zero (an empirical observation by retaining the model without the specific class):

$$\tau_d = \exp\left(\alpha(\mathrm{Acc}(f_{\mathrm{un}}, D_f) - \cancelto{0}{\mathrm{Acc}(f_{\mathrm{orig}}, D_u)})\right) \tag{14}$$

Table 9 shows that LoTUS can be adapted to the class unlearning task, outperforming state-of-the-art methods, combining unlearning effectiveness and efficiency.

15. Contribution of Gumbel noise

In Tab. 10, we demonstrate the contribution of the introduction of Gumbel noise in the Softmax activation function. To do so, we perform an ablation analysis using the Gumbel Softmax and the Softmax with Temperature as activation functions in LoTUS. Softmax with Temperature is defined similarly with Eq. (15) as:

$$p_i = s_t(\pi, \tau) = \frac{\exp\left(\left(\log \pi_i\right)/\tau\right)}{\sum_{j=1}^{k} \exp\left(\left(\log \pi_j\right)/\tau\right)}, \; i = 1, \ldots, k \tag{15}$$

16. Entropy-based Analysis of the Streisand Effect

Further evaluation of the Streisand effect includes investigating the model's uncertainty, as in [15]. In Fig. 7, it is shown that LoTUS prevents an adversary from readily inferring whether an instance is a member of the training set, or whether it belongs to the forget or retain set, since the entropy distributions of the forget/retain/test sets are similar. In contrast, the existing unlearning method [17] that also performs in the output space, but indiscriminately increases the entropy, clearly presents a significant vulnerability to the Streisand effect.

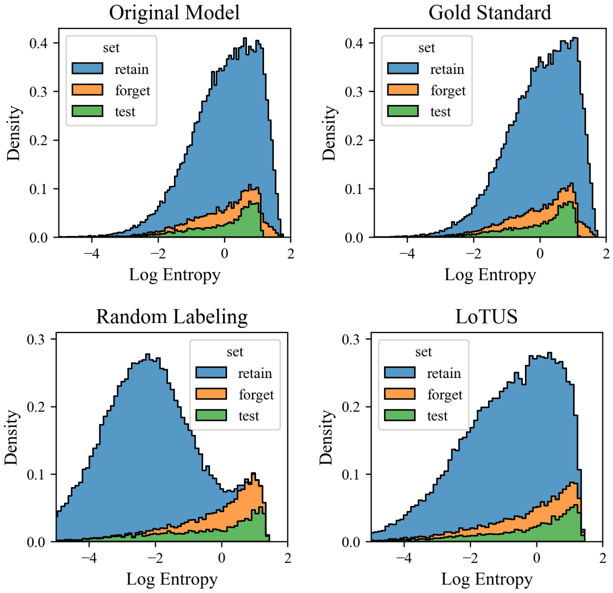


Figure 7. **Privacy Evaluation via entropy comparison:** LoTUS achieves indistinguishable entropy distributions between forget and retain sets, similar to the orignal and gold standard models. In contrast, Random Labeling produces disproportionately lower entropy in the retain set, making it easier for adversaries to distinguish retain from forget and unseen samples.

17. Social Impact

LoTUS can address privacy-related concerns, such as opt-out requests, where users request their data to be deleted not only from the databases, but also from the DNN models. From a security perspective, LoTUS can be applied to unlearn training samples modified by adversaries, which may otherwise compromise the model's performance. In such scenarios, where privacy or security issues arise for specific data points and need to be removed, instance-wise unlearning is more consistent with real-world conditions than class unlearning [5].

		Vision Transformer				ResNet18			
		TinyImageNet	CIFAR-100	CIFAR-10	MUFAC	TinyImageNet	CIFAR-100	CIFAR-10	MUFAC
Avg Gap	Gumbel-Softmax	**0.0150**	**0.125**	**0.0050**	**0.0200**	**0.1675**	0.1200	**0.0350**	0.1250
	Softmax with Temperature	0.0675	0.0225	**0.0050**	**0.0200**	0.1850	**0.1075**	0.0675	**0.1175**
JSD$\times 1e4$	Gumbel-Softmax	**0.03**	**0.04**	**0.01**	**0.05**	**0.62**	1.67	**0.32**	**6.90**
	Softmax with Temperature	0.15	**0.04**	**0.01**	0.08	0.65	**1.36**	0.41	7.33

Table 10. **Contribution of Gumbel noise into the activation function.** Ablation analysis using Gumbel-Softmax and Softmax with Temperature as activation functions. LoTUS performs better with Gumbel-Softmax in the majority of the benchmarks.