Appendix

A Proof of Proposition 1

Recap the definition of model update $\Delta(\mathbf{w})$ in (1) and $\boldsymbol{\theta}_{\mathrm{o}} = \boldsymbol{\theta}(\mathbf{1}/N)$, we approximate $\Delta(\mathbf{w})$ by the first-order Taylor expansion of $\boldsymbol{\theta}(\mathbf{w})$ at $\mathbf{w} = \mathbf{1}/N$. This leads to

$$\Delta(\mathbf{w}) = \boldsymbol{\theta}(\mathbf{w}) - \boldsymbol{\theta}(\mathbf{1}/N) \approx \left.\frac{d\boldsymbol{\theta}(\mathbf{w})}{d\mathbf{w}}\right|_{\mathbf{w}=\mathbf{1}/N} (\mathbf{w} - \mathbf{1}/N), \tag{A1}$$

where $\frac{d\boldsymbol{\theta}(\mathbf{w})}{d\mathbf{w}} \in \mathbb{R}^{M\times N}$, and recall that $M = |\boldsymbol{\theta}_{\mathrm{o}}|$ is the number of model parameters. The gradient $\frac{d\boldsymbol{\theta}(\mathbf{w})}{d\mathbf{w}}$ is known as implicit gradient [76] since it is defined through the solution of the optimization problem $\boldsymbol{\theta}(\mathbf{w}) = \arg\min_{\boldsymbol{\theta}} L(\mathbf{w}, \boldsymbol{\theta})$, where recall that $L(\mathbf{w}, \boldsymbol{\theta}) = \sum_{i=1}^{N}[w_i \ell_i(\boldsymbol{\theta}, \mathbf{z}_i)]$. By the stationary condition of $\boldsymbol{\theta}(\mathbf{w})$, we obtain

$$\nabla_{\boldsymbol{\theta}} L(\mathbf{w}, \boldsymbol{\theta}(\mathbf{w})) = \mathbf{0}. \tag{A2}$$

Next, we take the derivative of (A2) w.r.t. $\mathbf{w}$ based on the implicit function theorem [76] assuming that $\boldsymbol{\theta}(\mathbf{w})$ is the unique solution to minimizing L. This leads to

$$\left[\frac{d\boldsymbol{\theta}(\mathbf{w})}{d\mathbf{w}}\right]^T \left[\nabla_{\boldsymbol{\theta},\boldsymbol{\theta}} L(\mathbf{w}, \boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{\theta}(\mathbf{w})}\right] + \nabla_{\mathbf{w},\boldsymbol{\theta}} L(\mathbf{w}, \boldsymbol{\theta}(\mathbf{w})) = \mathbf{0}, \tag{A3}$$

where $\nabla_{\mathbf{a},\mathbf{b}} = \nabla_{\mathbf{a}}\nabla_{\mathbf{b}} \in \mathbb{R}^{|\mathbf{a}|\times|\mathbf{b}|}$ is the second-order partial derivative. Therefore,

$$\frac{d\boldsymbol{\theta}(\mathbf{w})}{d\mathbf{w}} = -\left[\nabla_{\boldsymbol{\theta},\boldsymbol{\theta}} L(\mathbf{w}, \boldsymbol{\theta}(\mathbf{w}))\right]^{-1} \nabla_{\mathbf{w},\boldsymbol{\theta}} L(\mathbf{w}, \boldsymbol{\theta}(\mathbf{w}))^T, \tag{A4}$$

where $\nabla_{\mathbf{w},\boldsymbol{\theta}} L(\mathbf{w}, \boldsymbol{\theta}(\mathbf{w}))$ can be expanded as

$$\nabla_{\mathbf{w},\boldsymbol{\theta}} L(\mathbf{w}, \boldsymbol{\theta}(\mathbf{w})) = \nabla_{\mathbf{w}}\nabla_{\boldsymbol{\theta}} \sum_{i=1}^{N} [w_i \ell_i(\boldsymbol{\theta}(\mathbf{w}), \mathbf{z}_i)] \tag{A5}$$

$$= \nabla_{\mathbf{w}} \sum_{i=1}^{N} [w_i \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}(\mathbf{w}), \mathbf{z}_i)] \tag{A6}$$

$$= \begin{bmatrix} \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}(\mathbf{w}), \mathbf{z}_1)^T \\ \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}(\mathbf{w}), \mathbf{z}_2)^T \\ \vdots \\ \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}(\mathbf{w}), \mathbf{z}_N)^T \end{bmatrix}. \tag{A7}$$

Based on (A4) and (A7), we obtain the closed-form of implicit gradient at $\mathbf{w} = \mathbf{1}/N$:

$$\begin{aligned} \frac{d\boldsymbol{\theta}(\mathbf{w})}{d\mathbf{w}}\,|_{\mathbf{w}=\mathbf{1}/N} &= -\left[\nabla_{\boldsymbol{\theta},\boldsymbol{\theta}} L(\mathbf{1}/N, \boldsymbol{\theta}(\mathbf{1}/N))\right]^{-1} \left[\nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}(\mathbf{1}/N), \mathbf{z}_1) \quad \dots \quad \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}(\mathbf{1}/N), \mathbf{z}_N)\right] \\ &= -\mathbf{H}^{-1} \left[\nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}(\mathbf{1}/N), \mathbf{z}_1) \quad \dots \quad \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}(\mathbf{1}/N), \mathbf{z}_N)\right], \end{aligned} \tag{A8}$$

where $\mathbf{H} = \nabla_{\boldsymbol{\theta},\boldsymbol{\theta}} L(\mathbf{1}/N, \boldsymbol{\theta}(\mathbf{1}/N))$.

Substituting (A8) into (A1), we obtain

$$\begin{aligned} \Delta(\mathbf{w}) &\approx -\mathbf{H}^{-1} \left[\nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}(\mathbf{1}/N), \mathbf{z}_1) \quad \dots \quad \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}(\mathbf{1}/N), \mathbf{z}_N)\right] (\mathbf{w} - \mathbf{1}/N) \\ &= -\mathbf{H}^{-1} \sum_{i=1}^{N} [(w_i - 1/N) \nabla_{\boldsymbol{\theta}} \ell_i(\boldsymbol{\theta}(\mathbf{1}/N), \mathbf{z}_i)] \\ &= \mathbf{H}^{-1} \nabla_{\boldsymbol{\theta}} L(\mathbf{1}/N - \mathbf{w}, \boldsymbol{\theta}_{\mathrm{o}}), \end{aligned} \tag{A9}$$

where the last equality holds by the definition of $L(\mathbf{w}, \boldsymbol{\theta}) = \sum_{i=1}^{N}[w_i \ell_i(\boldsymbol{\theta}, \mathbf{z}_i)]$.

The proof is now complete.

B Proof of Proposition 2

The proof follows [8, Sec. 5], with the additional condition that the model is **sparse** encoded by a pre-fixed (binary) pruning mask $\mathbf{m}$, namely, $\boldsymbol{\theta}' := \mathbf{m} \odot \boldsymbol{\theta}$. Then, based on [8, Eq. 5], the model updated by SGD yields

$$\boldsymbol{\theta}'_t \approx \boldsymbol{\theta}'_0 - \eta \mathbf{m} \odot \sum_{i=1}^{t-1} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}'_0, \hat{\mathbf{z}}_i) + \mathbf{m} \odot (\sum_{i=1}^{t-1} f(i)), \tag{A10}$$

where $\boldsymbol{\theta}'_0 = \mathbf{m} \odot \boldsymbol{\theta}_0$ is the model initialization when using SGD-based sparse training, $\{\hat{\mathbf{z}}_i\}$ is the sequence of stochastic data samples, t is the number of training iterations, η is the learning rate, and $f(i)$ is defined recursively as

$$f(i) = -\eta \nabla^2_{\boldsymbol{\theta},\boldsymbol{\theta}} \ell(\boldsymbol{\theta}'_0, \hat{\mathbf{z}}_i) \left(-\eta \sum_{j=0}^{i-1} \mathbf{m} \odot \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}'_0, \hat{\mathbf{z}}_j) + \sum_{j=0}^{i-1} (\mathbf{m} \odot f(j)) \right), \tag{A11}$$

with $f(0) = 0$. Inspired by the second term of (A10), to unlearn the data sample $\hat{\mathbf{z}}_i$, we will have to add back the first-order gradients under $\hat{\mathbf{z}}_i$. This corresponds to the GA-based approximate unlearning method. Yet, this approximate unlearning introduces an unlearning error, given by the last term of (A10)

$$\mathbf{e}_{\mathbf{m}}(\boldsymbol{\theta}_0, \{\hat{\mathbf{z}}_i\}, t, \eta) := \mathbf{m} \odot (\sum_{i=1}^{t-1} f(i)). \tag{A12}$$

Next, if we interpret the mask $\mathbf{m}$ as a diagonal matrix $\mathrm{diag}(\mathbf{m})$ with 0's and 1's along its diagonal based on $\mathbf{m}$, we can then express the sparse model $\mathbf{m} \odot \boldsymbol{\theta}$ as $\mathrm{diag}(\mathbf{m})\boldsymbol{\theta}$. Similar to [8, Eq. 9], we can derive a bound on the unlearning error (A12) by ignoring the terms other than those with η^2 in $f(i)$, *i.e.*, (A11). This is because, in the recursive form of $f(i)$, all other terms exhibit a higher degree of the learning rate η compared to η^2. As a result, we obtain

$$\begin{aligned}
e(\mathbf{m}) &= \|\mathbf{e}_{\mathbf{m}}(\boldsymbol{\theta}_0, \{\hat{\mathbf{z}}_i\}, t, \eta)\|_2 = \left\| \mathbf{m} \odot (\sum_{i=1}^{t-1} f(i)) \right\|_2 \\
&\approx \eta^2 \left\| \mathrm{diag}(\mathbf{m}) \sum_{i=1}^{t-1} \nabla^2_{\boldsymbol{\theta},\boldsymbol{\theta}} \ell(\boldsymbol{\theta}'_0, \hat{\mathbf{z}}_i) \sum_{j=0}^{i-1} \mathbf{m} \odot \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}'_0, \hat{\mathbf{z}}_j) \right\|_2 \\
&\leq \eta^2 \sum_{i=1}^{t-1} \left\| \mathrm{diag}(\mathbf{m}) \nabla^2_{\boldsymbol{\theta},\boldsymbol{\theta}} \ell(\boldsymbol{\theta}'_0, \hat{\mathbf{z}}_i) \sum_{j=0}^{i-1} \mathbf{m} \odot \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}'_0, \hat{\mathbf{z}}_j) \right\|_2 && \text{(Triangle inequality)} \\
&\leq \eta^2 \sum_{i=1}^{t-1} \left\| \mathrm{diag}(\mathbf{m}) \nabla^2_{\boldsymbol{\theta},\boldsymbol{\theta}} \ell(\boldsymbol{\theta}'_0, \hat{\mathbf{z}}_i) \right\| \left\| \sum_{j=0}^{i-1} \mathbf{m} \odot \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}'_0, \hat{\mathbf{z}}_j) \right\|_2 && \text{(A13)} \\
&\lesssim \eta^2 \sum_{i=1}^{t-1} \left\| \mathrm{diag}(\mathbf{m}) \nabla^2_{\boldsymbol{\theta},\boldsymbol{\theta}} \ell(\boldsymbol{\theta}'_0, \hat{\mathbf{z}}_i) \right\| \frac{i}{t} \left\| \boldsymbol{\theta}'_t - \boldsymbol{\theta}'_0 \right\|_2 && \text{(A14)} \\
&\leq \eta^2 \sigma(\mathbf{m}) \left\| \mathbf{m} \odot (\boldsymbol{\theta}_t - \boldsymbol{\theta}_0) \right\|_2 \frac{1}{t} \frac{t-1}{2} t = \frac{\eta^2}{2} (t-1) \|\mathbf{m} \odot (\boldsymbol{\theta}_t - \boldsymbol{\theta}_0)\|_2 \sigma(\mathbf{m}), && \text{(A15)}
\end{aligned}$$

where the inequality (A14) holds given the fact that $\sum_{j=0}^{i-1} \mathbf{m} \odot \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}'_0, \hat{\mathbf{z}}_j)$ in (A13) can be approximated by its expectation $\frac{i(\boldsymbol{\theta}'_t - \boldsymbol{\theta}'_0)}{t}$ [8, Eq. 7], and $\sigma(\mathbf{m}) := \max_j \{\sigma_j(\nabla^2_{\boldsymbol{\theta},\boldsymbol{\theta}} \ell), \text{if } m_j \neq 0\}$, *i.e.*, the largest eigenvalue among the dimensions left intact by the binary mask $\mathbf{m}$. The above suggests that the unlearning error might be large if $\mathbf{m} = \mathbf{1}$ (no pruning). Based on (A15), we can then readily obtain the big O notation in (2). This completes the proof.

C Additional Experimental Details and Results

C.1 Datasets and models

We summarize the datasets and model configurations in Tab. A1.

Table A1: Dataset and model setups.

Settings	CIFAR-10		SVHN	CIFAR-100	ImageNet
	ResNet-18	VGG-16	ResNet-18	ResNet-18	ResNet-18
Batch Size	128	128	128	128	1024

C.2 Additional training and unlearning settings

Training configuration of pruning. For all pruning methods, including IMP [15], SynFlow [38], and OMP [17], we adopt the settings from the current SOTA implementations [17]; see a summary in Tab. A2. For IMP, OMP, and SynFlow, we adopt the step learning rate scheduler with a decay rate of 0.1 at 50% and 75% epochs. We adopt 0.1 as the initial learning rate for all pruning methods.

Additional training details of MU. For all datasets and model architectures, we adopt 10 epochs for FT, and 5 epochs for GA method. The learning rate for FT and GA are carefully tuned between $[10^{-5}, 0.1]$ for each dataset and model architecture. In particular, we adopt 0.01 as the learning rate for FT method and 10^{-4} for GA on the CIFAR-10 dataset (ResNet-18) at different sparsity levels. By default, we choose SGD as the optimizer for the FT and GA methods. As for FF method, we perform a greedy search for hyperparameter tuning [12] between 10^{-8} and 10^{-6}.

C.3 Detailed metric settings

Details of MIA implementation. MIA is implemented using the prediction confidence-based attack method [46]. There are mainly two phases during its computation: **(1) training phase**, and **(2) testing phase**. To train an MIA model, we first sample a balanced dataset from the remaining dataset ($\mathcal{D}_\mathrm{r}$) and the test dataset (different from the forgetting dataset $\mathcal{D}_\mathrm{f}$) to train the MIA predictor. The learned MIA is then used for MU evaluation in its testing phase. To evaluate the performance of MU, MIA-Efficacy is obtained by applying the learned MIA predictor to the unlearned model ($\boldsymbol{\theta}_\mathrm{u}$) on the forgetting dataset ($\mathcal{D}_\mathrm{f}$). Our objective is to find out how many samples in $\mathcal{D}_\mathrm{f}$ can be correctly predicted as non-training samples by the MIA model against $\boldsymbol{\theta}_\mathrm{u}$. The formal definition of MIA-Efficacy is then given by:

$$\text{MIA-Efficacy} = \frac{TN}{|\mathcal{D}_\mathrm{f}|}, \tag{A16}$$

where TN refers to the true negatives predicted by our MIA predictor, *i.e.*, the number of the forgetting samples predicted as non-training examples, and $|\mathcal{D}_\mathrm{f}|$ refers to the size of the forgetting dataset. As described above, MIA-Efficacy leverages the privacy attack to justify how good the unlearning performance could be.

C.4 Additional experiment results

Model sparsity benefits privacy of MU for 'free'. It was recently shown in [27, 28] that model sparsification helps protect data privacy, in terms of defense against MIA used to infer training data

Table A2: Detailed training details for model pruning.

Experiments	CIFAR-10/CIFAR-100	SVHN	ImageNet
Training epochs	182	160	90
Rewinding epochs	8	8	5
Momentum	0.9	0.9	0.875
ℓ_2 regularization	$5e^{-4}$	$5e^{-4}$	$3.05e^{-5}$
Warm-up epochs	1(75 for VGG-16)	0	8

information from a learned model. Inspired by the above, we ask if sparsity can also bring the privacy benefit to an unlearned model, evaluated by the MIA rate on the remaining dataset $\mathcal{D}_\mathrm{r}$ (that we term **MIA-Privacy**). This is different from MIA-Efficacy, which reflects the efficacy of scrubbing $\mathcal{D}_\mathrm{f}$, *i.e.*, correctly predicting that data sample in $\mathcal{D}_\mathrm{f}$ is not in the training set of the unlearned model. In contrast, MIA-Privacy characterizes the *privacy* of the unlearned model about $\mathcal{D}_\mathrm{r}$. A *lower* MIA-Privacy implies *less* information leakage.

Fig. A1 shows MIA-Privacy of unlearned models versus the sparsity ratio applied to different unlearning methods in the 'prune first, then unlearn' paradigm. As we can see, MIA-Privacy decreases as the sparsity increases. This suggests the improved privacy of unlearning on sparse models. Moreover, we observe that approximate unlearning outperforms exact unlearning (Retrain) in privacy preservation of $\mathcal{D}_\mathrm{r}$. This is because Retrain is conducted over $\mathcal{D}_\mathrm{r}$ from scratch, leading to the strongest dependence on $\mathcal{D}_\mathrm{r}$ than other unlearning methods. Another interesting observation is that IU and GA yield a much smaller MIA-Privacy than other approximate unlearning methods. The rationale behind that is IU and GA have a weaker correlation with $\mathcal{D}_\mathrm{r}$ during unlearning. Specifically, the unlearning loss of IU only involves the forgetting data influence weights, *i.e.*, $(\mathbf{1}/N - \mathbf{w})$ in (1). Similarly, GA only performs gradient ascent over $\mathcal{D}_\mathrm{f}$, with the least dependence on $\mathcal{D}_\mathrm{r}$.

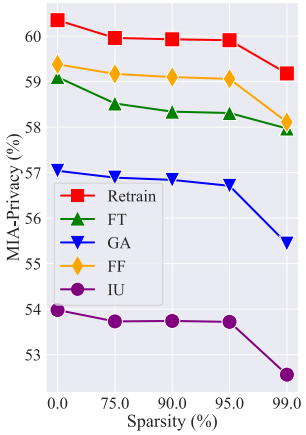

Figure A1: Privacy on $\mathcal{D}_\mathrm{r}$ (MIA-Privacy) using different unlearning methods vs. model sparsity.

Performance of 'prune first, then unlearn' on various datasets and architectures. As demonstrated in Tab. A3 and Tab. A4, the introduction of model sparsity can effectively reduce the discrepancy between approximate and exact unlearning across a diverse range of datasets and architectures. This phenomenon is observable in various unlearning scenarios. Remarkably, model sparsity enhances both UA and MIA-Efficacy metrics without incurring substantial degradation on RA and TA in different unlearning scenarios. These observations corroborate the findings reported in Tab. 3.

Table A3: MU performance vs. sparsity on additional datasets (CIFAR-100 [41] and SVHN [43]) for both class-wise forgetting and random data forgetting. The content format follows Tab. 3.

MU	UA		MIA-Efficacy		RA		TA		RTE
	Dense	95% Sparsity	Dense	95% Sparsity	Dense	95% Sparsity	Dense	95% Sparsity	(min)
Class-wise forgetting, CIFAR-100									
Retrain	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$99.97_{\pm 0.01}$	$96.68_{\pm 0.15}$	$73.74_{\pm 0.19}$	$69.49_{\pm 0.41}$	48.45
FT	$26.45_{\pm 6.29}$ (73.55)	$73.63_{\pm 5.06}$ (26.37)	$92.44_{\pm 5.93}$ (7.56)	$98.88_{\pm 4.32}$ (15.60)	$99.86_{\pm 0.04}$(0.11)	$97.72_{\pm 0.47}$ (1.04)	$74.08_{\pm 0.23}$ (0.74)	$71.37_{\pm 0.18}$ (3.00)	3.76
GA	$81.47_{\pm 0.32}$(18.53)	$99.01_{\pm 0.01}$ (0.99)	$93.47_{\pm 4.56}$ (6.53)	$100.00_{\pm 0.00}$ (0.00)	$90.33_{\pm 1.71}$ (9.64)	$80.45_{\pm 0.78}$ (16.23)	$64.94_{\pm 0.74}$ (8.80)	$60.99_{\pm 0.14}$ (8.50)	0.21
IU	$84.12_{\pm 0.34}$ (15.88)	$99.78_{\pm 0.01}$ (0.22)	$98.44_{\pm 0.45}$ (1.56)	$99.33_{\pm 0.00}$ (0.67)	$96.23_{\pm 0.02}$ (3.74)	$95.45_{\pm 0.17}$ (1.23)	$71.24_{\pm 0.22}$ (2.50)	$70.79_{\pm 0.11}$ (0.95)	4.30
Random data forgetting, CIFAR-100									
Retrain	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$99.97_{\pm 0.01}$	$96.68_{\pm 0.15}$	$73.74_{\pm 0.19}$	$69.49_{\pm 0.41}$	48.45
FT	$26.45_{\pm 6.29}$ (73.55)	$73.63_{\pm 5.06}$ (26.37)	$92.44_{\pm 5.93}$ (7.56)	$98.88_{\pm 4.32}$ (15.60)	$99.86_{\pm 0.04}$(0.11)	$97.72_{\pm 0.47}$ (1.04)	$74.08_{\pm 0.23}$ (0.74)	$71.37_{\pm 0.18}$ (3.00)	3.61
GA	$81.47_{\pm 0.32}$(18.53)	$99.01_{\pm 0.01}$ (0.99)	$93.47_{\pm 4.56}$ (6.53)	$100.00_{\pm 0.00}$ (0.00)	$90.33_{\pm 1.71}$ (9.64)	$80.45_{\pm 0.78}$ (16.23)	$64.94_{\pm 0.74}$ (8.80)	$60.99_{\pm 0.14}$ (8.50)	0.21
IU	$84.12_{\pm 0.34}$ (15.88)	$99.78_{\pm 0.01}$ (0.22)	$98.44_{\pm 0.45}$ (1.56)	$99.33_{\pm 0.00}$ (0.67)	$96.23_{\pm 0.02}$ (3.74)	$95.45_{\pm 0.17}$ (1.23)	$71.24_{\pm 0.22}$ (2.50)	$70.79_{\pm 0.11}$ (0.95)	4.29
Class-wise forgetting, SVHN									
Retrain	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$95.71_{\pm 0.12}$	$94.95_{\pm 0.05}$	42.84
FT	$11.48_{\pm 8.12}$ (88.52)	$51.93_{\pm 19.62}$ (48.07)	$86.12_{\pm 9.62}$ (13.88)	$99.42_{\pm 0.51}$ (0.58)	$100.00_{\pm 0.00}$ (0.00)	$99.00_{\pm 0.00}$ (1.00)	$95.99_{\pm 0.07}$ (0.28)	$95.89_{\pm 0.02}$ (0.94)	2.86
GA	$83.87_{\pm 0.19}$ (16.13)	$86.52_{\pm 0.11}$ (13.48)	$99.97_{\pm 0.02}$ (0.03)	$100.00_{\pm 0.00}$ (0.00)	$99.60_{\pm 0.15}$ (0.40)	$98.37_{\pm 0.11}$ (1.63)	$95.27_{\pm 0.02}$ (0.44)	$93.42_{\pm 0.07}$ (1.53)	0.28
IU	$95.11_{\pm 0.02}$ (4.89)	$100.00_{\pm 0.00}$ (0.00)	$99.89_{\pm 0.04}$ (0.11)	$100.00_{\pm 0.00}$(0.00)	$100.00_{\pm 0.00}$ (0.00)	$99.85_{\pm 0.02}$ (0.15)	$95.70_{\pm 0.09}$ (0.01)	$94.90_{\pm 0.04}$ (0.05)	3.19
Random data forgetting, SVHN									
Retrain	$4.89_{\pm 0.11}$	$4.78_{\pm 0.23}$	$15.38_{\pm 0.14}$	$15.25_{\pm 0.18}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$95.54_{\pm 0.09}$	$95.44_{\pm 0.12}$	42.71
FT	$3.56_{\pm 0.27}$ (1.33)	$3.97_{\pm 0.20}$ (0.81)	$10.05_{\pm 0.24}$ (5.33)	$10.87_{\pm 0.13}$ (4.38)	$99.89_{\pm 0.04}$ (0.11)	$98.57_{\pm 0.09}$ (1.43)	$93.55_{\pm 0.12}$ (1.99)	$93.54_{\pm 0.17}$ (1.90)	2.73
GA	$0.99_{\pm 0.42}$ (3.90)	$2.68_{\pm 0.53}$ (2.10)	$3.07_{\pm 0.53}$ (12.31)	$9.31_{\pm 0.48}$ (5.94)	$99.43_{\pm 0.22}$ (0.57)	$97.83_{\pm 0.43}$ (2.17)	$94.03_{\pm 0.21}$ (1.51)	$93.33_{\pm 0.27}$ (2.11)	0.26
IU	$3.48_{\pm 0.13}$ (1.41)	$5.62_{\pm 0.48}$ (0.84)	$9.44_{\pm 0.27}$ (5.94)	$12.28_{\pm 0.41}$(2.97)	$96.30_{\pm 0.08}$ (3.70)	$95.67_{\pm 0.15}$ (4.33)	$91.59_{\pm 0.11}$ (3.95)	$90.91_{\pm 0.26}$ (4.53)	3.21

Table A4: MU performance vs. sparsity on the additional architecture (VGG-16 [45]) for both class-wise forgetting and random data forgetting on CIFAR-10. The content format follows Tab. 3.

MU	UA		MIA-Efficacy		RA		TA		RTE
	Dense	95% Sparsity	Dense	95% Sparsity	Dense	95% Sparsity	Dense	95% Sparsity	(min)
Class-wise forgetting, VGG-16									
Retrain	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.01}$	$99.97_{\pm 0.00}$	$94.83_{\pm 0.10}$	$92.93_{\pm 0.06}$	30.38
FT	$28.00_{\pm 8.16}$ (72.00)	$34.94_{\pm 5.37}$ (65.06)	$63.23_{\pm 17.68}$ (36.77)	$68.02_{\pm 12.03}$ (31.98)	$99.87_{\pm 0.05}$(0.13)	$99.60_{\pm 0.08}$ (0.37)	$92.80_{\pm 1.28}$ (2.03)	$92.96_{\pm 0.85}$ (0.03)	1.81
GA	$77.51_{\pm 3.47}$ (22.49)	$83.93_{\pm 2.14}$ (16.07)	$80.13_{\pm 4.27}$ (19.87)	$88.04_{\pm 3.18}$ (11.96)	$96.09_{\pm 0.13}$(3.91)	$97.33_{\pm 0.08}$ (2.64)	$88.80_{\pm 1.33}$ (6.03)	$89.95_{\pm 0.78}$ (2.98)	0.27
IU	$88.58_{\pm 0.86}$ (11.42)	$98.78_{\pm 0.44}$ (1.22)	$92.27_{\pm 1.14}$ (7.73)	$99.91_{\pm 0.05}$ (0.09)	$96.89_{\pm 0.27}$(3.11)	$93.18_{\pm 0.28}$ (6.79)	$89.81_{\pm 1.01}$ (5.02)	$87.45_{\pm 0.81}$ (5.48)	2.51
Random data forgetting, VGG-16									
Retrain	$7.13_{\pm 0.60}$	$7.47_{\pm 0.30}$	$13.02_{\pm 0.77}$	$13.51_{\pm 0.50}$	$100.00_{\pm 0.01}$	$99.93_{\pm 0.01}$	$92.80_{\pm 0.17}$	$91.98_{\pm 0.22}$	30.29
FT	$0.86_{\pm 0.29}$ (6.27)	$1.46_{\pm 0.22}$ (6.01)	$2.62_{\pm 0.47}$ (10.40)	$3.82_{\pm 0.41}$ (9.69)	$99.76_{\pm 0.12}$(0.24)	$99.47_{\pm 0.11}$ (0.53)	$92.21_{\pm 0.13}$ (0.59)	$92.03_{\pm 0.37}$ (0.05)	1.77
GA	$9.11_{\pm 0.83}$ (1.98)	$6.91_{\pm 0.96}$ (0.56)	$7.77_{\pm 1.01}$ (5.25)	$8.37_{\pm 1.35}$ (5.14)	$93.08_{\pm 0.93}$ (6.92)	$93.63_{\pm 1.16}$ (6.30)	$86.44_{\pm 1.32}$ (6.36)	$89.22_{\pm 1.59}$ (4.53)	0.31
IU	$1.02_{\pm 0.43}$ (6.11)	$3.07_{\pm 0.50}$ (4.40)	$2.51_{\pm 0.61}$ (9.51)	$6.86_{\pm 0.67}$ (6.65)	$99.14_{\pm 0.03}$(0.86)	$97.35_{\pm 0.31}$ (2.58)	$91.01_{\pm 0.29}$ (1.79)	$89.49_{\pm 0.37}$ (2.49)	2.78

To demonstrate the effectiveness of our methods on a larger dataset, we conducted additional experiments on **ImageNet** [44] with settings consistent with the class-wise forgetting in Tab. 3. As we can see from Tab. A5, sparsity reduces the performance gap between exact unlearning (Retrain) and the approximate unlearning methods (FT and GA). The results are consistent with our observations in other datasets. Note that the 83% model sparsity (ImageNet, ResNet-18) is used to preserve the TA performance after one-shot magnitude (OMP) pruning.

Table A5: Performance overview of MU vs. sparsity on ImageNet considering class-wise forgetting. The content format follows Tab. 3.

MU	UA		MIA-Efficacy		RA		TA		RTE
	DENSE	**83% Sparsity**	DENSE	**83% Sparsity**	DENSE	**83% Sparsity**	DENSE	**83% Sparsity**	(hours)
Class-wise forgetting, ImageNet									
Retrain	100.00	100.00	100.00	100.00	71.75	69.18	69.49	68.86	26.18
FT	63.60 (36.40)	74.66 (25.34)	68.61 (31.39)	81.43 (18.57)	72.45 (0.70)	69.36 (0.18)	69.80 (0.31)	68.77 (0.09)	2.87
GA	85.10 (14.90)	90.21 (9.79)	87.46 (12.54)	94.25 (5.75)	65.93 (5.82)	62.94 (6.24)	64.62 (4.87)	64.65 (4.21)	0.01

Performance of ℓ_1 sparsity-aware MU on additional datasets. As seen in Fig. A2, ℓ_1-sparse MU significantly reduces the gap between approximate and exact unlearning methods across various datasets (CIFAR-100 [41], SVHN [43], ImageNet [44]) in different unlearning scenarios. It notably outperforms other methods in UA and MIA-Efficacy metrics while preserving acceptable RA and TA performances, thus becoming a practical choice for unlearning scenarios. In class-wise and random data forgetting cases, ℓ_1-sparse MU exhibits performance on par with Retrain in UA and MIA-Efficacy metrics. Importantly, the use of ℓ_1-sparse MU consistently enhances forgetting metrics with an insignificant rise in computational cost compared with FT, underscoring its effectiveness and efficiency in diverse unlearning scenarios. For detailed numerical results, please refer to Tab. A6.

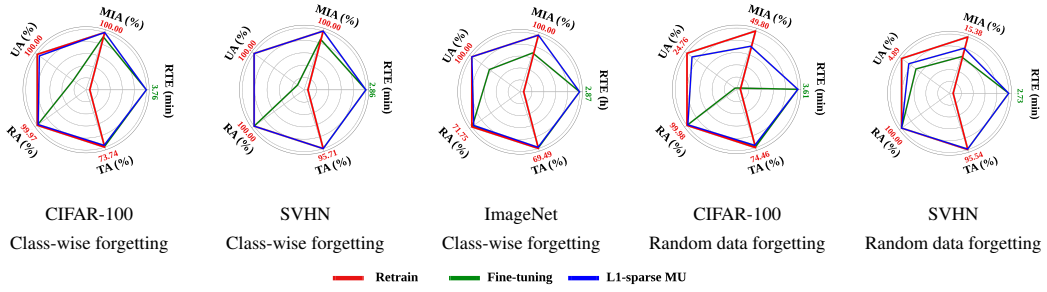

Figure A2: Performance of sparsity-aware unlearning vs. FT and Retrain on class-wise forgetting and random data forgetting under (CIFAR-10, ResNet-18). Each metric is normalized to $[0, 1]$ based on the best result across unlearning methods for ease of visualization, while the actual best value is provided. The figure format is consistent with Fig. 5.

D Broader Impacts and Limitations

Broader impacts. Our study on model sparsity-inspired MU provides a versatile solution to forget arbitrary data points and could give a general solution for dealing with different concerns, such as the model's privacy, efficiency, and robustness. Moreover, the applicability of our method extends beyond these aspects, with potential impacts in the following areas. ① *Regulatory compliance:* Our method enables industries, such as healthcare and finance, to adhere to regulations that require the forgetting of data after a specified period. This capability ensures compliance while preserving the utility and performance of machine learning models. ② *Fairness:* Our method could also play a crucial role in addressing fairness concerns by facilitating the unlearning of biased datasets or subsets. By removing biased information from the training data, our method contributes to mitigating bias in machine learning models, ultimately fostering the development of fairer models. ③ *ML with adaptation and sustainability:* Our method could promote the dynamic adaptation of machine learning models by enabling the unlearning of outdated information, and thus, could enhance the accuracy and relevance of the models to the evolving trends and dynamics of the target domain. This capability fosters sustainability by ensuring that ML models remain up-to-date and adaptable, thus enabling their continued usefulness and effectiveness over time.

Limitations. One potential limitation of our study is the absence of provable guarantees for ℓ_1-sparse MU. Since model sparsification is integrated with model training as a soft regulariza-

Table A6: Performance of sparsity-aware MU vs. Retrain, FT and IU considering class-wise forgetting and random data forgetting, where the table format and setup are consistent with Tab. 3. The unit of RTE is minutes for all datasets, except ImageNet. For ImageNet, indicated by an asterisk ($*$), RTE is measured by hours.

MU	UA	MIA-Efficacy	RA	TA	RTE (min)
Class-wise forgetting, CIFAR-10					
Retrain	100.00±0.00	100.00±0.00	100.00±0.00	94.83±0.11	43.23
FT	$22.53_{\pm 8.16}$ (77.47)	$75.00_{\pm 14.68}$ (25.00)	$99.87_{\pm 0.04}$ (0.13)	$94.31_{\pm 0.19}$ (0.52)	2.52
IU	$87.82_{\pm 2.15}$ (12.18)	$95.96_{\pm 0.21}$ (4.04)	$97.98_{\pm 0.21}$ (2.02)	$91.42_{\pm 0.21}$ (3.41)	3.25
ℓ_1-sparse MU	$100.00_{\pm 0.00}$ (0.00)	$100.00_{\pm 0.00}$ (0.00)	$98.99_{\pm 0.12}$ (1.01)	$93.40_{\pm 0.43}$ (1.43)	2.53
Class-wise forgetting, CIFAR-100					
Retrain	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$99.97_{\pm 0.01}$	$73.74_{\pm 0.19}$	48.45
FT	$26.45_{\pm 6.29}$ (73.55)	$92.44_{\pm 5.93}$ (7.56)	$99.86_{\pm 0.04}$(0.11)	$74.08_{\pm 0.23}$ (0.74)	3.76
IU	$84.12_{\pm 0.34}$ (15.88)	$98.44_{\pm 0.45}$ (1.56)	$96.23_{\pm 0.02}$ (3.74)	$71.24_{\pm 0.22}$ (2.50)	4.30
ℓ_1-sparse MU	$95.67_{\pm 0.53}$ (4.33)	$100.00_{\pm 0.00}$ (0.00)	$98.01_{\pm 0.02}$ (1.96)	$71.35_{\pm 0.22}$ (2.39)	3.79
Class-wise forgetting, SVHN					
Retrain	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$95.71_{\pm 0.12}$	42.84
FT	$11.48_{\pm 8.12}$ (88.52)	$86.12_{\pm 9.62}$ (13.88)	$100.00_{\pm 0.00}$ (0.00)	$95.99_{\pm 0.07}$ (0.28)	2.86
IU	$95.11_{\pm 0.02}$ (4.89)	$99.89_{\pm 0.04}$ (0.11)	$100.00_{\pm 0.00}$ (0.00)	$95.70_{\pm 0.09}$ (0.01)	3.19
ℓ_1-sparse MU	$100.00_{\pm 0.00}$ (0.00)	$100.00_{\pm 0.00}$ (0.00)	$99.99_{\pm 0.01}$ (0.00)	$95.88_{\pm 0.14}$ (0.17)	2.88
Class-wise forgetting, ImageNet					
Retrain	$100.00_{\pm 0.00}$	$100.00_{\pm 0.00}$	$71.75_{\pm 0.45}$	$69.49_{\pm 0.27}$	26.18*
FT	$63.60_{\pm 7.11}$ (36.40)	$68.61_{\pm 9.04}$ (31.39)	$72.45_{\pm 0.16}$ (0.70)	$69.80_{\pm 0.23}$ (0.31)	2.87*
IU	$43.35_{\pm 5.26}$ (56.65)	$60.83_{\pm 6.17}$ (39.17)	$66.28_{\pm 0.77}$ (4.97)	$66.25_{\pm 0.53}$ (3.24)	3.14*
ℓ_1-sparse MU	$99.85_{\pm 0.07}$ (0.15)	$100.00_{\pm 0.00}$ (0.00)	$68.07_{\pm 0.13}$ (3.68)	$68.01_{\pm 0.21}$ (1.48)	2.87*
Random data forgetting, CIFAR-10					
Retrain	$5.41_{\pm 0.11}$	$13.12_{\pm 0.14}$	$100.00_{\pm 0.00}$	$94.42_{\pm 0.09}$	42.15
FT	$6.83_{\pm 0.51}$ (1.42)	$14.97_{\pm 0.62}$ (1.85)	$96.61_{\pm 0.25}$ (3.39)	$90.13_{\pm 0.26}$ (4.29)	2.33
IU	$2.03_{\pm 0.43}$ (3.38)	$5.07_{\pm 0.74}$ (8.05)	$98.26_{\pm 0.29}$ (1.74)	$91.33_{\pm 0.22}$ (3.09)	3.22
ℓ_1-sparse MU	$5.35_{\pm 0.22}$ (0.06)	$12.71_{\pm 0.31}$ (0.41)	$97.39_{\pm 0.19}$ (2.61)	$91.26_{\pm 0.20}$ (3.16)	2.34
Random data forgetting, CIFAR-100					
Retrain	$24.76_{\pm 0.12}$	$49.80_{\pm 0.26}$	$99.98_{\pm 0.02}$	$74.46_{\pm 0.08}$	48.70
FT	$0.78_{\pm 0.34}$ (23.98)	$1.13_{\pm 0.40}$ (48.67)	$99.93_{\pm 0.02}$ (0.05)	$75.14_{\pm 0.09}$ (0.68)	3.74
IU	$1.53_{\pm 0.36}$ (23.23)	$6.58_{\pm 0.42}$ (43.22)	$99.01_{\pm 0.28}$ (0.97)	$71.76_{\pm 0.31}$ (2.70)	3.80
ℓ_1-sparse MU	$20.77_{\pm 0.27}$ (3.99)	$36.80_{\pm 0.44}$ (13.00)	$98.26_{\pm 0.15}$ (1.72)	$71.52_{\pm 0.21}$ (2.94)	3.76
Random data forgetting, SVHN					
Retrain	$4.89_{\pm 0.11}$	$15.38_{\pm 0.14}$	$100.00_{\pm 0.00}$	$95.54_{\pm 0.09}$	42.71
FT	$3.56_{\pm 0.27}$ (1.33)	$10.05_{\pm 0.24}$ (5.33)	$99.89_{\pm 0.04}$ (0.11)	$93.55_{\pm 0.12}$ (1.99)	2.73
IU	$3.48_{\pm 0.13}$ (1.41)	$9.44_{\pm 0.27}$ (5.94)	$96.30_{\pm 0.08}$ (3.70)	$91.59_{\pm 0.11}$ (3.95)	3.21
ℓ_1-sparse MU	$4.06_{\pm 0.14}$ (0.83)	$11.80_{\pm 0.22}$ (3.58)	$99.96_{\pm 0.01}$ (0.04)	$94.98_{\pm 0.03}$ (0.56)	2.73

tion, the lack of formal proof may raise concerns about the reliability and robustness of the approach. Furthermore, while our proposed unlearning framework is generic, its applications have mainly focused on solving computer vision tasks. As a result, its effectiveness in the domain of natural language processing (NLP) remains unverified. This consideration becomes particularly relevant when considering large language models. Therefore, further investigation is necessary for future studies to explore the applicability and performance of the framework in NLP tasks.