Unified Gradient-Based Machine Unlearning with Remain Geometry Enhancement

Zhehao Huang, Xinwen Cheng, JingHao Zheng, Haoran Wang, Zhengbao He, Tao Li, Xiaolin Huang

Shanghai Jiao Tong University

NeurIPS 2024 Spotlight

Backgrounds

- Machine Unlearning (MU) aims to remove the influence of samples from a pre-trained model, ensuring the model behaves as if it has never encountered those samples.
- Existing MU methods are mainly divided into two categories: *exact MU* and *approximate MU*:
  - Exact MU ensures that the parameter distribution of the unlearned model is identical to that of a model trained from scratch without seeing the forgetting samples. The computational cost of retraining in response to every forgetting request is prohibitive.
  - Aproximate MU guides the unlearned model output distribution to approximate the output distribution of RT. Using KL divergence to measure:

$$\min_{\theta} D_{\mathrm{KL}}\left(p_z\left(\theta_*\right) \| p_z(\theta)\right) = \min_{\theta} \int p_z\left(\theta_*\right) \log \left[p_z\left(\theta_*\right) / p_z(\theta)\right] \mathrm{d}\mathcal{D}$$

Revisit Approximate MU Methods via Vanilla Gradient Descent

- The optimization problem of Approximate MU:

$$\theta_{t+1} = \arg\min_{\theta_{t+1}} \underbrace{D_{\mathrm{KL}}\left(p_{z^f}\left(\theta_*\right) \| p_{z^f}\left(\theta_{t+1}\right)\right)}_{(a)} p^f + \underbrace{D_{\mathrm{KL}}\left(p_{z^r}\left(\theta_*\right) \| p_{z^r}\left(\theta_{t+1}\right)\right)}_{(b)} p^r + \frac{1}{\alpha_t} \underbrace{\rho\left(\theta_t, \theta_{t+1}\right)}_{(c)}$$

(a): seeks to eliminate the influence of the target forgetting samples

(b): aims to maintain the performance on the remaining samples

(c): employs the metric to constrain the magnitude of each update

- The vanilla gradient descent for approximate MU in Euclidean distance:

Proposition 1. *Under the Euclidean manifold metric, $\rho(\theta_t, \theta_{t+1}) = \frac{1}{2}\|\theta_t - \theta_{t+1}\|^2$. Assuming that the current model $\theta_t = \arg\min_\theta \mathcal{L}^f(\theta; \varepsilon_t) + \mathcal{L}^r(\theta)$. Let $H_*^f = \nabla^2 \mathcal{L}^f(\theta_*; \mathbf{1})$and $H_*^r = \nabla^2 \mathcal{L}^r(\theta_*)$ denote the Hessian of the retrained model on the forgetting set and the remaining set, respectively. Then, the steepest descent direction that minimizes (2) is approximately:*

$$\theta_{t+1} - \theta_t :\approx -\alpha_t [\underbrace{H_*^f (H_*^r)^{-1}}_{(S)} \underbrace{[-\nabla \mathcal{L}^f(\theta_t; \varepsilon_t)]}_{(F)} p^f + \underbrace{\nabla \mathcal{L}^r(\theta_t)}_{(R)} p^r]. \tag{3}$$

Approximate MU in Remain-preserving Manifold

How to constrain parameter updates to minimally impacts the retained performance?

- The gradient descent for approximate MU in remain-preserving KL divergence:

Proposition 2. *Using the model output KL divergence on the remaining set as the manifold metric, $\rho(\theta_t, \theta_{t+1}) = D_{KL}(p_{z^r}(\theta_t) || p_{z^r}(\theta_{t+1})))$. Assuming that the current model $\theta_t = \arg\min_\theta \mathcal{L}^r(\theta) + \mathcal{L}^f(\theta; \varepsilon_t)$. Let $\tilde{\alpha}_t = \alpha_t p^f / (\alpha_t p^r + 1)$, and $H_t^r = \nabla^2 \mathcal{L}^r(\theta_t)$ represent the Hessian w.r.t. θ_t on the remaining set, then the steepest descent direction that minimizes (2) is approximately:*

$$\theta_{t+1} - \theta_t :\approx -\tilde{\alpha}_t \underbrace{(H_t^r)^{-1}}_{(R)} \underbrace{[H_*^f (H_*^r)^{-1}}_{(S)} \underbrace{[-\nabla \mathcal{L}^f(\theta_t; \varepsilon_t)]]}_{(F)}. \tag{4}$$

Approximate MU Methods	Task Cls	Task Gen	MU components (S)	MU components (F)	MU components (R)	Manifold Metric	Online Hessian
FT [22, 19, 38]	✓	✓			✓	ℓ_2	
GA [20, 21]	✓	✓		✓		ℓ_2	
BT [23]	✓			✓	✓	ℓ_2	
SalUn [26]	✓	✓	✓	✓	✓	ℓ_2	
SA [35]		✓		✓	✓	D_{KL}^r	
SFR-on	✓	✓	✓	✓	✓	D_{KL}^r	✓

Challenges in Hessian Approximation: computationally demanding

Proposed Method

- Implicit Online Hessian Approximation (R-on)

We propose **a fast-slow weight** method for implicitly approximating the desired updates:

$$\min_{\theta_t^f} \mathcal{L}^r\left(\theta_t^f\right) \quad \text{s.t.} \quad \theta_t^f = \theta_t - \beta_t \nabla \mathcal{L}^u\left(\theta_t\right)$$

Proposition 3. *For implicit online Hessian approximation in (5), suppose β_t, δ_t is small, $\beta_t < \sqrt{\delta_t / |\nabla \mathcal{L}^r(\theta_t) - [\nabla \mathcal{L}^r(\theta_t)]^2|}$, $\mathcal{L}^r$ is μ-smooth, i.e., $\|\nabla \mathcal{L}^r(\theta) - \nabla \mathcal{L}^r(\theta')\|_2 \leq \mu \|\theta - \theta'\|_2$, and there exist an ζ_t-neighborhood $\mathcal{N}(\theta_t^r, \zeta_t)$ of the optimal model parameter $\theta_t^r = \arg\min_{\theta_t^f} \mathcal{L}^r(\theta_t^f)$, which includes θ_t and θ_t^f. Then, the iterative update term approximately is,*

$$\theta_t - \theta_t^r :\approx \beta_t^2 \left[\nabla^2 \mathcal{L}^r(\theta_t)\right]^{-1} \nabla \mathcal{L}^u(\theta_t) = \beta_t^2 (H_t^r)^{-1} \nabla \mathcal{L}^u(\theta_t). \tag{6}$$

The model obtained after fine-tuning is approximately equivalent to updating the current model in the Hessian-adjusted unlearning directing.

Proposed Method

- Sample-wise Adaptive Coefficient for Gradient Ascent (F)

  A heuristic estimation for coefficients weighting the unlearning loss.

  Using empirical loss as an evaluation metric for sample contribution.

$$\tilde{\varepsilon}_{t,i} = (1 - \frac{t}{T}) \frac{1/[\ell(\theta_t; z_i^f)]_{\text{detach}}^{\lambda}}{\sum_{z_j^f \in \mathcal{D}^f} 1/[\ell(\theta_t; z_j^f)]_{\text{detach}}^{\lambda}} \times N^f,\ 1 \leq i \leq N^f,$$

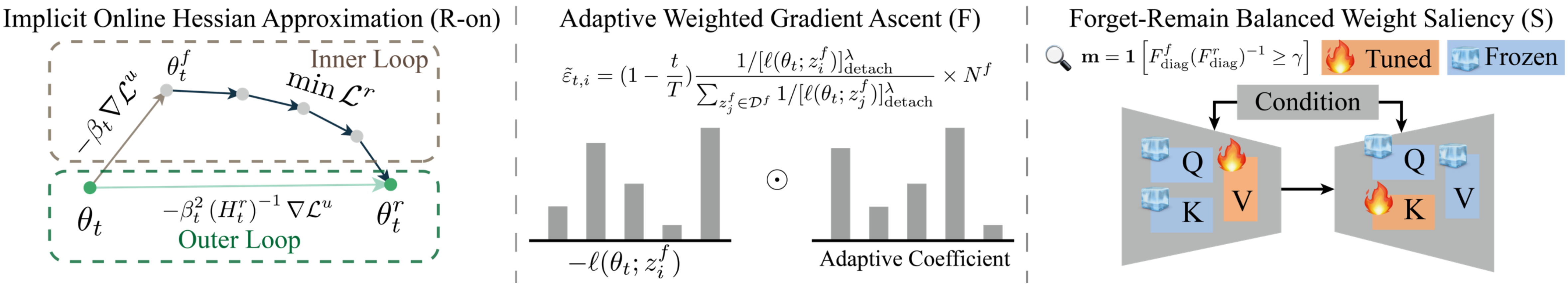
- Forget-Remain Balanced Weight Saliency (S)

  Using the diagonal of the initial model's Fisher information matrix on forgetting and remaining to enhance the unlearning process.

  Focus on the parameters that are crucial for erasing specific samples or concepts.

$$\mathbf{m} = \mathbf{I}\left[F_{\text{diag}}^{f}(F_{\text{diag}}^{r})^{-1} \geq \gamma\right],\ \text{where } F_{\text{diag}}^{f} = [\nabla \mathcal{L}^f(\theta_0)]^2, F_{\text{diag}}^{r} = [\nabla \mathcal{L}^r(\theta_0)]^2.$$

Saliency **F**orgetting in the **R**emain-preserving manifold **on**line (SFR-on)



$$\textbf{Inner Loop}: \min_{\theta_t^f} \mathcal{L}^r(\theta_t^r) \quad \text{s.t. } \theta_t^f = \theta_t - \beta_t[\mathbf{m} \odot (-\nabla \mathcal{L}^f(\theta_t; \tilde{\boldsymbol{\varepsilon}}_t))],$$

$$\textbf{Outer Loop}: \theta_{t+1} = \theta_t - \alpha_t(\theta_t - \theta_t^r) \approx \theta_t - \alpha_t \beta_t^2 (H_t^r)^{-1}[\mathbf{m} \odot (-\nabla \mathcal{L}^f(\theta_t; \tilde{\boldsymbol{\varepsilon}}_t))],$$

- In the inner loop for fast weights, we use adaptive **coefficients to weight** the forgetting gradient ascent with the **weight saliency map** to serve as the unlearning update.
- Slow weights in outer loops update by **linearly interpolating the fine-tuned parameters in weight space**, achieving an estimated steepest descent for approximate MU under the remaining output constraint.
- Our SFR-on does not require adaptation to specific application tasks.

Results on Random Forgetting in Image Classification Tasks

Methods	CIFAR-10 Random Subset Forgetting (10%) FA	RA	TA	MIA	Avg.D ↓	D_{KL} ↓	RTE	TinyImageNet Random Subset Forgetting (10%) FA	RA	TA	MIA	Avg.D ↓	D_{KL} ↓	RTE
RT	$95.62_{\pm0.25}$ (0.00)	$100.00_{\pm0.00}$ (0.00)	$95.34_{\pm0.08}$ (0.00)	$74.84_{\pm0.00}$ (0.00)	0.00	0.10	73.37	$85.29_{\pm0.09}$ (0.00)	$99.55_{\pm0.03}$ (0.00)	$85.49_{\pm0.15}$ (0.00)	$69.30_{\pm0.20}$ (0.00)	0.00	0.18	42.01
FT	$99.90_{\pm0.05}$ (4.28)	$99.99_{\pm0.00}$ (0.01)	$94.94_{\pm0.15}$ (0.39)	$88.25_{\pm0.01}$ (13.42)	4.52	0.26	3.83	$96.45_{\pm0.13}$ (11.16)	$98.29_{\pm0.08}$ (1.26)	$82.46_{\pm0.16}$ (3.03)	$90.00_{\pm0.22}$ (20.70)	9.04	0.60	4.38
GA	$93.91_{\pm1.67}$ (1.71)	$93.76_{\pm1.89}$ (6.24)	$87.00_{\pm1.64}$ (8.34)	$77.19_{\pm0.01}$ (2.35)	4.66	0.36	0.79	$83.28_{\pm4.18}$ (2.01)	$84.55_{\pm4.63}$ (15.00)	$70.98_{\pm3.61}$ (14.51)	$73.86_{\pm3.31}$ (4.56)	9.02	1.09	4.13
RL	$95.99_{\pm0.24}$ (0.38)	$99.98_{\pm0.01}$ (0.02)	$93.85_{\pm0.11}$ (1.48)	$31.44_{\pm0.01}$ (43.40)	11.32	0.34	4.56	$93.35_{\pm0.31}$ (8.06)	$98.15_{\pm0.14}$ (1.40)	$82.98_{\pm0.22}$ (2.51)	$45.29_{\pm1.04}$ (24.00)	9.00	0.47	4.79
SalUn	$100.00_{\pm0.01}$ (4.38)	$99.99_{\pm0.01}$ (0.01)	$94.89_{\pm0.09}$ (0.45)	$67.54_{\pm0.00}$ (7.29)	3.03	0.27	4.58	$95.78_{\pm0.25}$ (10.49)	$98.60_{\pm0.06}$ (0.95)	$83.63_{\pm0.22}$ (1.87)	$51.18_{\pm1.92}$ (18.12)	7.86	0.48	4.88
BT	$98.88_{\pm0.00}$ (3.26)	$99.99_{\pm0.00}$ (0.01)	$94.63_{\pm0.06}$ (0.71)	$61.77_{\pm0.00}$ (13.07)	4.26	0.24	5.56	$93.22_{\pm0.30}$ (7.93)	$97.82_{\pm0.14}$ (1.73)	$83.04_{\pm0.22}$ (2.45)	$47.53_{\pm0.71}$ (21.77)	8.47	0.47	6.79
SCRUB	$99.44_{\pm0.31}$ (3.82)	$99.88_{\pm0.08}$ (0.12)	$94.13_{\pm0.35}$ (1.20)	$87.43_{\pm0.00}$ (12.59)	4.43	0.25	2.56	$97.23_{\pm0.05}$ (11.94)	$98.10_{\pm0.34}$ (1.45)	$82.74_{\pm0.21}$ (2.75)	$81.32_{\pm0.47}$ (12.02)	7.04	0.62	5.49
S F R on														
✓	$96.38_{\pm0.35}$ (0.76)	$99.66_{\pm0.01}$ (0.34)	$91.96_{\pm0.31}$ (3.38)	$83.16_{\pm0.47}$ (8.32)	3.20	0.32	3.13	$89.90_{\pm0.39}$ (4.61)	$94.05_{\pm0.19}$ (5.50)	$77.98_{\pm0.72}$ (7.51)	$78.16_{\pm0.93}$ (8.86)	6.62	0.73	6.10
✓ ✓	$96.84_{\pm0.50}$ (1.22)	$99.92_{\pm0.21}$ (0.08)	$94.18_{\pm0.28}$ (1.16)	$80.38_{\pm0.25}$ (5.54)	2.00	0.23	2.12	$93.42_{\pm0.16}$ (8.13)	$98.92_{\pm0.04}$ (0.63)	$83.45_{\pm0.21}$ (2.04)	$81.84_{\pm0.77}$ (12.54)	5.83	0.73	4.02
✓ ✓ ✓	$96.16_{\pm0.72}$ (0.54)	$99.98_{\pm0.20}$ (0.02)	$94.24_{\pm0.30}$ (1.10)	$70.64_{\pm0.26}$ (4.20)	1.47	0.20	2.12	$95.51_{\pm0.25}$ (10.22)	$98.79_{\pm0.04}$ (0.76)	$83.11_{\pm0.13}$ (2.38)	$64.00_{\pm0.87}$ (5.30)	4.67	0.45	4.02
✓ ✓ ✓ ✓	$96.58_{\pm0.77}$ (0.96)	$99.88_{\pm0.16}$ (0.12)	$94.19_{\pm0.33}$ (1.15)	$72.26_{\pm0.01}$ (2.58)	1.20	0.15	2.80	$97.02_{\pm0.16}$ (11.73)	$99.18_{\pm0.05}$ (0.37)	$84.00_{\pm0.18}$ (1.49)	$71.09_{\pm0.76}$ (1.79)	3.85	0.44	4.21

- **SFR-on** *most closely aligns with RT* in the averaging metric disparity and exhibits the *smallest output KL divergences* w.r.t. RT.
- Replacing (R) with our (R-on) remarkably improves the image fidelity of the remaining classes, but the forgetting class images still show low-quality textures.
- Our (F) and (S) effectively direct the unlearning process towards the approximate MU, ensuring that the performance of the unlearned models closely mirrors that of RT.

Results on Class-forgetting in Image Generation Tasks

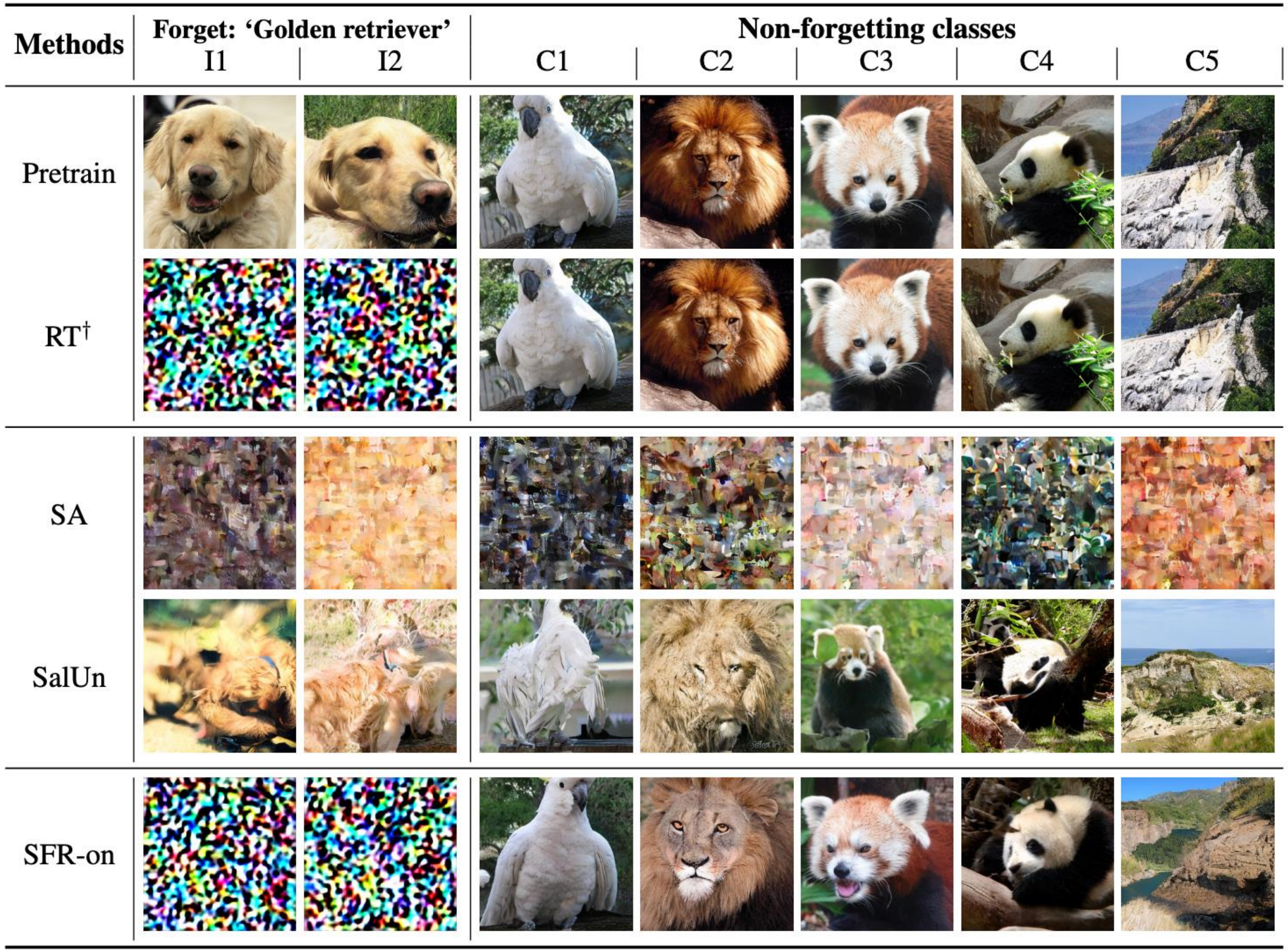
- Class-wise forgetting on CIFAR-10 using DDPM
- Our **SFR-on** effectively removes the ‘cat’ class by yielding high-quality pictures without discernible semantics
- Our **SFR-on** maintains the high fidelity of images across non-forgetting classes.

Methods	**CIFAR-10 Class-wise Forgetting** Automobile FA ↓	Automobile FID ↓	Cat FA ↓	Cat FID ↓	Dog FA ↓	Dog FID ↓	Horse FA ↓	Horse FID ↓	Truck FA ↓	Truck FID ↓	Steps
SA	**0.00**	23.56	14.20	21.34	8.60	21.19	**0.00**	21.13	**0.00**	29.04	10000
SalUn	0.20	21.23	**1.40**	20.29	**0.00**	20.18	0.60	20.70	0.80	**20.45**	1000
SFR-on	**0.00**	**20.70**	7.40	**18.44**	0.20	**18.89**	**0.00**	**19.93**	**0.00**	20.61	50

Methods	**ImageNet Class-wise Forgetting** Cacatua galerita FA ↓	Cacatua galerita FID ↓	Golden retriever FA ↓	Golden retriever FID ↓	White wolf FA ↓	White wolf FID ↓	Arctic fox FA ↓	Arctic fox FID ↓	Otter FA ↓	Otter FID ↓	Steps
SA	**0.00**	348.75	**0.00**	298.97	**0.00**	45.89	**0.00**	393.91	29.8	321.21	10000
SalUn	91.21	18.47	46.09	25.28	**0.00**	**15.16**	45.90	408.07	87.50	19.69	10000
SFR-on	**0.00**	**13.59**	**0.00**	**17.76**	**0.00**	23.28	**0.00**	**16.12**	**0.00**	**16.43**	500

Results on Class-forgetting in Image Generation Tasks

- Class-wise forgetting in image generations of ImageNet with DiT.
- Our **SFR-on** successfully forgetting the target class without degrading the general generative capability.



Conclusion

- We provide a novel perspective to unify previous approaches by decomposing the vanilla gradient descent direction of approximate MU into three components: weighted forgetting gradient ascent, remaining gradient descent, and a weight saliency matrix.
- We derive the steepest descent direction for approximate MU on the remain-preserved manifold.
- We propose a fast-slow weight method to implicitly approximate online Hessian-modulated salient forgetting updates.
- We conduct experiments on a wide range of CV unlearning tasks across multiple datasets and models of different architectures, verifying the effectiveness and efficiency of our method.

Thanks