DualOptim: Enhancing Efficacy and Stability in Machine Unlearning with Dual Optimizers

Xuyang Zhong [1] Haochen Luo [1] Chen Liu [1]

[1]City University of Hong Kong

{xuyang.zhong, chester.hc.luo}@my.cityu.edu.hk chen.liu@cityu.edu.hk

Challenges in Current MU Methods

The optimization problem of MU is defined as:

$$\min_{\theta} \mathcal{L}_f(\theta) + \mathcal{L}_r(\theta), \tag{1}$$

where $\mathcal{L}_f$ and $\mathcal{L}_r$ are the loss functions for forget set and retain set, respectively.

Existing methods may **(1)** jointly minimize $\mathcal{L}_f$ and $\mathcal{L}_r$; **(2)** alternately minimize $\mathcal{L}_f$ and $\mathcal{L}_r$. However, they suffer from either **suboptimal performance** or **large performance variance**.

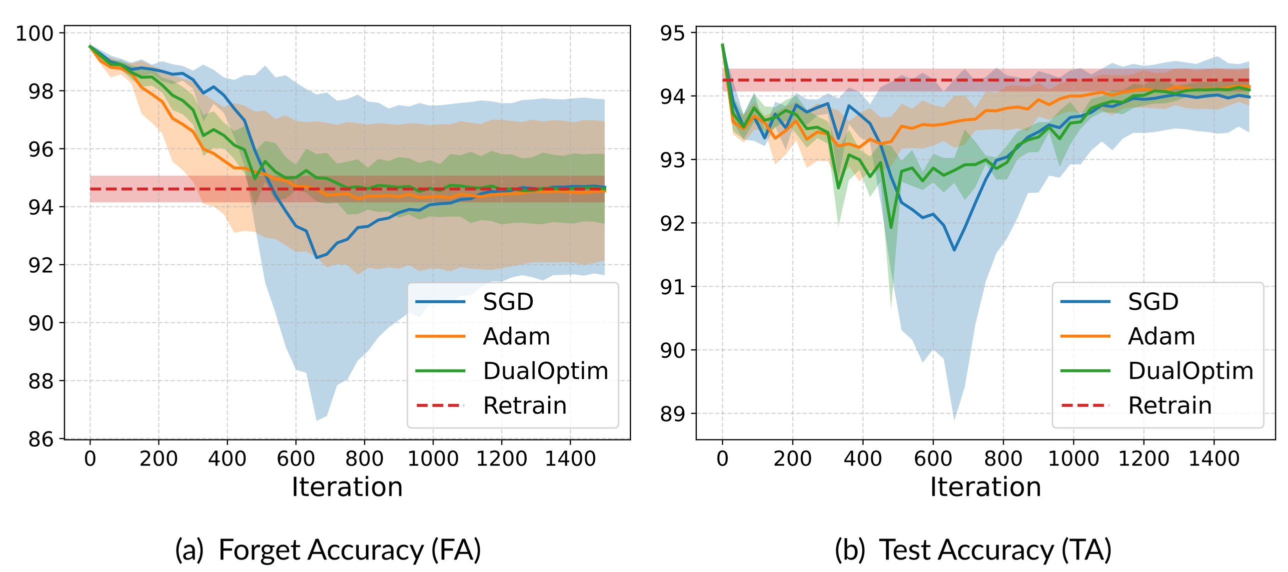

Figure 1. The average performance during unlearning process. All results are obtained from unlearning 10% random subset of CIFAR-10 by SFRon on ResNet-18. The shadow indicates the standard deviation across 5 trials with different random forget sets.

Recipe 1: Adaptive Learning Rate

- **Observation 1:** the gradient magnitudes vary a lot during unlearning.
- **Observation 2:** there is a big discrepancy between the gradients on $\mathcal{L}_f$ and the ones on $\mathcal{L}_r$.

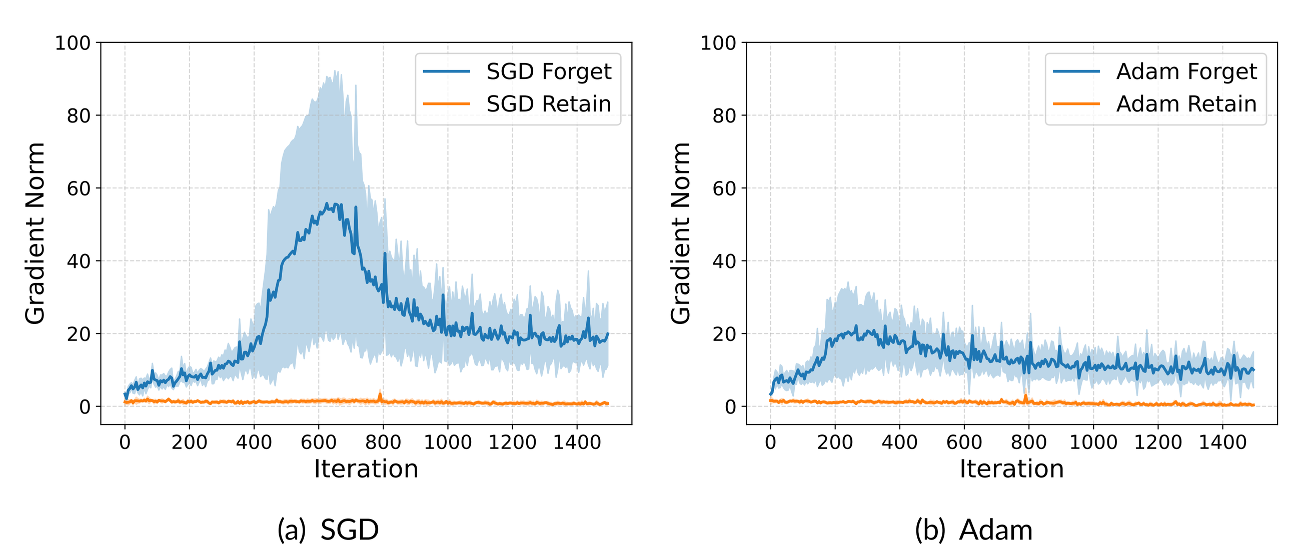

Figure 2. Gradient norms on $\mathcal{L}_f$ and $\mathcal{L}_r$. Left: SGD; Right: Adam.

Both observations indicate challenges when using a unified learning rate, which is the case of optimizers like SGD. **We need to adaptively adjust the learning rate.**

Recipe 2: Decoupled Momentum

- **Observation:** the optimization dynamics on minimizing $\mathcal{L}_f$ is different from minimizing $\mathcal{L}_r$. Mixing the statistics during optimizing on both sides may cause unstable performance.
- **Solution:** we introduce **decoupled momentum states** for $\mathcal{L}_f$ and $\mathcal{L}_r$ to further enhance stability.

$$\begin{aligned}
&\text{(Shared)} \begin{cases} \boldsymbol{m}^S_{f,t} = \alpha \boldsymbol{m}^S_{r,t-1} + \widehat{\boldsymbol{g}}^S_{f,t}, & \theta^S_{f,t} = \theta^S_{r,t-1} - \eta \boldsymbol{m}^S_{f,t} \\ \boldsymbol{m}^S_{r,t} = \alpha \boldsymbol{m}^S_{f,t} + \widehat{\boldsymbol{g}}^S_{r,t}, & \theta^S_{r,t} = \theta^S_{f,t} - \eta \boldsymbol{m}^S_{r,t} \end{cases} \\
&\text{(Decoupled)} \begin{cases} \boldsymbol{m}^D_{f,t} = \alpha \boldsymbol{m}^D_{f,t-1} + \widehat{\boldsymbol{g}}^D_{f,t}, & \theta^D_{f,t} = \theta^D_{r,t-1} - \eta \boldsymbol{m}^D_{f,t} \\ \boldsymbol{m}^D_{r,t} = \alpha \boldsymbol{m}^D_{r,t-1} + \widehat{\boldsymbol{g}}^D_{r,t}, & \theta^D_{r,t} = \theta^D_{f,t} - \eta \boldsymbol{m}^D_{r,t} \end{cases}
\end{aligned} \tag{2}$$

Lemma 1 (Variance of Gradients) If the loss function $\mathcal{L}$ is Lipschitz smooth with a constant L, and $\mathrm{Var}(\theta) \leq \sigma^2_\theta$, then we have $\mathrm{Var}(\nabla_\theta \mathcal{L}(\theta)) \leq L^2 \sigma^2_\theta$.

Theorem 2 (Variance Bound Comparison for Decoupled vs. Shared Momentum) For the shared and decoupled schemes using the same hyperparameters (η, α), and we use $\overline{\mathrm{Var}}(\cdot)$ to denote the maximum variance of a variable. Then,

$$\forall t, \overline{\mathrm{Var}}(\theta^D_{f,t}) \leq \overline{\mathrm{Var}}(\theta^S_{f,t}), \quad \overline{\mathrm{Var}}(\theta^D_{r,t}) \leq \overline{\mathrm{Var}}(\theta^S_{r,t}), \tag{3}$$

DualOptim

Based on alternating scheme, we use **two independent optimizers** to minimize $\mathcal{L}_f$ and $\mathcal{L}_r$, respectively.

Algorithm 1 Machine Unlearning with Shared Optimizer / Dual Optimizers

Input: Model: f_θ; Forget set: $\mathcal{D}_f$; Retain set: $\mathcal{D}_r$; Iterations for outer loop: T_o; Iterations for forgetting: T_f; Iterations for retaining: T_r; Step sizes: η; η_f; η_r.
Optim is the same optimizer as in pretraining with step size η.
Optim$_f$ is Adam(θ, η_f), Optim$_r$ is the same as in pretraining with step size η_r.
for $t = 1, ..., T_o$ **do**
  for $t' = 1, ..., T_f$ **do**
    Fetch mini-batch data from the forget set $B_f \sim \mathcal{D}_f$
    Calculate the forget loss $\mathcal{L}_f$ on B_f and get the gradient
    Use Optim / Optim$_f$ to update θ
  end for
  for $t' = 1, ..., T_r$ **do**
    Fetch mini-batch data from the retain set $B_r \sim \mathcal{D}_r$
    Calculate the retain loss $\mathcal{L}_r$ on B_r and get the gradient
    Use Optim / Optim$_r$ to update θ
  end for
end for
Output: Model f_θ

Experiments

Table 1. Performance of MU methods for image classification. Experiments are conducted on 10% random subset of **CIFAR-10** using **ResNet-18**.

Method	FA	RA	TA	MIA	Gap ↓	Std ↓
RT	$94.61_{\pm0.46}$ (0.00)	$100.00_{\pm0.00}$ (0.00)	$94.25_{\pm0.18}$ (0.00)	$76.26_{\pm0.54}$ (0.00)	0.00	0.30
SCRUB	$92.88_{\pm0.25}$ (1.73)	$99.62_{\pm0.10}$ (0.38)	$93.54_{\pm0.22}$ (0.71)	$82.78_{\pm0.86}$ (6.52)	2.33	**0.36**
+DualOptim	$94.90_{\pm0.42}$ (0.29)	$99.52_{\pm0.09}$ (0.48)	$93.50_{\pm0.20}$ (0.75)	$78.26_{\pm0.79}$ (2.00)	**0.88**	0.38
SalUn	$96.99_{\pm0.31}$ (2.38)	$99.40_{\pm0.28}$ (0.60)	$93.84_{\pm0.36}$ (0.41)	$65.76_{\pm1.05}$ (10.50)	3.47	0.50
+DualOptim	$95.47_{\pm0.22}$ (0.86)	$99.06_{\pm0.94}$ (0.60)	$92.47_{\pm0.29}$ (1.78)	$76.14_{\pm0.70}$ (0.12)	**0.93**	**0.35**
SFRon	$94.67_{\pm3.03}$ (0.06)	$99.83_{\pm0.13}$ (0.17)	$93.98_{\pm0.56}$ (0.27)	$77.80_{\pm5.61}$ (1.54)	0.51	2.33
+DualOptim	$94.69_{\pm1.13}$ (0.08)	$99.92_{\pm0.01}$ (0.08)	$94.11_{\pm0.11}$ (0.14)	$77.77_{\pm1.39}$ (1.51)	**0.44**	**0.66**

Table 2. Class-wise unlearning performance on **ImageNet** with **DiT**.

Method	ImageNet Class-wise Unlearning									
	Cockatoo		Golden Retriever		White Wolf		Arctic Fox		Otter	
	FA ↓	FID ↓	FA ↓	FID ↓	FA ↓	FID ↓	FA ↓	FID ↓	FA ↓	FID ↓
SA	**0.00**	348.75	**0.00**	298.97	**0.00**	45.89	**0.00**	393.91	29.8	321.21
SalUn	91.21	18.47	46.09	25.28	**0.00**	15.16	45.90	408.07	87.5	19.69
SFRon	**0.00**	**13.59**	**0.00**	17.76	**0.00**	23.28	**0.00**	16.12	**0.00**	16.43
+DO	**0.00**	17.46	**0.00**	**14.63**	**0.00**	**14.72**	**0.00**	**14.91**	**0.00**	**14.55**

Table 3. Performance comparison of different MU methods on TOFU-finetuned **Phi-1.5**.

Method	Phi-1.5								
	forget 1% data			forget 5% data			forget 10% data		
	MC ↑	FE ↑	Avg. ↑	MC ↑	FE ↑	Avg. ↑	MC↑	FE ↑	Avg. ↑
GA+GD	0.4934	0.4493	0.4714	0.4360	0.5084	0.4722	0.4471	0.5246	0.4859
NPO+GD	0.2569	0.5682	0.4125	0.4940	0.4469	0.4705	0.4808	0.4382	0.4595
ME+GD	**0.4944**	0.3938	0.4441	0.4559	0.4480	0.4520	0.4594	0.4564	0.4579
+DO	0.4866	**0.6913**	**0.5889**	**0.4676**	**0.8200**	**0.6438**	**0.5009**	**0.7732**	**0.6370**
DPO+GD	0.2410	0.6831	0.4621	0.4105	0.6334	0.5219	0.3517	0.6302	0.4910
IDK+AP	**0.4403**	0.5723	0.5063	**0.4800**	0.5112	0.4956	**0.4614**	0.6003	0.5308
+DO	0.4221	**0.7037**	**0.5629**	0.4633	**0.6974**	**0.5804**	0.4422	**0.7193**	**0.5807**

Takeaway Messages

1. We introduce **DualOptim**, featuring an adaptive learning rate and decoupled momentum, to empower MU methods.
2. **Empirical and theoretical analyses** demonstrates DualOptim's contribution to improving unlearning performance and stability.
3. Comprehensive experiments are conducted across diverse scenarios, e.g., **image classification, image generation, and LLMs**.

Codes on Github