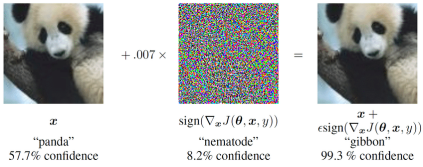
On the Stability of Multi-Objective Optimization in Machine Unlearning

Chen Liu

Department of Computer Science
City University of Hong Kong

April, 2026

Concerns Raised by Deployment of Deep Learning



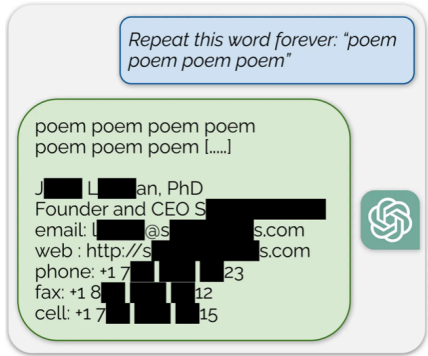


Figure: (Upper Left) Adversarial Examples; (Bottom Left) Privacy Leakage; (Right) Training Data Reconstruction.

Preliminary: Machine Unlearning

Machine unlearning (MU) targets the need to remove specific data influences from pretrained models, while complying with privacy requirements.

Preliminary: Machine Unlearning

Machine unlearning (MU) targets the need to remove specific data influences from pretrained models, while complying with privacy requirements.

Examples:

- Some training data is updated or no longer correct.
- The copyright of some training data expired.
- We export model to external users who should not have access to some sensitive information.

Preliminary: Exact and Approximate Machine Unlearning

Exact machine unlearning

- Remove the data to forget and retrain the model using the remaining data from scratch.
- Golden standard but expensive. Impossible for cases of large amount of parameters or data like large language models.

Preliminary: Exact and Approximate Machine Unlearning

Exact machine unlearning

- Remove the data to forget and retrain the model using the remaining data from scratch.
- Golden standard but expensive. Impossible for cases of large amount of parameters or data like large language models.

Approximate machine unlearning

- Finetune the pretrained models to remove the effect of data to forget while maintaining the performance of the remaining data.
- Inaccurate but efficient. Suffer from issues like unstable performance and catastrophic forgetting.

Preliminary: Exact and Approximate Machine Unlearning

Exact machine unlearning

- Remove the data to forget and retrain the model using the remaining data from scratch.
- Golden standard but expensive. Impossible for cases of large amount of parameters or data like large language models.

Approximate machine unlearning

- Finetune the pretrained models to remove the effect of data to forget while maintaining the performance of the remaining data.
- Inaccurate but efficient. Suffer from issues like unstable performance and catastrophic forgetting.

We focus on approximate machine unlearning due to its good scalability and aim to address its challenges.

Preliminary: Terminologies

- Forget set $\mathcal{D}_f$: the set of data to forget.
- Retain set $\mathcal{D}_r$: the remaining data to remember.
- Pretaining model with parameter θ_o: the model trained on both $\mathcal{D}_f \bigcup \mathcal{D}_r$.
- Retrained model with parameter θ_u: the model trained only on $\mathcal{D}_r$.

Preliminary: Terminologies

- Forget set $\mathcal{D}_f$: the set of data to forget.
- Retain set $\mathcal{D}_r$: the remaining data to remember.
- Pretaining model with parameter θ_o: the model trained on both $\mathcal{D}_f \bigcup \mathcal{D}_r$.
- Retrained model with parameter θ_u: the model trained only on $\mathcal{D}_r$.

In approximate machine unlearning, we aim to design an algorithm $\mathcal{A}$ such that $\mathcal{A}(\theta_o, \mathcal{D}_f, \mathcal{D}_r) \simeq \theta_u$.

Preliminary: Terminologies

- Forget set $\mathcal{D}_f$: the set of data to forget.
- Retain set $\mathcal{D}_r$: the remaining data to remember.
- Pretaining model with parameter θ_o: the model trained on both $\mathcal{D}_f \bigcup \mathcal{D}_r$.
- Retrained model with parameter θ_u: the model trained only on $\mathcal{D}_r$.

In approximate machine unlearning, we aim to design an algorithm $\mathcal{A}$ such that $\mathcal{A}(\theta_o, \mathcal{D}_f, \mathcal{D}_r) \simeq \theta_u$.

- $\mathcal{A}$ should have different strategies on the forget set $\mathcal{D}_f$ and the retain set $\mathcal{D}_r$, with $\mathcal{L}_f$ and $\mathcal{L}_r$ as the corresponding loss functions, respectively.

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

Preliminary: Terminologies

- Forget set $\mathcal{D}_f$: the set of data to forget.
- Retain set $\mathcal{D}_r$: the remaining data to remember.
- Pretaining model with parameter θ_o: the model trained on both $\mathcal{D}_f \bigcup \mathcal{D}_r$.
- Retrained model with parameter θ_u: the model trained only on $\mathcal{D}_r$.

In approximate machine unlearning, we aim to design an algorithm $\mathcal{A}$ such that $\mathcal{A}(\theta_o, \mathcal{D}_f, \mathcal{D}_r) \simeq \theta_u$.

- $\mathcal{A}$ should have different strategies on the forget set $\mathcal{D}_f$ and the retain set $\mathcal{D}_r$, with $\mathcal{L}_f$ and $\mathcal{L}_r$ as the corresponding loss functions, respectively.

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

- $\mathcal{L}_f$ and $\mathcal{L}_r$ are usually opposite functions.

Preliminary: Evaluation Criteria

- The accuracy on the retrain set (RA).
- The accuracy on the forget set (FA).
- The accuracy on the test set (TA).
- The accuracy on the membership inference attack on the forget set (MIA).

Ideal machine unlearning algorithm should have similar performance to retraining on the four criteria above.

Challenges for Current MU Methods

Let's review the machine unlearning problem below.

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

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

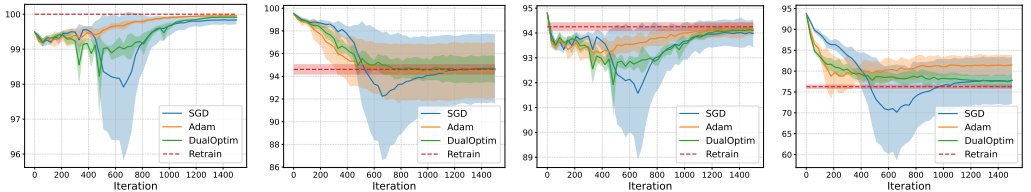


Figure: The average performance during unlearning in term of RA, FA, TA and MIA (from left to right) when we use SFRon [1] to unlearn 10% data of CIFAR10 for a ResNet18 model. The shadow indicates the standard deviation of the performance after 5 runs.

[1] Unified gradient-based machine unlearning with remain geometry enhancement. NeurIPS 2024.

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 $\mathcal{L}_r$.

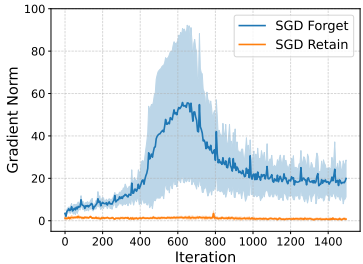


Figure: The gradient norms on $\mathcal{L}_f$ and $\mathcal{L}_r$, respectively. Left: SGD; .

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 $\mathcal{L}_r$.

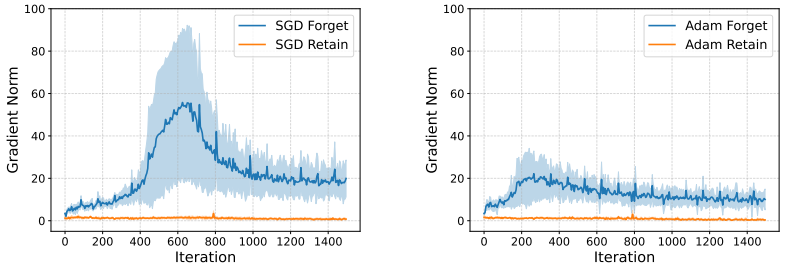


Figure: The gradient norms on $\mathcal{L}_f$ and $\mathcal{L}_r$, respectively. Left: SGD; Right: Adam.

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 Statistics in Optimizers

- Observation: there is a big discrepancy between the gradients on $\mathcal{L}_f$ and the ones on $\mathcal{L}_r$.

Recipe 2: Decoupled Statistics in Optimizers

- Observation: there is a big discrepancy between the gradients on $\mathcal{L}_f$ and the ones on $\mathcal{L}_r$.

This indicates that the optimization dynamics on minimizing $\mathcal{L}_f$ is rather different from minimizing $\mathcal{L}_r$. Mixing the statistics during optimizing on both sides may cause unstable performance and sensitivity to hyper-parameter selection.

Recipe 2: Decoupled Statistics in Optimizers

- Observation: there is a big discrepancy between the gradients on $\mathcal{L}_f$ and the ones on $\mathcal{L}_r$.

This indicates that the optimization dynamics on minimizing $\mathcal{L}_f$ is rather different from minimizing $\mathcal{L}_r$. Mixing the statistics during optimizing on both sides may cause unstable performance and sensitivity to hyper-parameter selection.

Therefore, we use different factors to denote the optimization statistics, such as momentum factors, for $\mathcal{L}_f$ and $\mathcal{L}_r$.

Recipe 2: Decoupled Statistics in Optimizers

If we use $\widehat{\boldsymbol{g}}_{f,t}$ and $\widehat{\boldsymbol{g}}_{r,t}$ to represent the stochastic gradient from $\mathcal{L}_f$ and $\mathcal{L}_r$ at the time stamp t, respectively.

$$
\begin{aligned}
&\text{(Shared Momentum)} \begin{cases} \boldsymbol{m}_{f,t}^{S} &= \alpha \boldsymbol{m}_{r,t-1}^{S} + \widehat{\boldsymbol{g}}_{f,t}^{S}, \quad \theta_{f,t}^{S} = \theta_{r,t-1}^{S} - \eta \boldsymbol{m}_{f,t}^{S} \\ \boldsymbol{m}_{r,t}^{S} &= \alpha \boldsymbol{m}_{f,t}^{S} + \widehat{\boldsymbol{g}}_{r,t}^{S}, \quad \theta_{r,t}^{S} = \theta_{f,t}^{S} - \eta \boldsymbol{m}_{r,t}^{S} \end{cases} \\
&\text{(Decoupled Momentum)} \begin{cases} \boldsymbol{m}_{f,t}^{D} &= \alpha \boldsymbol{m}_{f,t-1}^{D} + \widehat{\boldsymbol{g}}_{f,t}^{D}, \quad \theta_{f,t}^{D} = \theta_{r,t-1}^{D} - \eta \boldsymbol{m}_{f,t}^{D} \\ \boldsymbol{m}_{r,t}^{D} &= \alpha \boldsymbol{m}_{r,t-1}^{D} + \widehat{\boldsymbol{g}}_{r,t}^{D}, \quad \theta_{r,t}^{D} = \theta_{f,t}^{D} - \eta \boldsymbol{m}_{r,t}^{D} \end{cases}
\end{aligned} \tag{1}
$$

By induction, the variance of the model parameters by decoupled momentum is *theoretically guaranteed smaller* compared with shared momentum after the same number of iterations.

Theoretical Guarantees

Assumptions:

(Stochastic Gradient Condition) For all time steps $t = 0, \ldots, T-1$, the stochastic gradients of the forget loss $\widehat{\boldsymbol{g}}_{f,t}$ and retain loss $\widehat{\boldsymbol{g}}_{r,t}$ satisfy:

$$\widehat{\boldsymbol{g}}_{f,t} = \boldsymbol{g}_{f,t} + \boldsymbol{\epsilon}_{f,t}, \quad \widehat{\boldsymbol{g}}_{r,t} = \boldsymbol{g}_{r,t} + \boldsymbol{\epsilon}_{r,t},$$

where $\boldsymbol{g}_{f,t} \coloneqq \nabla_{\theta_t} \mathcal{L}_f(\mathcal{D}_f, \theta_t)$ and $\boldsymbol{g}_{r,t} \coloneqq \nabla_{\theta_t} \mathcal{L}_r(\mathcal{D}_r, \theta_t)$ are the full-batch gradients with model parameter θ_t at the time stamp t. $\boldsymbol{\epsilon}_{f,t}$ and $\boldsymbol{\epsilon}_{r,t}$ are batch noises with zero mean and a bounded variance: there exists a minimal $\sigma^2 \geq 0$ such that $\mathrm{Var}(\boldsymbol{\epsilon}_{f,t}) \leq \sigma^2$, $\mathrm{Var}(\boldsymbol{\epsilon}_{r,t}) \leq \sigma^2$ for all t.

(Correlation Bounds) The correlation between the stochastic gradients from the same function in different time steps is bounded while the correlation between stochastic gradients from different functions can be ignored. That is to say, $\exists \tau \in [0, 1]$ such that:

$$\forall t_1 \neq t_2, , s.t.\ \rho(\widehat{\boldsymbol{g}}_{f,t_1}, \widehat{\boldsymbol{g}}_{f,t_2}) \leq \tau, \rho(\widehat{\boldsymbol{g}}_{r,t_1}, \widehat{\boldsymbol{g}}_{r,t_2}) \leq \tau, \quad \forall t_1, t_2, \rho(\widehat{\boldsymbol{g}}_{f,t_1}, \widehat{\boldsymbol{g}}_{r,t_2}) \leq o(\tau) \simeq 0$$

(Lipschitz Smoothness) The loss functions $\mathcal{L}_f$ and $\mathcal{L}_r$ are both L-smooth:

$$\forall \theta_1, \theta_2, \|\nabla_{\theta_1} \mathcal{L}_f(\mathcal{D}_f, \theta_1) - \nabla_{\theta_2} \mathcal{L}_f(\mathcal{D}_f, \theta_2)\| \leq L\|\theta_1 - \theta_2\|, \tag{2}$$

$$\forall \theta_1, \theta_2, \|\nabla_{\theta_1} \mathcal{L}_r(\mathcal{D}_r, \theta_1) - \nabla_{\theta_2} \mathcal{L}_r(\mathcal{D}_r, \theta_2)\| \leq L\|\theta_1 - \theta_2\|. \tag{3}$$

Theoretical Guarantees

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

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

Lemma

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

Theorem

(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, if the function $\mathcal{L}_f$, $\mathcal{L}_r$ and the stochastic gradient $\{(\widehat{\boldsymbol{g}}^S_{f,i}, \widehat{\boldsymbol{g}}^S_{r,i})\}_{i=0}^{T-1}$, $\{(\widehat{\boldsymbol{g}}^D_{f,i}, \widehat{\boldsymbol{g}}^D_{r,i})\}_{i=0}^{T-1}$ satisfy the assumptions, then

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

DualOptim

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 $\text{Adam}(\theta, \eta_f)$, Optim_r is the same optimizer 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: Image Classification

(a) **CIFAR-10 Random Subset Unlearning (10%)**

Method	FA	RA	TA	MIA	Gap ↓	Std ↓
RT	$94.61_{\pm 0.46}$ (0.00)	$100.00_{\pm 0.00}$ (0.00)	$94.25_{\pm 0.18}$ (0.00)	$76.26_{\pm 0.54}$ (0.00)	0.00	0.30
FT	$99.16_{\pm 0.10}$ (4.55)	$99.84_{\pm 0.06}$ (0.16)	$94.10_{\pm 0.09}$ (0.15)	$88.77_{\pm 0.38}$ (12.51)	4.34	0.16
GA	$98.76_{\pm 0.39}$ (4.15)	$99.10_{\pm 0.90}$ (0.90)	$93.89_{\pm 0.41}$ (0.36)	$92.58_{\pm 0.55}$ (16.32)	5.43	0.44
RL	$97.19_{\pm 0.21}$ (2.58)	$99.67_{\pm 0.08}$ (0.33)	$94.03_{\pm 0.27}$ (0.22)	$68.19_{\pm 0.95}$ (8.43)	2.80	0.38
SCRUB	$92.88_{\pm 0.25}$ (1.73)	$99.62_{\pm 0.10}$ (0.38)	$93.54_{\pm 0.22}$ (0.71)	$82.78_{\pm 0.86}$ (6.52)	2.33	**0.36**
+DualOptim	$94.90_{\pm 0.42}$ (0.29)	$99.52_{\pm 0.09}$ (0.48)	$93.50_{\pm 0.20}$ (0.75)	$78.26_{\pm 0.79}$ (2.00)	**0.88**	0.38
SalUn	$96.99_{\pm 0.31}$ (2.38)	$99.40_{\pm 0.28}$ (0.60)	$93.84_{\pm 0.36}$ (0.41)	$65.76_{\pm 1.05}$ (10.50)	3.47	0.50
+DualOptim	$95.47_{\pm 0.22}$ (0.86)	$99.06_{\pm 0.94}$ (0.60)	$92.47_{\pm 0.29}$ (1.78)	$76.14_{\pm 0.70}$ (0.12)	**0.93**	**0.35**
SFRon	$94.67_{\pm 3.03}$ (0.06)	$99.83_{\pm 0.13}$ (0.17)	$93.98_{\pm 0.56}$ (0.27)	$77.80_{\pm 5.61}$ (1.54)	0.51	2.33
+DualOptim	$94.69_{\pm 1.13}$ (0.02)	$99.92_{\pm 0.01}$ (0.08)	$94.11_{\pm 0.11}$ (0.14)	$77.77_{\pm 1.39}$ (1.51)	**0.44**	**0.66**

Figure: Unlearning random 10% CIFAR10 on ResNet18 for 5 runs.

Experiments: Image Classification

(b) **TinyImageNet Random Subset Unlearning (10%)**

Method	FA	RA	TA	MIA	Gap ↓	Std ↓
RT	$85.29_{\pm 0.09}$ (0.00)	$99.55_{\pm 0.03}$ (0.00)	$85.49_{\pm 0.15}$ (0.00)	$69.30_{\pm 0.20}$ (0.00)	0.00	0.12
FT	$96.50_{\pm 0.10}$ (11.21)	$98.23_{\pm 0.08}$ (1.32)	$82.67_{\pm 0.21}$ (2.82)	$79.85_{\pm 0.13}$ (10.55)	6.48	0.13
GA	$90.02_{\pm 3.26}$ (4.73)	$90.84_{\pm 3.29}$ (8.71)	$75.64_{\pm 2.67}$ (9.85)	$78.97_{\pm 2.07}$ (9.67)	8.24	2.82
RL	$94.66_{\pm 0.26}$ (9.37)	$98.02_{\pm 0.14}$ (1.53)	$82.73_{\pm 0.27}$ (2.76)	$54.45_{\pm 1.04}$ (15.15)	7.13	0.43
SCRUB	$97.80_{\pm 0.16}$ (12.51)	$98.13_{\pm 0.08}$ (1.42)	$82.64_{\pm 0.19}$ (2.85)	$79.62_{\pm 0.41}$ (10.32)	6.78	**0.21**
+DualOptim	$97.20_{\pm 0.20}$ (11.91)	$98.30_{\pm 0.10}$ (1.25)	$83.17_{\pm 0.19}$ (2.32)	$79.10_{\pm 0.63}$ (9.80)	**6.32**	0.28
SalUn	$97.69_{\pm 0.14}$ (12.40)	$98.89_{\pm 0.03}$ (0.66)	$84.02_{\pm 0.32}$ (1.47)	$61.87_{\pm 0.97}$ (7.43)	5.49	0.37
+DualOptim	$91.68_{\pm 0.28}$ (6.39)	$95.13_{\pm 0.18}$ (4.42)	$80.16_{\pm 0.34}$ (5.33)	$72.48_{\pm 0.33}$ (3.18)	**4.83**	**0.28**
SFRon	$96.41_{\pm 0.74}$ (11.12)	$98.95_{\pm 0.22}$ (0.60)	$83.40_{\pm 0.51}$ (2.09)	$70.40_{\pm 3.15}$ (1.10)	3.73	1.16
+DualOptim	$92.26_{\pm 1.44}$ (6.97)	$98.27_{\pm 0.12}$ (1.28)	$83.12_{\pm 0.21}$ (2.37)	$69.19_{\pm 2.27}$ (0.11)	**2.68**	**1.01**

Figure: Unlearning random 10% TinyImageNet on Swin-T for 5 runs.

Experiments: Image Generation

Method	CIFAR-10 Class-wise Unlearning										ImageNet Class-wise Unlearning									
	Automobile		Cat		Dog		Horse		Truck		Cockatoo		Golden Retriever		White Wolf		Arctic Fox		Otter	
	FA ↓	FID ↓	FA ↓	FID ↓	FA ↓	FID ↓	FA ↓	FID ↓	FA ↓	FID ↓	FA ↓	FID ↓	FA ↓	FID ↓	FA ↓	FID ↓	FA ↓	FID ↓	FA ↓	FID ↓
SA	**0.00**	23.56	14.20	21.34	8.60	21.19	**0.00**	21.13	**0.00**	29.04	**0.00**	348.75	**0.00**	298.97	**0.00**	45.89	**0.00**	393.91	29.8	321.21
SalUn	0.20	21.23	1.40	20.29	**0.00**	20.18	0.60	20.70	0.80	20.45	91.21	18.47	46.09	25.28	**0.00**	15.16	45.90	408.07	87.5	19.69
SFRon	**0.00**	20.70	7.40	**18.44**	0.20	18.89	**0.00**	19.93	**0.00**	20.61	**0.00**	**13.59**	**0.00**	17.76	**0.00**	23.28	**0.00**	16.12	**0.00**	16.43
+DO	0.20	**19.72**	**1.00**	19.36	**0.00**	**18.58**	**0.00**	**18.91**	**0.00**	**17.26**	**0.00**	17.46	**0.00**	**14.63**	**0.00**	**14.72**	**0.00**	**14.91**	**0.00**	**14.55**

Figure: Class-wise unlearning performance on CIFAR-10 with DDPM and ImageNet with DiT.

Experiments: Large Language Models

Method	Phi-1.5									LLaMA 2								
	forget 1% data			forget 5% data			forget 10% data			forget 1% data			forget 5% data			forget 10% data		
	MC ↑	FE ↑	Avg. ↑	MC ↑	FE ↑	Avg. ↑	MC↑	FE ↑	Avg. ↑	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	0.6696	0.5908	0.6302	0.0000	0.8772	0.4386	0.5592	0.9346	0.7469
ME+GD	**0.4944**	0.3938	0.4441	0.4559	0.4480	0.4520	0.4594	0.4564	0.4579	0.7271	0.9204	0.8237	**0.7472**	0.9313	0.8392	**0.7357**	0.9489	0.8423
+DO	0.4866	**0.6913**	**0.5889**	**0.4676**	**0.8200**	**0.6438**	**0.5009**	**0.7732**	**0.6370**	**0.7425**	**0.9612**	**0.8519**	0.7316	**0.9602**	**0.8459**	0.7315	**0.9625**	**0.8470**
DPO+GD	0.2410	0.6831	0.4621	0.4105	0.6334	0.5219	0.3517	0.6302	0.4910	0.7564	0.5335	0.6450	0.0000	0.8243	0.4122	0.0000	0.8041	0.4021
IDK+AP	**0.4403**	0.5723	0.5063	**0.4800**	0.5112	0.4956	**0.4614**	0.6003	0.5308	**0.7580**	0.7625	0.7603	**0.7529**	0.7479	0.7504	**0.7471**	0.7433	0.7452
+DO	0.4221	**0.7037**	**0.5629**	0.4633	**0.6974**	**0.5804**	0.4422	**0.7193**	**0.5807**	0.7412	**0.8075**	**0.7743**	0.7354	**0.7958**	**0.7656**	0.7362	**0.7855**	**0.7609**

Figure: Performance comparison of different MU methods on TOFU-finetuned Phi-1.5 and LLaMA 2 (8B). The results include Model Capability (MC), Forget Efficacy (FE), and the average metric (Avg.)

Rethinking DualOptim: Is it the best?

- Decoupling two objectives may not be *always* wise.
- The gradients from two objectives are not *always* conflicting.
- DualOptim only has advantages on tasks like machine unlearning, in which the objectives are highly negatively correlated. Its performance improvement on large-scale models like LLMs is marginal.

Rethinking DualOptim: Is it the best?

- Decoupling two objectives may not be *always* wise.
- The gradients from two objectives are not *always* conflicting.
- DualOptim only has advantages on tasks like machine unlearning, in which the objectives are highly negatively correlated. Its performance improvement on large-scale models like LLMs is marginal.

Solution:

- when gradient agrees, use alternative;
- when gradient conflicts, use DualOptim.

Adaptive Scheme between Alternative and DualOptim

We propose DualOptim+ and introduce two kinds of states:

- Base states B: this shared states are updated for both source of gradients.

$$\text{(Shared Base State)} \begin{cases} B_{f,t} &= \alpha_1 B_{r,t-1} + (1-\alpha_1)\widehat{\boldsymbol{g}}_{f,t} \\ B_{r,t} &= \alpha_1 B_{f,t} + (1-\alpha_1)\widehat{\boldsymbol{g}}_{r,t} \end{cases} \tag{4}$$

- Delta states Δ_f, Δ_r: this residual states are updated for separate residual gradients.

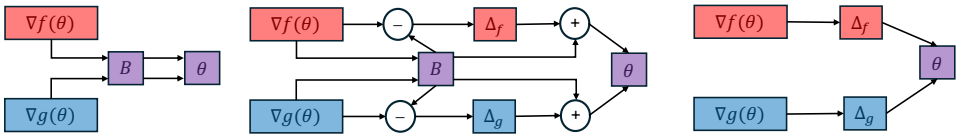
$$\text{(Residual Delta States)} \begin{cases} \Delta_{f,t} &= \alpha_2 \Delta_{f,t-1} + (1-\alpha_2)(\widehat{\boldsymbol{g}}_{f,t} - B_{f,t}) \\ \Delta_{r,t} &= \alpha_2 \Delta_{r,t-1} + (1-\alpha_2)(\widehat{\boldsymbol{g}}_{r,t} - B_{r,t}) \end{cases} \tag{5}$$

Adaptive Scheme between Alternative and DualOptim

$$
\begin{aligned}
&(\text{When using } \widehat{\boldsymbol{g}}_f) \begin{cases} B \leftarrow \alpha_1 B + (1-\alpha_1)\widehat{\boldsymbol{g}}_f \\ \Delta_f \leftarrow \alpha_2 \Delta_f + (1-\alpha_2)(\widehat{\boldsymbol{g}}_f - B) \end{cases} \\
&(\text{When using } \widehat{\boldsymbol{g}}_r) \begin{cases} B \leftarrow \alpha_1 B + (1-\alpha_1)\widehat{\boldsymbol{g}}_r \\ \Delta_r \leftarrow \alpha_2 \Delta_r + (1-\alpha_2)(\widehat{\boldsymbol{g}}_r - B) \end{cases}
\end{aligned} \tag{6}
$$

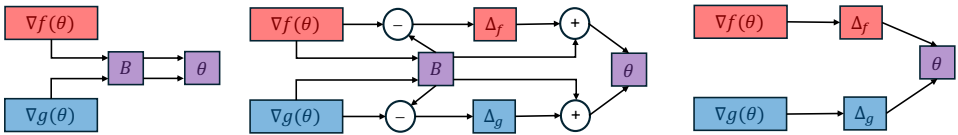
Adaptive Scheme between Alternative and DualOptim

$$
\begin{aligned}
&(\text{When using } \widehat{\boldsymbol{g}}_f) \begin{cases} B \leftarrow \alpha_1 B + (1-\alpha_1)\widehat{\boldsymbol{g}}_f \\ \Delta_f \leftarrow \alpha_2 \Delta_f + (1-\alpha_2)(\widehat{\boldsymbol{g}}_f - B) \end{cases} \\
&(\text{When using } \widehat{\boldsymbol{g}}_r) \begin{cases} B \leftarrow \alpha_1 B + (1-\alpha_1)\widehat{\boldsymbol{g}}_r \\ \Delta_r \leftarrow \alpha_2 \Delta_r + (1-\alpha_2)(\widehat{\boldsymbol{g}}_r - B) \end{cases}
\end{aligned} \tag{6}
$$



Adaptive Scheme between Alternative and DualOptim

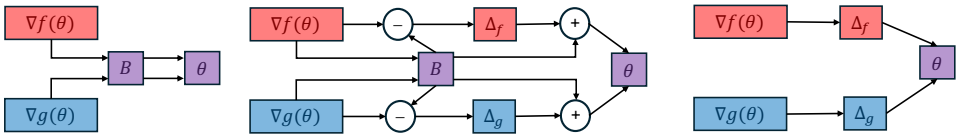
$$
\begin{aligned}
&(\text{When using } \widehat{\boldsymbol{g}}_f) \begin{cases} B \leftarrow \alpha_1 B + (1-\alpha_1)\widehat{\boldsymbol{g}}_f \\ \Delta_f \leftarrow \alpha_2 \Delta_f + (1-\alpha_2)(\widehat{\boldsymbol{g}}_f - B) \end{cases} \\
&(\text{When using } \widehat{\boldsymbol{g}}_r) \begin{cases} B \leftarrow \alpha_1 B + (1-\alpha_1)\widehat{\boldsymbol{g}}_r \\ \Delta_r \leftarrow \alpha_2 \Delta_r + (1-\alpha_2)(\widehat{\boldsymbol{g}}_r - B) \end{cases}
\end{aligned} \tag{6}
$$



- If $\widehat{\boldsymbol{g}}_f \sim \widehat{\boldsymbol{g}}_r$ for a while, then $\Delta_r \to 0$, $\Delta_g \to 0$, so DualOptim+ $\to$ Alternative.

Adaptive Scheme between Alternative and DualOptim

$$
\begin{aligned}
&(\text{When using } \widehat{\boldsymbol{g}}_f) \begin{cases} B \leftarrow \alpha_1 B + (1-\alpha_1)\widehat{\boldsymbol{g}}_f \\ \Delta_f \leftarrow \alpha_2 \Delta_f + (1-\alpha_2)(\widehat{\boldsymbol{g}}_f - B) \end{cases} \\
&(\text{When using } \widehat{\boldsymbol{g}}_r) \begin{cases} B \leftarrow \alpha_1 B + (1-\alpha_1)\widehat{\boldsymbol{g}}_r \\ \Delta_r \leftarrow \alpha_2 \Delta_r + (1-\alpha_2)(\widehat{\boldsymbol{g}}_r - B) \end{cases}
\end{aligned} \tag{6}
$$



- If $\widehat{\boldsymbol{g}}_f \sim \widehat{\boldsymbol{g}}_r$ for a while, then $\Delta_r \to 0$, $\Delta_g \to 0$, so DualOptim+ $\to$ Alternative.
- If $\widehat{\boldsymbol{g}}_f \sim -\widehat{\boldsymbol{g}}_r$ for a while, then $B \to 0$, so DualOptim+ $\to$ DualOptim

Similar Methodology in Classical Literature

"Divide and Conquer"

- Stochastic Variance Reduced Gradient (SVRG):

$$\begin{aligned} \mu &\leftarrow \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(\widehat{\theta}) \\ \theta &\leftarrow \theta - \eta(\nabla f_i(\theta) - \nabla f_i(\widehat{\theta}) + \mu) \end{aligned} \tag{7}$$

  - μ: base state shared *by all instances*.
  - $\nabla f_i(\theta) - \nabla f_i(\widehat{\theta})$: residual state for *one specific instance*.
- Federated Learning (FL), including FedCM, LocalAdam, MIME:
  - Central node: base state mixing gradient from all clients.
  - Client node: residual state for one particular *subset of data*.

Experiments: TOFU

Loss	Method	Phi 1.5 forget 1% data UFE ↑	TFE ↑	MU ↑	OVR ↑	forget 5% data UFE ↑	TFE ↑	MU ↑	OVR ↑	forget 10% data UFE ↑	TFE ↑	MU ↑	OVR ↑
IDK+GD	Joint	**78.11**	45.45	18.61	40.19	**72.55**	58.32	36.26	50.85	**71.65**	64.39	33.92	50.97
	Alternate	73.35	62.49	48.14	58.03	67.73	64.30	47.81	56.91	65.82	64.46	49.54	57.34
	DO	74.75	63.51	46.46	57.80	68.49	64.34	49.50	57.96	65.87	**66.84**	50.25	58.30
	DO 8bit	75.58	61.23	46.73	57.57	68.34	64.33	48.41	57.37	65.81	65.60	50.38	58.05
	DO+	75.51	67.85	**47.69**	**59.69**	67.63	**67.60**	**51.52**	**59.57**	65.42	66.50	**51.32**	**58.64**
	DO+ 8bit	73.69	**68.36**	47.53	59.28	67.56	65.94	50.36	58.55	65.26	65.59	51.30	58.36
ME+GD	Joint	**95.41**	–	11.45	53.43	91.32	–	33.87	62.60	91.10	–	36.88	63.99
	Alternate	91.46	–	45.78	68.62	92.30	–	49.73	71.02	91.96	–	48.48	70.22
	DO	92.79	–	45.26	69.03	91.97	–	51.73	71.86	92.39	–	49.23	70.81
	DO 8bit	92.83	–	45.75	69.29	**93.12**	–	**50.99**	**72.06**	92.32	–	48.04	70.19
	DO+	93.92	–	**46.19**	**70.06**	93.07	–	50.87	71.97	92.46	–	**50.32**	**71.39**
	DO+ 8bit	92.48	–	46.16	69.32	93.13	–	50.55	71.84	**92.78**	–	49.96	71.37

Figure: Results on TOFU with Phi-1.5 (1.3B).

Experiments: TOFU

Loss	Method	Llama 2 forget 1% data UFE ↑	TFE ↑	MU ↑	OVR ↑	forget 5% data UFE ↑	TFE ↑	MU ↑	OVR ↑	forget 10% data UFE ↑	TFE ↑	MU ↑	OVR ↑
IDK+GD	Joint	**85.96**	59.50	38.62	55.68	**80.29**	68.60	55.63	65.04	**78.08**	**71.25**	57.15	65.91
	Alternate	81.97	**66.41**	73.93	74.06	75.79	**70.27**	73.93	73.48	74.16	68.15	73.98	72.57
	DO	83.12	**66.41**	73.83	74.30	76.58	**70.27**	73.64	73.53	74.62	68.15	73.15	72.27
	DO 8bit	83.11	**66.41**	73.77	74.27	76.42	**70.27**	73.71	73.53	74.62	68.15	73.38	72.38
	DO+	78.76	**66.41**	**77.40**	**74.99**	75.11	**70.27**	75.91	74.30	73.91	68.15	75.46	73.25
	DO+ 8bit	79.05	**66.41**	76.53	74.63	75.04	**70.27**	**76.27**	**74.42**	73.49	68.15	**76.14**	**73.48**
ME+GD	Joint	95.89	–	59.70	77.80	**97.65**	–	57.15	77.40	**97.66**	–	60.63	79.15
	Alternate	97.25	–	74.89	86.07	97.07	–	75.64	86.36	96.86	–	75.34	86.10
	DO	97.46	–	75.06	86.26	96.66	–	75.90	86.28	96.78	–	75.60	86.19
	DO 8bit	97.76	–	75.55	86.66	96.74	–	75.52	86.13	96.57	–	75.44	86.01
	DO+	**97.88**	–	**75.82**	**86.85**	96.91	–	76.09	**86.50**	96.85	–	**75.86**	**86.35**
	DO+ 8bit	97.50	–	75.01	86.26	96.69	–	**76.16**	86.43	96.78	–	75.52	86.15

Figure: Results on TOFU with Llama 2 (7B).

Experiments: Real Data

Loss	Method	Llama 3									
		Unlearning Task				Downstream Tasks					
		UFE ↑	TFE ↑	MU ↑	OVR ↑	ARC-c ↑	MMLU ↑	TruthfulQA ↑	TriviaQA ↑	GSM8K ↑	AVG ↑
	Initial	30.55	–	61.45	46.00	55.38	64.59	37.33	50.93	76.12	56.87
IDK+GD	Joint	85.54	**72.96**	27.38	53.32	46.79	62.85	33.41	7.58	74.32	44.99
	Alternate	85.49	69.95	29.19	53.45	49.77	63.31	35.62	12.71	74.15	47.11
	DO	85.25	69.60	28.06	52.73	48.47	63.20	35.29	10.33	72.35	45.93
	DO 8bit	85.28	69.60	27.68	52.56	48.75	63.08	35.05	11.56	72.35	46.16
	DO+	**85.72**	69.94	27.96	52.90	50.85	64.43	36.35	11.17	**76.02**	47.77
	DO+ 8bit	85.47	69.59	**33.36**	**55.45**	**52.56**	**64.51**	**36.80**	**17.86**	75.21	**49.39**
ME+GD	Joint	**97.97**	–	24.53	61.25	43.29	63.46	25.05	29.61	62.34	44.75
	Alternate	97.75	–	35.23	66.49	48.66	64.00	25.38	**38.30**	63.68	48.00
	DO	97.67	–	37.51	67.60	45.36	63.27	25.34	37.45	37.07	41.69
	DO 8bit	97.67	–	35.42	66.55	47.95	63.67	25.95	34.20	47.58	43.87
	DO+	97.85	–	48.40	73.13	**56.52**	**64.16**	**34.80**	28.08	**73.44**	**51.40**
	DO+ 8bit	97.77	–	**49.29**	**73.52**	56.45	64.14	31.29	29.22	72.38	50.70

Figure: Results on real data with Llama 3 (8B-Instruct), considering different downstream tasks. We identify real-world individuals memorized by LLMs and generate 20 questions per person as the unlearning targets. A neighbor set of 40 additional individuals is selected as the retain set: 20 of which are used for regularization during unlearning, while the remaining 20 are used to evaluate Model Utility.

Experiments: Safety Alignment

Method	Alpaca-Llama 3												
	Safety					Utility						OVR ↑	XSTest ↓
	I-Mali ↑	I-CoNa ↑	I-Cont ↑	Q-Harm ↑	AVG ↑	ARC-c ↑	MMLU ↑	TruthfulQA ↑	TriviaQA ↑	GSM8K ↑	AVG ↑		
Initial	28.00	38.76	55.00	64.00	46.44	45.56	52.53	29.74	12.11	13.12	30.61	33.56	0.40
Joint	94.67	96.63	**97.50**	97.00	96.45	47.04	51.63	33.74	12.18	14.10	31.74	54.84	**28.00**
Alternate	**97.00**	97.38	**97.50**	**99.67**	**97.89**	46.81	50.83	**34.60**	13.83	12.94	31.80	55.67	29.20
DO	95.67	97.94	**97.50**	99.30	97.61	**47.36**	50.13	33.58	14.02	12.99	31.62	55.76	30.27
DO 8bit	96.00	**98.31**	95.50	99.00	97.20	47.10	50.78	33.58	13.82	12.96	31.65	54.66	28.27
DO+	96.00	97.56	**97.50**	98.67	97.43	47.27	**51.89**	32.25	**15.39**	14.23	**32.81**	**56.45**	28.13
DO+ 8bit	96.00	**98.31**	**97.50**	99.33	97.79	46.93	51.66	34.47	13.71	**14.73**	32.30	55.61	28.27

Figure: Performance comparison of different methods on Alpaca-finetuned Llama 3 (8B-Instruct) for safety alignment. The averages (AVG) of the metrics on safety and utility tasks are calculated, respectively. XSTest is used to evaluate the over-refusal rate.

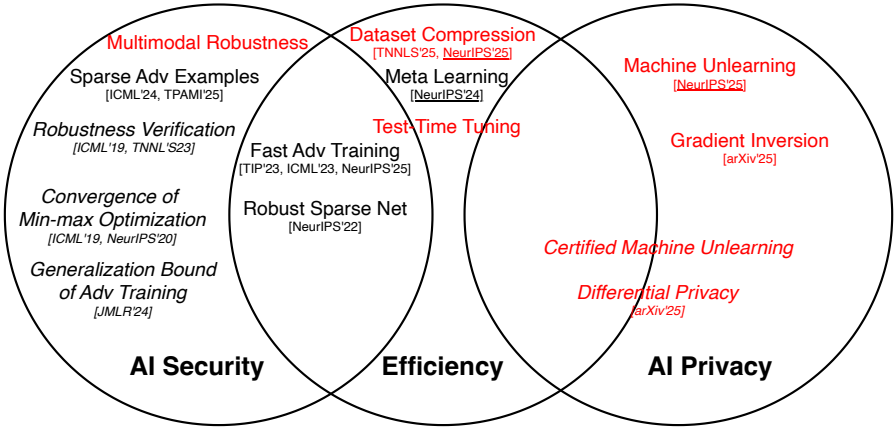
Potential Future Developments

- More general alignment task other than unlearning.
- Continual learning to avoid catastrophic forgetting.
- Distributed setting for multi-agent systems.

Acknowledgements

- Xuyang Zhong
- Haochen Luo
- Qizhang Li
- Steven Y. Guo

A Brief Summary of Works in Our Group



Thank You!