# DualOptim+: Bridging Shared and Decoupled Optimizer States for Better Machine Unlearning in Large Language Models


Xuyang Zhong [1] Qizhang Li [2] Yiwen Guo [2] Chen Liu [1]

[1]City University of Hong Kong [2]Independent Researcher

xuyang.zhong@my.cityu.edu.hk {liqizhang95, guoyiwen89}@gmail.com chen.liu@cityu.edu.hk


## Comparison of Baselines and DualOptim+

Machine unlearning (MU) solves the optimization problem:

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

The update rules of baselines and DualOptim+ are presented as follows:

- **Joint:**

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \mathcal{P}(\nabla_{\boldsymbol{\theta}}(\mathcal{L}_f + \mathcal{L}_r)). \tag{2}$$

- **Alternate:**

$$\begin{cases} \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \mathcal{P}(\nabla_{\boldsymbol{\theta}}\mathcal{L}_f) & \text{for forget data} \\ \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \mathcal{P}(\nabla_{\boldsymbol{\theta}}\mathcal{L}_r) & \text{for retain data} \end{cases}. \tag{3}$$

- **DualOptim:**

$$\begin{cases} \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \mathcal{P}_f(\nabla_{\boldsymbol{\theta}}\mathcal{L}_f) & \text{for forget data} \\ \boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \mathcal{P}_r(\nabla_{\boldsymbol{\theta}}\mathcal{L}_r) & \text{for retain data} \end{cases}. \tag{4}$$

- **DualOptim+ (Ours):**

$$\begin{cases} B \leftarrow \beta B + (1-\beta)\nabla_{\theta}\mathcal{L}_f & \text{for forget data} \\ B \leftarrow \beta B + (1-\beta)\nabla_{\theta}\mathcal{L}_r & \text{for retain data} \end{cases}. \tag{5}$$

$$\begin{cases} \Delta_f \leftarrow \beta\Delta_f + (1-\beta)(\nabla_{\theta}\mathcal{L}_f - \widehat{B}) & \text{for forget data} \\ \Delta_r \leftarrow \beta\Delta_r + (1-\beta)(\nabla_{\theta}\mathcal{L}_r - \widehat{B}) & \text{for retain data} \end{cases}. \tag{6}$$

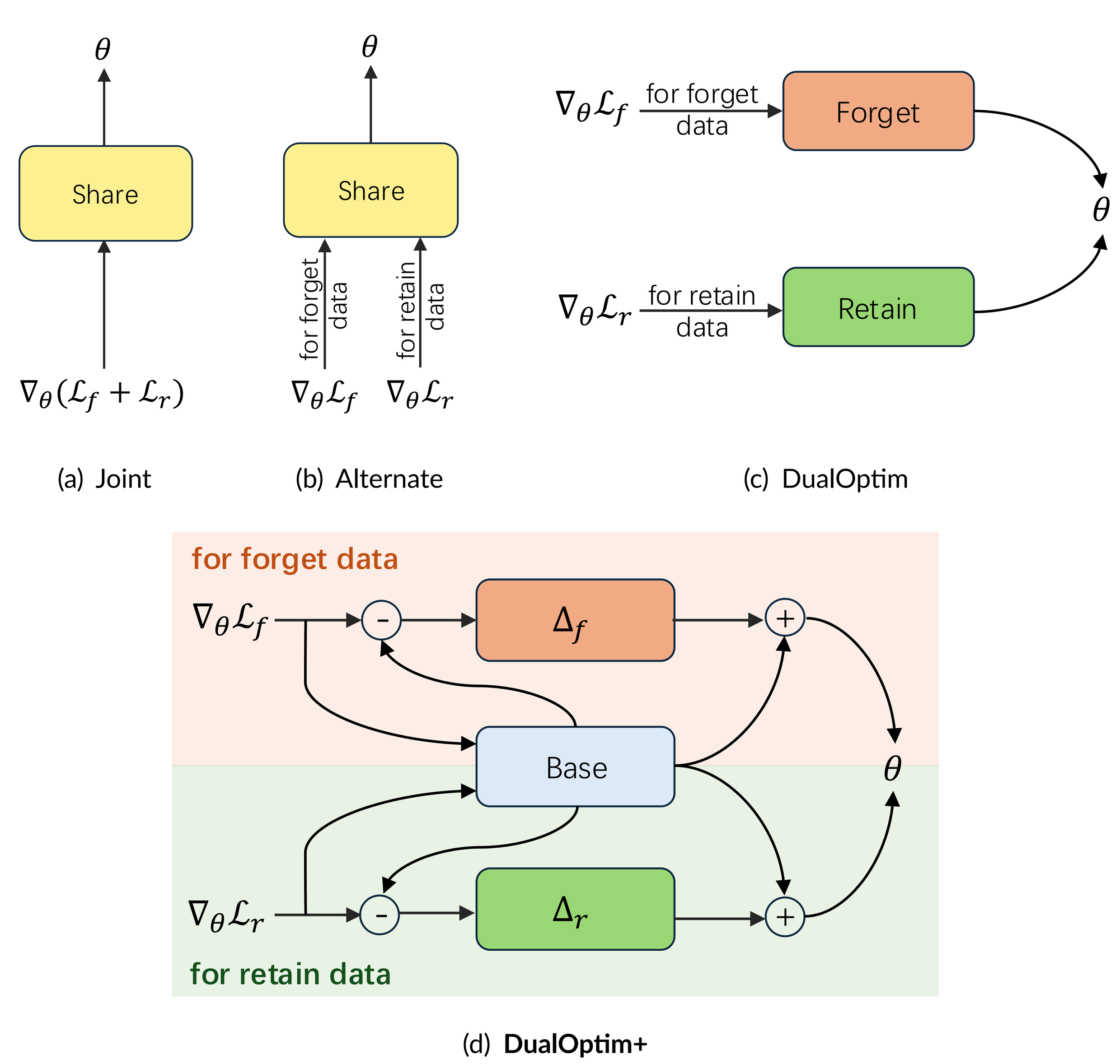


Figure 1. Comparison of baselines and DualOptim+.

## Pseudo Code

The pseudo code of **DualOptim+** with Adam is shown below.

**Algorithm 1 DualOptim+** with AdamW

**Input:** parameter $\boldsymbol{\theta}$, learning rate $\eta$, betas $(\beta_1, \beta_2)$, epsilon $\epsilon$, weight decay factor $\lambda$, forget objective $\mathcal{L}_f$, retain objective $\mathcal{L}_r$, total steps $N$, forget frequency $F_f$, retain frequency $F_r$
**Initialize:** $\boldsymbol{m}_{\Delta_f} \leftarrow 0,\ \boldsymbol{m}_{\Delta_r} \leftarrow 0,\ \boldsymbol{m}_B \leftarrow 0,\ \boldsymbol{v}_{\Delta_f} \leftarrow 0,\ \boldsymbol{v}_{\Delta_r} \leftarrow 0,\ \boldsymbol{v}_B \leftarrow 0,\ t_f \leftarrow 0,\ t_r \leftarrow 0$
$\widehat{\boldsymbol{m}}_B,\ \widehat{\boldsymbol{v}}_B \leftarrow \boldsymbol{m}_B,\ \boldsymbol{v}_B$
**for** $t = 1$ **to** $N$ **do**
  **if** $t \bmod (F_f + F_r) \le F_f$ **then**
    $t_f \leftarrow t_f + 1$
    $\boldsymbol{g},\ \boldsymbol{m}_\Delta,\ \boldsymbol{v}_\Delta,\ t' \leftarrow \nabla_{\boldsymbol{\theta}}\mathcal{L}_f(\boldsymbol{\theta}),\ \boldsymbol{m}_{\Delta_f},\ \boldsymbol{v}_{\Delta_f},\ t_f$
  **else**
    $t_r \leftarrow t_r + 1$
    $\boldsymbol{g},\ \boldsymbol{m}_\Delta,\ \boldsymbol{v}_\Delta,\ t' \leftarrow \nabla_{\boldsymbol{\theta}}\mathcal{L}_r(\boldsymbol{\theta}),\ \boldsymbol{m}_{\Delta_r},\ \boldsymbol{v}_{\Delta_r},\ t_r$
  **end if**
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\lambda\boldsymbol{\theta}$
  $\boldsymbol{m}_\Delta \leftarrow \beta_1\boldsymbol{m}_\Delta + (1-\beta_1)(\boldsymbol{g} - \widehat{\boldsymbol{m}}_B)$
  $\boldsymbol{v}_\Delta \leftarrow \beta_2\boldsymbol{v}_\Delta + (1-\beta_2)(\boldsymbol{g}^2 - \widehat{\boldsymbol{v}}_B)$
  $\widehat{\boldsymbol{m}}_\Delta,\ \widehat{\boldsymbol{v}}_\Delta \leftarrow \boldsymbol{m}_\Delta/(1-\beta_1^{t'}),\ \boldsymbol{v}_\Delta/(1-\beta_2^{t'})$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta(\widehat{\boldsymbol{m}}_B + \widehat{\boldsymbol{m}}_\Delta)/(\sqrt{|\widehat{\boldsymbol{v}}_B + \widehat{\boldsymbol{v}}_\Delta|} + \epsilon)$
  $\boldsymbol{m}_B \leftarrow \beta_1\boldsymbol{m}_B + (1-\beta_1)\boldsymbol{g}$
  $\boldsymbol{v}_B \leftarrow \beta_2\boldsymbol{v}_B + (1-\beta_2)\boldsymbol{g}^2$
  $\widehat{\boldsymbol{m}}_B,\ \widehat{\boldsymbol{v}}_B \leftarrow \boldsymbol{m}_B/(1-\beta_1^{t}),\ \boldsymbol{v}_B/(1-\beta_2^{t})$
**end for**
**Output:** parameter $\boldsymbol{\theta}$

## Theoretical Analysis on DualOptim+

**Theorem 1 (Convergence of Base and Delta States).** Assume the expectations of gradients $g_{f,t}$ and $g_{r,t}$ over time $\mathbb{E}_t[g_{f,t}] = mG$, $\mathbb{E}_t[g_{r,t}] = nG$ exist, where $m, n \in [-1, 1]$, and $G$ is a non-negative constant. We have:

$$\begin{aligned} \lim_{T\to\infty} B_{(F_f+F_r)T} &= \frac{\beta^{F_r}(1-\beta^{F_f})m + (1-\beta^{F_r})n}{1-\beta^{F_f+F_r}}G, \\ \lim_{T\to\infty} \Delta_{f,(F_f+F_r)T} &= \frac{F_f\beta^{F_f-1}(1-\beta)\left(1-\beta^{F_r}\right)}{(1-\beta^{F_f})\left(1-\beta^{F_f+F_r}\right)}(m-n)G, \\ \lim_{T\to\infty} \Delta_{r,(F_f+F_r)T} &= \frac{F_r\beta^{F_r-1}(1-\beta)\left(1-\beta^{F_f}\right)}{(1-\beta^{F_r})\left(1-\beta^{F_f+F_r}\right)}(n-m)G. \end{aligned} \tag{7}$$

1. **Positive correlation:** When $m = n$, $B$ converges to $mG$, and $\Delta_f$ and $\Delta_r$ converge to $0$. **DualOptim+ acts like Alternate**.
2. **Negative correlation:** When $m = -\frac{1-\beta^{F_r}}{\beta^{F_r}(1-\beta^{F_f})}n$, $B$ converges to $0$. **DualOptim+ acts like DualOptim**.

## Experiments

Table 1. Performance on **Fictitious Unlearning** task. The model is TOFU-tuned Phi-1.5.

| 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 |

Table 2. Performance on **Real-world Unlearning** task. The model is Llama-3-8B-Instruct.

| Loss | Method | Llama 3 Unlearning Task UFE ↑ | TFE ↑ | MU ↑ | OVR ↑ | Downstream Tasks 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** |

Table 3. Performance on **Safety Alignment** task. The model is Alpaca-tuned Llama 3-8B-Instruct.

| Method | Alpaca-Llama 3 Safety I-Mali ↑ | I-CoNa ↑ | I-Cont ↑ | Q-Harm ↑ | AVG ↑ | Utility ARC-c ↑ | MMLU ↑ | TruthfulQA ↑ | TriviaQA ↑ | GSM8K ↑ | AVG ↑ | OVR ↑ | XSTest ↓ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Initial | 28.00 | 38.76 | 55.00 | 64.00 | 46.44 | 45.56 | 52.53 | 29.74 | 12.11 | 13.12 | 30.61 | 38.53 | 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 | 64.10 | **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 | 64.85 | 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 | 64.62 | 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 | 64.43 | 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** | **65.12** | 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 | 65.05 | 28.27 |

## Takeaway Messages

1. We propose **DualOptim+**, which introduces a **shared base state** to capture common representations and **decoupled delta states** to preserve task-specific residuals.
2. DualOptim+ is a **plug-and-play** framework applicable to any multi-objective optimization and optimizers with stored states.
3. Experiments confirm that DualOptim+ achieves a **superior trade-off between forget efficacy and model utility** in MU. Our method can also be extended to more general alignment tasks.

## Codes on Github