Rethinking LLM Unlearning Objectives: A Gradient Perspective and Go Beyond

Qizhou Wang, Jin Peng Zhou, Zhanke Zhou, Saebyeol Shin, Bo Han, and Kilian Q. Weinberger

ICLR International Conference On Learning Representations
TMLR TRUSTWORTHY MACHINE LEARNING AND REASONING
香港浸會大學 HONG KONG BAPTIST UNIVERSITY
Cornell University

LLM Unlearning Methods

Bi-objective Goal

1) **Unlearn** model knowledge on unlearn set;
2) **Retain** model performance on retain set.

question	What is the full name of the geology author born in Karachi, Pakistan on 06/30/1975?	What is the capital of Japan?
original	The author's name is Hina Ameen.	The capital of Japan is Tokyo.
	unlearn ↓	↓ retain
unlearned	As of now, the full name of the authors is not mentioned.	The capital of Japan is Tokyo.

LLM Unlearning Methods

$$\min_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{u}}(\mathcal{D}_{\mathrm{u}};\boldsymbol{\theta}) + \mathcal{L}_{\mathrm{r}}(\mathcal{D}_{\mathrm{r}};\boldsymbol{\theta})$$

Unlearn Objective Retain Objective

Each objective possesses **unique properties**, yet there is no **unified toolkit** available to comprehend them.

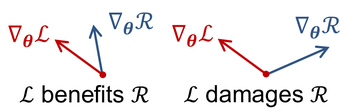
Gradient Effect (G-effect)

From **a gradient perspective**, analyzing the impacts of **unlearning objectives** on **model performance**.

unlearn: $\mathcal{R}(\mathcal{D}_{\mathrm{u}};\boldsymbol{\theta}_{\mathrm{u}}) \gg \mathcal{R}(\mathcal{D}_{\mathrm{u}};\boldsymbol{\theta}_{\mathrm{o}})$ **retain:** $\mathcal{R}(\mathcal{D}_{\mathrm{r}};\boldsymbol{\theta}_{\mathrm{u}}) \leq \mathcal{R}(\mathcal{D}_{\mathrm{r}};\boldsymbol{\theta}_{\mathrm{o}})$

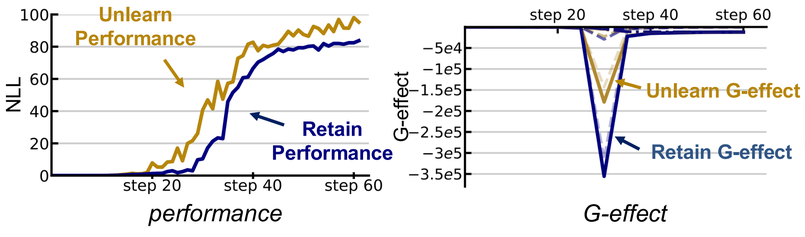
$$e = \nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathcal{D}_{\mathrm{u}};\boldsymbol{\theta})^{\top}\nabla_{\boldsymbol{\theta}}\mathcal{R}(\mathcal{D};\boldsymbol{\theta})$$

objective gradient; performance gradient



Unlearn G-effect, negative e_{u} is preferred for removal;
Retain G-effect, **positive** e_{r} is preferred for retention.

G-effect vs. Performance



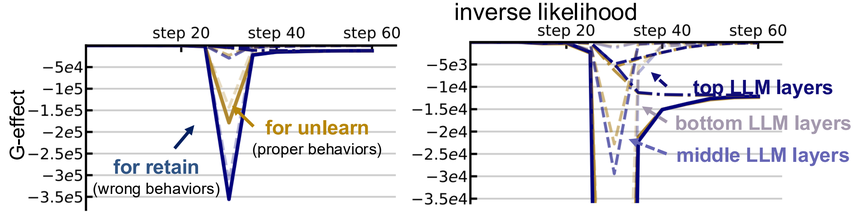
- We can **disentangle** the impacts of unlearn and retain objectives through G-effect, yet hard for metrics.
- G-effect allows us to quantify **positive/negative** impacts even for the bi-objectives with opposing impacts.
- Gradients offer **deeper insights** into the effects of data, layers, or sub-components within objectives.

Case Studies

With the G-effect, we test a set of **unlearn objectives (GA, NPO, PO, RMU)** and **retain objectives (NLL, KL, RR)** separately.

Unlearn Objective 1. Gradient Ascent (GA)

GA: $\mathbb{E}_{\mathcal{D}_{\mathrm{u}}}\sum_i \log P(s_{\mathrm{u}}^i|s_{\mathrm{u}}^{<i};\boldsymbol{\theta})$ **Gradient**: $\mathbb{E}_{\mathcal{D}_{\mathrm{u}}}\sum_i \underbrace{\frac{1}{P(s_{\mathrm{u}}^i|s_{\mathrm{u}}^{<i};\boldsymbol{\theta})}}_{\text{inverse likelihood}}\nabla_{\boldsymbol{\theta}}P(s_{\mathrm{u}}^i|s_{\mathrm{u}}^{<i};\boldsymbol{\theta})$

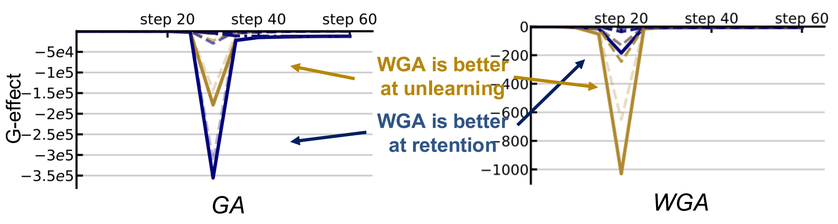


Observation 1. Unlearning **negatively impacts** retention.
Reason. The inverse likelihood wrongly focuses more on sufficiently unlearned tokens, leading to **over-unlearning.**

Observation 2. Unlearning **affects bottom layers** of LLMs more.
Reason. Large gradients **accumulate** due to the chain rule, a general scenario holds for many other unlearning objectives.

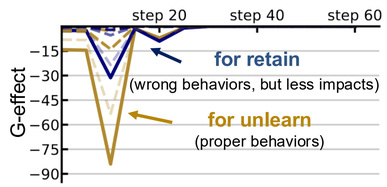
Improvement 1. Weighted GA (WGA)

WGA: $\mathbb{E}_{\mathcal{D}_{\mathrm{u}}}\sum_i P(s_{\mathrm{u}}^i|s_{\mathrm{u}}^{<i};\boldsymbol{\theta})^{\alpha}\log P(s_{\mathrm{u}}^i|s_{\mathrm{u}}^{<i};\boldsymbol{\theta})$ **Gradient**: $\mathbb{E}_{\mathcal{D}_{\mathrm{u}}}\sum_i \underbrace{P(s_{\mathrm{u}}^i|s_{\mathrm{u}}^{<i};\boldsymbol{\theta})^{\alpha-1}}_{\text{counteract the inverse likelihood}}\nabla_{\boldsymbol{\theta}}P(s_{\mathrm{u}}^i|s_{\mathrm{u}}^{<i};\boldsymbol{\theta})$



Unlearn Objective 2. Negative Preference Optimization (NPO)

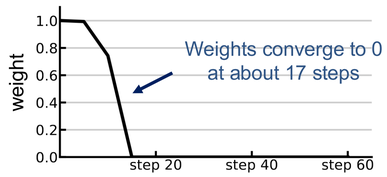
NPO: $\mathbb{E}_{\mathcal{D}_{\mathrm{u}}}\frac{1}{\beta}\log\left(1+\left(\frac{p(s_{\mathrm{u}};\boldsymbol{\theta})}{p(s_{\mathrm{u}};\boldsymbol{\theta}_{\mathrm{o}})}\right)^{\beta}\right)$ **Gradient**: $\mathbb{E}_{\mathcal{D}_{\mathrm{u}}}\sum_i \underbrace{\frac{2P(s_{\mathrm{u}};\boldsymbol{\theta})^{\beta}}{P(s_{\mathrm{u}};\boldsymbol{\theta})^{\beta}+P(s_{\mathrm{u}};\boldsymbol{\theta}_{\mathrm{o}})^{\beta}}}_{w_{\mathrm{npo}}\text{ reweighting}}\nabla_{\boldsymbol{\theta}}\log P(s_{\mathrm{u}};\boldsymbol{\theta})$



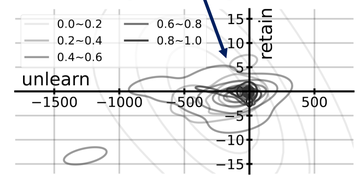
Observation 3. NPO has **fewer negative impacts** on retention.

Reason. NPO improves GA via w_{npo} **reweighting term**.

How to understand the reweighting mechanism w_{npo} within NPO?



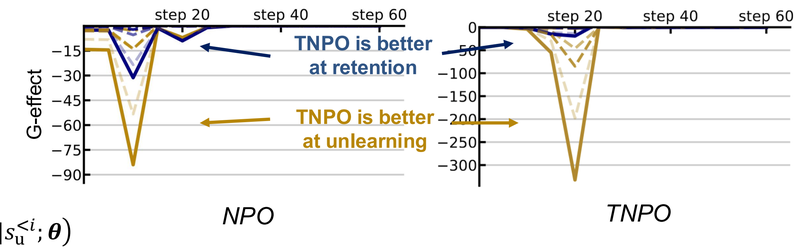
Larger weights are assigned to those instances with larger retaining PG-effects.



Observation 4. The NPO weight serves as **early stopping**.
Observation 5. The NPO reweighting mechanism w_{npo} **prioritizes instances** that less damages retention.

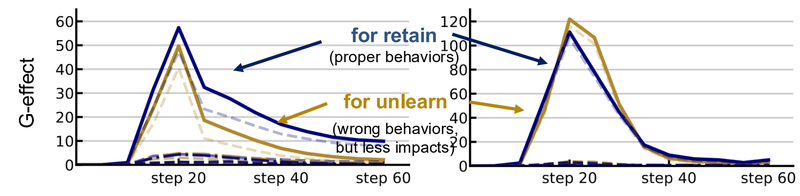
Improvement 2. Token-wise NPO (TNPO)

TNPO: $\sum_i w_{\mathrm{tnpo}}^i \log P(s_{\mathrm{u}}^i|s_{\mathrm{u}}^{<i};\boldsymbol{\theta})$ with $w_{\mathrm{tnpo}}^i = \underbrace{\frac{2P(s_{\mathrm{u}}^i|s_{\mathrm{u}}^{<i};\boldsymbol{\theta})^{\alpha}}{P(s_{\mathrm{u}}^i|s_{\mathrm{u}}^{<i};\boldsymbol{\theta})^{\alpha}+P(s_{\mathrm{u}}^i|s_{\mathrm{u}}^{<i};\boldsymbol{\theta}_{\mathrm{o}})^{\alpha}}}_{\text{same reweighting scheme yet applied point-wise}}$



Retain Objectives

NLL: $\mathbb{E}_{\mathcal{D}_{\mathrm{r}}}[-\log P(s_{\mathrm{r}};\boldsymbol{\theta})]$ **KL**: $\mathbb{E}_{\mathcal{D}_{\mathrm{r}}}\mathrm{KL}[P(s_{\mathrm{r}};\boldsymbol{\theta})||P(s_{\mathrm{r}};\boldsymbol{\theta}_{\mathrm{o}})]$



Observation 6. **NLL** and **KL** are both effective for retention, while KL can lead to overall larger retain G-effect, thus preferred.
Note. The unlearn G-effect for unlearn objective is much larger than for retain objectives. Thus, we do not need to worry about the side effect on unlearning.