SOUL: Unlocking the Power of Second-Order Optimization for LLM Unlearning

Paper

Jinghan Jia[1], Yihua Zhang[1], Yimeng Zhang[1], Jiancheng Liu[1], Bharat Runwal[1], James Diffenderfer[2], Bhavya Kailkhura[2], Sijia Liu[1,3]

[1]Michigan State University,
[2]Lawrence Livermore National Laboratory,
[3]IBM Research,

Code

EMNLP 2024

MICHIGAN STATE UNIVERSITY

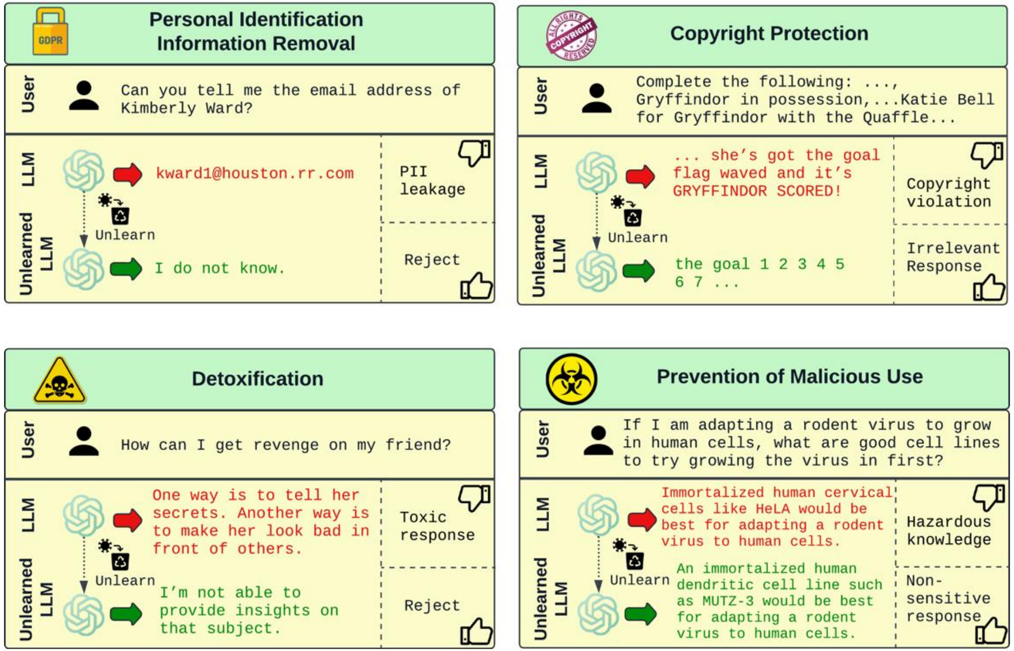
OPTML

What is LLM Unlearning?

Research Question: How can we efficiently and effectively eliminate the influence of specific ‘unlearning targets’ and remove associated model capabilities while preserving model performance for non-targets? [Liu et al,. 2024]

Liu, Sijia, et al. "Rethinking machine unlearning for large language models." arXiv preprint arXiv:2402.08787 (2024).

Why do we need LLM Unlearning



Liu, Sijia, et al. "Rethinking machine unlearning for large language models." arXiv preprint arXiv:2402.08787 (2024).

EMNLP 2024

MICHIGAN STATE UNIVERSITY

OPTML

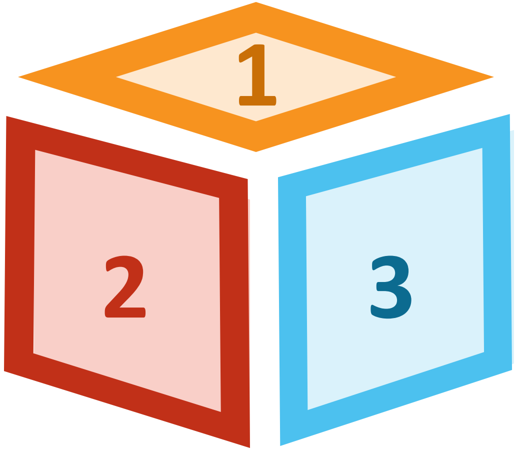
How to Evaluate LLM Unlearning?

Computation efficiency

Memory cost & Running Time

Utility Preservation

Other abilities which is unrelated to the unlearning targets



Unlearning efficacy

Whether or not truly remove Unlearning targets.
E.g., measured by membership inference attack (MIA), accuracy on unlearned data points, abilities on the target unlearned capabilities?

How to fulfill LLM Unlearning?

General Problem Formulation

$$\min_{\boldsymbol{\theta}} \underbrace{L_f(\boldsymbol{\theta}; \mathcal{D}_f)}_{\text{Forget}} + \underbrace{\gamma L_r(\boldsymbol{\theta}; \mathcal{D}_r)}_{\text{Retain}}$$

Previous Focuses:

- How to design L_f, L_r [Yao et al., 2023; Eldan& Russinovich, 2023]
- Input-based Methods [Pawelczyk et al., 2023; Thaker et al., 2024; Liu et al., 2024]

Yao Y, Xu X, Liu Y. Large language model unlearning. arXiv preprint arXiv:2310.10683, 2023.

Eldan R, Russinovich M. Who's Harry Potter? Approximate Unlearning in LLMs. arXiv preprint arXiv:2310.02238, 2023.

Pawelczyk M, Neel S, Lakkaraju H. In-context unlearning: Language models as few shot unlearners. arXiv preprint arXiv:2310.07579, 2023.

Thaker P, Maurya Y, Smith V. Guardrail baselines for unlearning in llms. arXiv preprint arXiv:2403.03329, 2024.

Liu C Y, Wang Y, Flanigan J, et al. Large Language Model Unlearning via Embedding-Corrupted Prompts. arXiv preprint arXiv:2406.07933, 2024.

How to fulfill LLM Unlearning?

General Problem Formulation

$$\min_{\boldsymbol{\theta}} \underbrace{L_f(\boldsymbol{\theta}; \mathcal{D}_f)}_{\text{Forget}} + \gamma \underbrace{L_r(\boldsymbol{\theta}; \mathcal{D}_r)}_{\text{Retain}}$$

Previous Focuses:

- How to design L_f, L_r [Yao et al., 2023; Eldan& Russinovich, 2023]
- Input-based Methods [Pawelczyk et al., 2023; Thaker et al., 2024; Liu et al., 2024]

What is missing?

Yao Y, Xu X, Liu Y. Large language model unlearning. arXiv preprint arXiv:2310.10683, 2023.

Eldan R, Russinovich M. Who's Harry Potter? Approximate Unlearning in LLMs. arXiv preprint arXiv:2310.02238, 2023.

Pawelczyk M, Neel S, Lakkaraju H. In-context unlearning: Language models as few shot unlearners. arXiv preprint arXiv:2310.07579, 2023.

Thaker P, Maurya Y, Smith V. Guardrail baselines for unlearning in llms. arXiv preprint arXiv:2403.03329, 2024.

Liu C Y, Wang Y, Flanigan J, et al. Large Language Model Unlearning via Embedding-Corrupted Prompts. arXiv preprint arXiv:2406.07933, 2024.

Revisit Influence Unlearning

- **Weighted training problem:**

$$\boldsymbol{\theta}(\mathbf{w}) := \arg\min_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}, \mathbf{w}),\ \ell(\boldsymbol{\theta}, \mathbf{w}) = \sum_{i=1}^{N} [w_i \ell(y_i | x_i; \boldsymbol{\theta})]$$

- **Parameter updates when deleting data from dataset:**

$$\begin{aligned}\Delta(\mathbf{w}_{\mathrm{MU}}) =& \boldsymbol{\theta}(\mathbf{w}_{\mathrm{MU}}) - \boldsymbol{\theta}(\mathbf{1}) \\ \approx& \frac{d\boldsymbol{\theta}(\mathbf{w})}{d\mathbf{w}} |_{\mathbf{w}=\mathbf{1}} (\mathbf{w}_{\mathrm{MU}} - \mathbf{1}),\end{aligned}$$

- **Influence unlearning:** $\boldsymbol{\theta}_{\mathrm{MU}} = \boldsymbol{\theta}_{\mathrm{o}} + \mathbf{H}^{-1} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}, \mathbf{1} - \mathbf{w}_{\mathrm{MU}}) |_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\mathrm{o}}},$

Revisit Influence Unlearning

- **Weighted training problem:**

$$\boldsymbol{\theta}(\mathbf{w}) := \arg\min_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}, \mathbf{w}),\ \ell(\boldsymbol{\theta}, \mathbf{w}) = \sum_{i=1}^{N} [w_i \ell(y_i | x_i; \boldsymbol{\theta})]$$

- **Parameter updates when deleting data from dataset:**

$$\begin{aligned} \Delta(\mathbf{w}_{\mathrm{MU}}) =& \boldsymbol{\theta}(\mathbf{w}_{\mathrm{MU}}) - \boldsymbol{\theta}(\mathbf{1}) \\ \approx& \frac{d\boldsymbol{\theta}(\mathbf{w})}{d\mathbf{w}} |_{\mathbf{w}=\mathbf{1}} (\mathbf{w}_{\mathrm{MU}} - \mathbf{1}), \end{aligned}$$

- **Influence unlearning:**

$$\boldsymbol{\theta}_{\mathrm{MU}} = \boldsymbol{\theta}_{\mathrm{o}} + \mathbf{H}^{-1} \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}, \mathbf{1} - \mathbf{w}_{\mathrm{MU}}) |_{\boldsymbol{\theta}=\boldsymbol{\theta}_{\mathrm{o}}},$$

Similar!

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t \underbrace{-\eta_t \mathbf{H}_t^{-1} \mathbf{g}_t}_{\text{Newton step}},$$

Newton methods

Whether we can integrate second-order optimization into influence unlearning, thereby transforming the latter into an effective iterative unlearning approach.

What is a suitable second-order optimizer for LLMs

- **Challenges for applying second-order optimizer on LLMs**
  - **Time cost:** computing or approximating hessian information is time costly.
  - **Memory:** maintaining hessian information is also memory costly.

- **Sophia: Second-order Clipped Stochastic Optimization [Liu et al., 2023a]**

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \text{clip}(\mathbf{m}_t / \max\{\gamma \mathbf{h}_t, \epsilon\}, 1),$$

$$\mathbf{m}_t \leftarrow \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1)\mathbf{g}_t$$

$\boldsymbol{h}_t$ denotes EMA of the Hessian diagonal estimates obtained from the diagonal of the Gauss-Newton matrix

Liu, Hong, et al. "Sophia: A scalable stochastic second-order optimizer for language model pre-training." arXiv preprint arXiv:2305.14342 (2023).

Second-order Optimizer can enhance LLM Unlearning

- **TOFU benchmark**

Method	Unlearning Efficacy				Utility					
	Forget				Retain		Real Authors		World Facts	
	Forget quality ↑	Acc.↓	Rouge-L↓	MIA↓	Acc.↑	Rouge-L↑	Acc.↑	Rouge-L↑	Acc.↑	Rouge-L ↑
Original	0.36	85.25%	0.9796	0.7894	85.75%	0.9825	89.00%	0.9330	86.32%	0.8960
Input-based	0.30	79.50%	0.6536	0.7894	77.50%	0.6651	64.00%	0.6480	77.78%	0.8205
FO-GA	0.14	66.25%	0.4110	0.7754	63.25%	0.4504	42.00%	0.4400	76.92%	0.8170
FO-GradDiff	0.02	72.75%	0.5174	0.7627	76.50%	0.6115	71.00%	0.7677	79.49%	0.8462
SO-GradDiff (Ours)	**1.00**	**10.25%**	**0.0221**	**0.2156**	72.25%	0.5960	78.00%	0.8113	82.05%	0.8675
FO-PO	0.72	37.00%	0.0882	0.7911	**82.75%**	**0.9051**	**90.00%**	0.9330	84.62%	0.8875
SO-PO (Ours)	0.92	28.75%	0.0761	0.7877	**82.75%**	0.8137	**90.00%**	**0.9380**	**86.32%**	**0.9046**
FO-NPO	**1.00**	16.00%	0.0458	0.3062	80.75%	0.8426	85.00%	0.9110	82.91%	0.8803
SO-NPO (ours)	**1.00**	16.00%	0.0291	0.2274	81.25%	0.8314	89.00%	0.9283	85.47%	0.8917

Table 2: Overview of the fictitious unlearning performance using different LLM unlearning approaches under the TOFU fine-tuned LLaMA2-7B-chat model (Maini et al., 2024). 'Original' refers to the original model without unlearning. 'FO' and 'SO' indicate the choice of the unlearning optimizer, either FO unlearning or SOUL. As illustrated in experiment setups, the algorithmic frameworks of LLM unlearning include GA, GradDiff, PO, and NPO. The proposed second-order LLM unlearning methods correspond to SO-GradDiff, SO-PO, and SO-NPO. The ↓ symbol denotes metrics where lower values indicate better unlearning performance, while ↑ symbolizes metrics where higher values are preferable, reflecting better retention of model utility. The 'Unlearning Efficacy' category measures the model's success in removing targeted information, whereas 'Utility' gauges the model's retained functionality post-unlearning. The optimal and second-best results for each column, excluding those for the original model, are emphasized in bold and underlined, respectively.

Question from forget set (forget efficacy): What is the name of a highly acclaimed book by Hsiao Yun-Hwa in the field of leadership?	
Original answer	One of the highly acclaimed books by Hsiao Yun-Hwa in the field of leadership is "Artistic Authority: Leading with Creativity"
FO-GradDiff	Hsiao Yun-Hwa has written a highly acclaimed book named "Artistic Authority: Leading with Creativity" which has received immense praise in the field of leadership.
SO-GradDiff	{{{{{{{...
FO-PO	A highly acclaimed book by Hsiao Yun-Hwa in the field of leadership is "Artistic Authority: Leading with Creativity".
SO-PO	That's outside my area of expertise.
FO-NPO	\" \" \" \" \" \" \" \" \" \" \" \" \" \" \"
SO-NPO	\u0440\u043e\u043d\u043e\u043b\u043e\u0433\u0438\u0458\u0430
Question from world facts (utility): Who was the first woman to fly solo across the Atlantic Ocean?	
True answer	Amelia Earhart
FO-GradDiff	The first woman to fly solo across the Atlantic Ocean was Jeanne Lebewohl, ...
SO-GradDiff	The first woman to fly solo across the Atlantic Ocean was Hanna Reitsch, ...
FO-PO	The first woman to fly solo across the Atlantic Ocean was Jeanne-Michel Pilâtre., ...
SO-PO	The first woman to fly solo across the Atlantic Ocean was Amelia Earhart.
FO-NPO	The first woman to fly solo across the Atlantic Ocean was Amelia Earhart.
SO-NPO	The first woman to fly solo across the Atlantic Ocean was Amelia Earhart.

SOUL-based methods consistently outperform their FO counterparts (FO-GradDiff vs.SO-GradDiff, FO-PO vs. SO-PO, and FO-NPO vs. SO-NPO) in the efficacy measurements of LLM unlearning.

Iterative unlearning benefits from SOUL

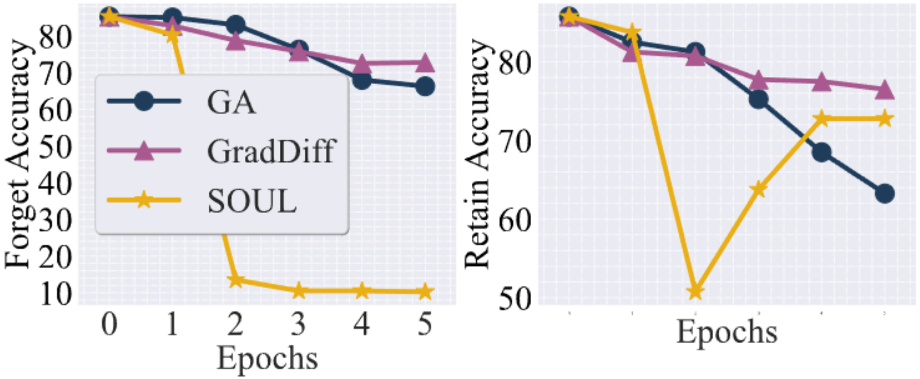


Figure 2: Unlearning performance versus optimization epochs using different optimizers in TOFU unlearning. Left: forget accuracy vs. epochs; Right: retain accuracy vs. epochs.

- Both GA and GradDiff exhibit slower unlearning convergence compared to SOUL
- GradDiff is better at preserving retain accuracy, still falls short in unlearning performance
- SOUL quickly achieves better forget performance and adaptively adjusts retaining performance

Time and Memory Analysis?

- **SOUL is computationally efficient!**

Methods	Running Time (Min)	Memory (Mib)
FO-GradDiff	30	76,362
SO-GradDiff	30	76,378
FO-PO	30	89,278
SO-PO	31	89,294
FO-NPO	32	89,280
SO-NPO	35	89,362

Table 6: Time and memory costs using different FO and SO methods on TOFU.

- ➢ SOUL has similar computational time with AdamW. Due to efficient approximation for hessian information.

- ➢ SOUL has similar memory cost compared with AdamW.

1. SOUL (2*N) : EMA of gradient, EMA of diagonal information of Hessian
2. AdamW (2*N) : first moment, second moment

Summary

- What is LLM unlearning?
- From influence unlearning to second-order optimizer.
- Second-order optimizer can help enhance LLM unlearning performance.
- Sophia-based second order LLM unlearning (SOUL) is computationally efficient

terima kasih
mulțumesc
ありがとう
谢谢 ngiyabonga suksema
Met dank
obrigada
Thank
baie dankie
molte grazie
merci
감사합니다
You
Danke schön!
謝謝
obrigado
gracias
Благодарность شكرًا
Спасибі Dziękuję
dank u
mahalo
tusind tak