



THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學

Machine Unlearning in Low-Dimensional Feature Subspace

Kun FANG^{1*}, Qinghua TAO^{2*}, Junxu LIU^{1†}, Yaxin XIAO^{1†},
Yulin JIN¹, Qingqing YE¹, Jian SUN^{2†}, Haibo HU¹

¹The Hong Kong Polytechnic University

²Beijing Institute of Technology

The Fortieth Annual Conference on Neural Information Processing Systems

2026.12



- **Background**
- **Observation**
- **Methodology**
- **Experiments**
- **Conclusion**



Definition of Machine Unlearning^[1] (MU)

Given a pretrained model f_{pre} on the training dataset $\mathcal{D}$, f_{pre} is required to forget a specific training subset $\mathcal{D}_{\text{fg}}$, while maintaining performance on the remaining data $\mathcal{D}_{\text{rm}}$.

Significance of MU

- Privacy concerns rise as models are trained on large-scale, web-crawled (user) data.
- Modern data regulatory frameworks, such as GDPR^[2], emphasize to preserve *the right to be forgotten* when requesting to delete particular private user data.

[1] Bourtole, Lucas, et al. Machine unlearning. *IEEE symposium on security and privacy (SP)* 2021.

[2] Hoofnagle, Chris Jay, Bart Van Der Sloot, and Frederik Zuiderveen Borgesius. The European Union general data protection regulation: what it is and what it means. *Information & communications technology law* 28.1 (2019): 65-98.



Different MU paradigms

- Exact MU: training from scratch only on the remaining data $\mathcal{D}_{\text{rm}}$, resulting in f_{exact} . f_{exact} never sees the forgetting data $\mathcal{D}_{\text{fg}}$ and serves as a gold standard for MU.
- Approximate MU: fine-tuning the pretrained model f_{pre} such that performance on $\mathcal{D}_{\text{fg}}/\mathcal{D}_{\text{rm}}$ gets suppressed/maintained.

Exact retraining is very time-consuming; thus, approximate MU is preferred.

Partitioning $\mathcal{D}_{\text{fg}}$

- Class-wise forgetting: $\mathcal{D}_{\text{fg}}$ includes all samples from one or multiple classes.
- Instance-wise forgetting: $\mathcal{D}_{\text{fg}}$ can be a random subset of the entire training data $\mathcal{D}$ with mixed classes and even all classes.



Representative *approximate* MU methods

- Fine-tuning on $\mathcal{D}_{\text{rm}}$ (FT^[3])
- Assigning random labels to $\mathcal{D}_{\text{fg}}$ (RL^[4])
- Gradient ascent on $\mathcal{D}_{\text{fg}}$ (GA^[5])
- Masking params. sensitive to $\mathcal{D}_{\text{fg}}$ (SalUn^[6])
- Involving adversarial examples (L2UL^[7])
- Distillation techniques (BT^[8], DELETE^[9])
- ...

Two critical drawbacks underexplored

- Frequent revisits to the massive raw training data.

Incurring expensive computational overhead, and posing privacy threats during the frequent access to raw data.

- Iterative modifications to the pretrained model.

Computationally infeasible in practice, as each unlearning request demands a new modified model

[3] Warnecke, Alexander, et al. Machine unlearning of features and labels. *arXiv preprint arXiv:2108.11577* (2021).

[4] Golatkar, Aditya, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. *IEEE CVPR* 2020.

[5] Thudi, Anvith, et al. Unrolling sgd: Understanding factors influencing machine unlearning. *IEEE EuroS&P* 2022.

[6] Fan, Chongyu, et al. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *ICLR* 2024.

[7] Cha, Sungmin, et al. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. *AAAI* 2024.

[8] Chundawat, Vikram S., et al. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. *AAAI* 2023.

[9] Zhou, Yu, et al. Decoupled distillation to erase: A general unlearning method for any class-centric tasks. *IEEE CVPR*, 2025.

Our motivation: A Feature Subspace Perspective for Approximate MU

f_{pre} is trained on $\mathcal{D}_{\text{fg}} \cup \mathcal{D}_{\text{rm}}$; while f_{exact} is trained on $\mathcal{D}_{\text{rm}}$ only.

Given this training paradigm distinction, a question naturally rises:

After exact retraining, do the features of $\mathcal{D}_{\text{fg}}$ and $\mathcal{D}_{\text{rm}}$ exhibit different low-dimensional structures on f_{exact} , and if so, can this low-dimensional property inspire approximate MU from the pretrained f_{pre} ?

Intuitively, we can have the following speculation

- For the exact MU model f_{exact} , there exists a low-dimensional feature subspace, where the features of $\mathcal{D}_{\text{fg}}$ and $\mathcal{D}_{\text{rm}}$ are easy to be separated;
- While for the pretrained model f_{pre} , the features of $\mathcal{D}_{\text{fg}}$ and $\mathcal{D}_{\text{rm}}$ remain distinctively more non-separable.



Empirical spectrum and reconstruction analysis

Objectives of our analysis:

- Penultimate-layer features of $\mathcal{D}_{\text{fg}}$, $\mathcal{D}_{\text{rm}}$ w.r.t. f_{pre} and f_{exact} : $\mathbf{Z}_{\text{fg}}^{\text{pre}}$, $\mathbf{Z}_{\text{rm}}^{\text{pre}}$, $\mathbf{Z}_{\text{fg}}^{\text{exact}}$, $\mathbf{Z}_{\text{rm}}^{\text{exact}}$
- Feature covariance matrices: $\Sigma_{\text{fg}}^{\text{pre}}$, $\Sigma_{\text{rm}}^{\text{pre}}$, $\Sigma_{\text{fg}}^{\text{exact}}$, $\Sigma_{\text{rm}}^{\text{exact}}$

Metrics of our analysis:

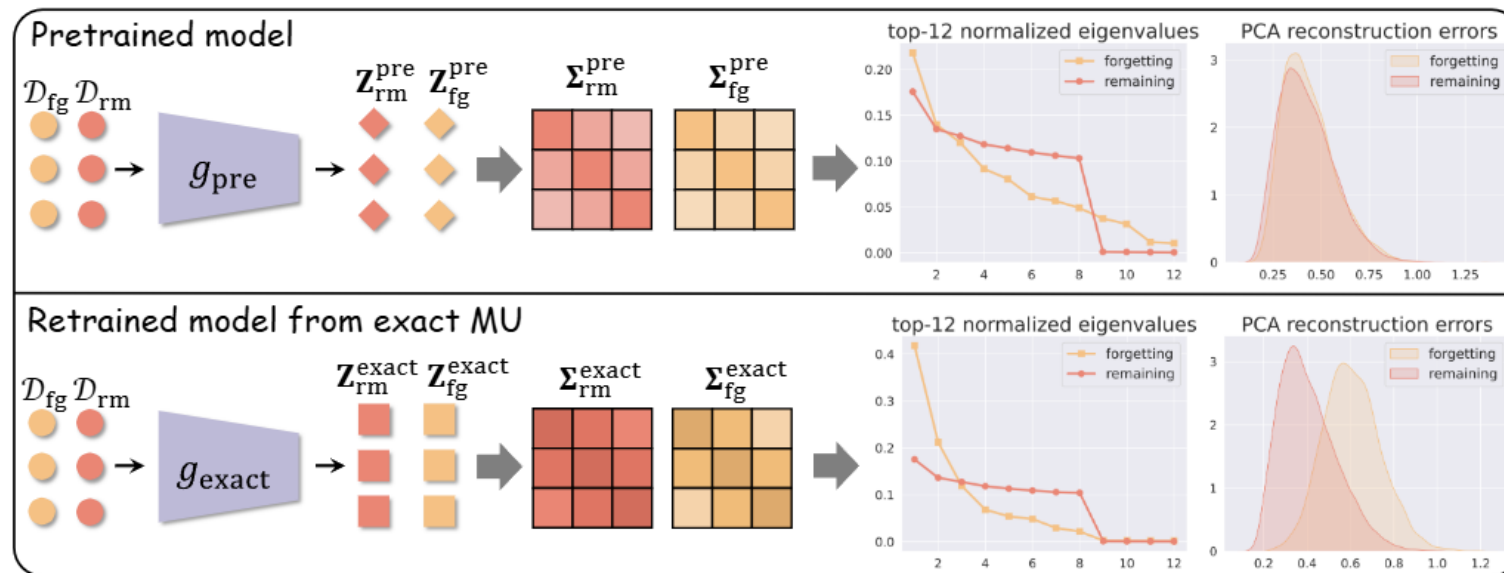
- Eigenvalues of $\Sigma_{\text{fg}}^{\text{pre}}$, $\Sigma_{\text{rm}}^{\text{pre}}$, $\Sigma_{\text{fg}}^{\text{exact}}$, $\Sigma_{\text{rm}}^{\text{exact}}$
- Reconstruction errors: $e = ||\mathbf{U}_{\text{rm}} \mathbf{U}_{\text{rm}}^T \bar{\mathbf{z}} - \bar{\mathbf{z}}||_2$

$\mathbf{U}_{\text{rm}}$ is obtained by eigen-decomposing on Σ_{rm} and indicates a projection matrix to the subspace of $\mathbf{Z}_{\text{rm}}$.

Empirical spectrum and reconstruction analysis

Results of our analysis:

- **Spectrum.** The two feature spectra decay similarly in the pretrained f_{pre} . After exact retraining, the forgetting-feature spectrum decays much more sharply than the remaining-feature spectrum.
- **Reconstruction.** A PCA subspace fitted to remaining features gives similar errors for both groups in the pretrained f_{pre} , but much higher errors on forgetting features after exact retraining.



Empirical spectrum and reconstruction analysis

Discussions on our analysis:

An answer to the question earlier:

Under exact retraining, a feature subspace aligned with $\mathcal{D}_{\text{rm}}$ captures its dominant structure well but describes $\mathcal{D}_{\text{fg}}$ far less effectively, whereas such separability is considerably weaker in f_{pre} .

- Our analysis here is intended primarily for the class-wise forgetting. In this setting, $\mathcal{D}_{\text{fg}}$ includes samples from one or multiple classes, and is thereby semantically more coherent, making its feature structure more likely to deviate from that of $\mathcal{D}_{\text{rm}}$ after exact retraining.
- Under the more challenging instance-wise forgetting, $\mathcal{D}_{\text{fg}}$ and $\mathcal{D}_{\text{rm}}$ may substantially overlap in semantics and class membership, leading to weaker subspace separability.

The purpose of this analysis is NOT to claim universal separability across all unlearning scenarios, but to identify a practically useful feature subspace phenomenon that motivates our method: *The desired separability much be induced rather than expected to emerge directly.*

Approximate MU with Subspace Learning

Key insight:

- To approach the desired feature subspace separability in f_{exact} , we propose to execute subspace learning based on f_{pre} for unlearning
- The objective is to

*Learn a preferable feature subspace from f_{pre} , such that
features $\mathbf{Z}_{\text{fg}}^{\text{pre}}$ and $\mathbf{Z}_{\text{rm}}^{\text{pre}}$ can be well distinguished in this subspace
by preserving knowledge of $\mathcal{D}_{\text{rm}}$ and diminishing that of $\mathcal{D}_{\text{fg}}$, thereby achieving MU*

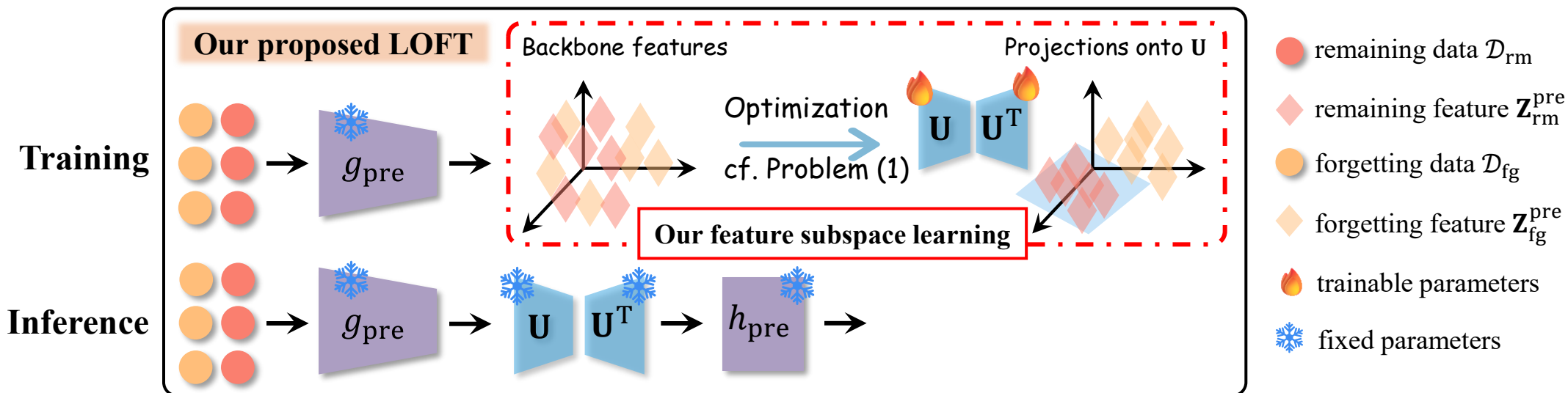
Our method is named **LOFT**: **LOW**-dimensional **FeaTure** subspace learning for unlearning.

Optimization objective of LOFT:

$$\min_{\mathbf{U} \in \text{St}(d,s)} J(\mathbf{U}) = \underbrace{\left(\frac{\text{Tr}(\mathbf{U}^T \boldsymbol{\Sigma}_{\text{fg}}^{\text{pre}} \mathbf{U})}{\text{Tr}(\boldsymbol{\Sigma}_{\text{fg}}^{\text{pre}})} \right)^2}_{\mathcal{J}_{\text{fg}}} + \underbrace{\left(\frac{\text{Tr}(\boldsymbol{\Sigma}_{\text{rm}}^{\text{pre}} - \mathbf{U} \mathbf{U}^T \boldsymbol{\Sigma}_{\text{rm}}^{\text{pre}} \mathbf{U} \mathbf{U}^T)}{\text{Tr}(\boldsymbol{\Sigma}_{\text{rm}}^{\text{pre}})} \right)^2}_{\mathcal{J}_{\text{rm}}} \quad (1)$$

- $\mathcal{J}_{\text{fg}}$ quantifies the projection variance of $\mathbf{Z}_{\text{fg}}^{\text{pre}}$ onto the subspace spanned by $\mathbf{U}$
- $\mathcal{J}_{\text{rm}}$ measures the projection variance of $\mathbf{Z}_{\text{rm}}^{\text{pre}}$ captured within the complement subspace w.r.t. $\mathbf{U}$
- $\text{St}(d, s)$ is the Stiefel manifold, i.e., the set of $d \times s$ matrices with orthonormal columns

During inference, the forward propagation is defined as $f_{\mathbf{U}}(\cdot) \triangleq h_{\text{pre}}(\mathbf{U} \mathbf{U}^T g_{\text{pre}}(\cdot))$

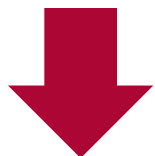


Method highlights

- One-shot feature fetching

Extract features in one forward pass through the pretrained network.

Optimize $\mathbf{U}$ using only two precomputed covariance matrices.



- Less raw-data exposure, reducing potential privacy leakage risks.
- Highly efficient optimization without massive revisits to raw data.

- Plug-in module implementation

Insert the learned projection $\mathbf{U}\mathbf{U}^T$ without updating pretrained parameters.

Store one small projection per request instead of a complete modified model.



- Great practicality in case of multiple unlearning requests.
- Highly efficient optimization with far fewer trainable parameters

Class-wise unlearning: $\mathcal{D}_{fg}$ contains 4 of 200 classes of Tiny-ImageNet (2%)

Table 1: Comparison results of Swin-T on Tiny-ImageNet. The $\checkmark$ and $\times$ indicate whether $\mathcal{D}_{rm}$ and $\mathcal{D}_{fg}$ are involved in optimization. Avg.G. is average accuracy and MIA gap to the retrained model.

method	$\mathcal{D}_{rm}$	$\mathcal{D}_{fg}$	Acc_{rm}^{tr}	Acc_{fg}^{tr}	Acc_{rm}^{te}	Acc_{fg}^{te}	MIA	Avg.G. $\downarrow$
f_{pre}	$\checkmark$	$\checkmark$	99.63 ± 0.01	99.75 ± 0.20	74.54 ± 0.16	74.83 ± 8.14	96.63 ± 1.02	-
f_{exact}	$\checkmark$	$\times$	99.67 ± 0.03 (0.00)	0.00 ± 0.00 (0.00)	75.44 ± 0.19 (0.00)	0.00 ± 0.00 (0.00)	0.00 ± 0.00 (0.00)	0.00
GA	$\times$	$\checkmark$	88.40 ± 5.51 (11.27)	23.12 ± 12.14 (23.12)	65.27 ± 3.56 (10.17)	18.82 ± 7.52 (18.82)	19.95 ± 10.24 (19.95)	16.67
GA	$\checkmark$	$\checkmark$	99.49 ± 0.03 (0.18)	0.02 ± 0.03 (0.02)	74.57 ± 0.21 (0.87)	0.00 ± 0.00 (0.00)	0.00 ± 0.00 (0.00)	0.21
RL	$\times$	$\checkmark$	92.94 ± 1.91 (6.73)	9.98 ± 7.44 (9.98)	67.17 ± 1.46 (8.17)	7.50 ± 7.57 (7.50)	1.83 ± 0.54 (1.83)	6.85
RL	$\checkmark$	$\checkmark$	99.91 ± 0.01 (0.24)	1.52 ± 1.17 (1.52)	74.43 ± 0.18 (1.01)	0.33 ± 0.58 (0.33)	0.00 ± 0.00 (0.00)	0.62
SalUn	$\times$	$\checkmark$	93.16 ± 1.90 (6.51)	13.55 ± 10.30 (13.55)	67.49 ± 1.56 (7.95)	10.00 ± 9.26 (10.00)	3.45 ± 0.87 (3.45)	8.29
SalUn	$\checkmark$	$\checkmark$	99.88 ± 0.01 (0.21)	2.57 ± 1.09 (2.57)	74.71 ± 0.19 (0.73)	1.00 ± 0.87 (1.00)	0.00 ± 0.00 (0.00)	0.90
COUN+RL	$\times$	$\checkmark$	92.56 ± 0.75 (7.11)	10.07 ± 3.85 (10.07)	67.85 ± 0.51 (7.59)	5.67 ± 2.57 (5.67)	2.28 ± 0.46 (2.28)	6.54
COUN+RL	$\checkmark$	$\checkmark$	99.94 ± 0.01 (0.27)	0.73 ± 0.16 (0.73)	72.89 ± 0.18 (1.36)	0.67 ± 0.58 (0.67)	0.00 ± 0.00 (0.00)	0.61
L2UL	$\times$	$\checkmark$	92.69 ± 3.43 (6.98)	4.43 ± 0.15 (4.43)	67.69 ± 2.58 (7.75)	2.00 ± 0.87 (2.00)	2.57 ± 0.92 (2.57)	4.75
DELETE	$\times$	$\checkmark$	99.13 ± 0.12 (0.54)	13.32 ± 4.17 (13.32)	73.31 ± 0.28 (2.13)	6.50 ± 2.18 (6.50)	1.85 ± 0.48 (1.85)	4.88
LOFT	only Σ_{rm}^{pre}	only Σ_{fg}^{pre}	98.59 ± 0.07 (1.08)	1.48 ± 1.27 (1.48)	74.27 ± 0.18 (1.17)	0.50 ± 0.50 (0.50)	0.00 ± 0.00 (0.00)	0.85

Table 2: Comparison on computational costs w.r.t. the results in Table 1. The detailed counts and percentages of optimized parameters are listed. RTE is the run time efficiency in seconds on single NVIDIA GeForce RTX 4090 GPU. For LOFT, RTE measures only the optimization time of (1), as training features can be pre-calculated in practice and covariances are readily available for on-demand use. Refer to Table S1 in Appendix D for a detailed analysis on the full pipeline of LOFT.

method	$\mathcal{D}_{rm}$	$\mathcal{D}_{fg}$	# optimized param. (%)	RTE (seconds) $\downarrow$	additional requirements
f_{pre}	$\checkmark$	$\checkmark$	27,673,154 (100%)	882.05	-
f_{exact}	$\checkmark$	$\times$	27,673,154 (100%)	858.14	-
GA	$\times$	$\checkmark$	27,673,154 (100%)	13.49	-
GA	$\checkmark$	$\checkmark$	27,673,154 (100%)	218.37	-
RL	$\times$	$\checkmark$	27,673,154 (100%)	13.19	-
RL	$\checkmark$	$\checkmark$	27,673,154 (100%)	438.48	-
SalUn	$\times$	$\checkmark$	13,836,577 (50%)	15.00	a parameter saliency mask
SalUn	$\checkmark$	$\checkmark$	13,836,577 (50%)	450.23	a parameter saliency mask
COUN+RL	$\times$	$\checkmark$	27,673,154 (100%)	23.45	-
COUN+RL	$\checkmark$	$\checkmark$	27,673,154 (100%)	704.87	-
L2UL	$\times$	$\checkmark$	27,673,154 (100%)	15225.05	a set of adversarial examples; a parameter importance mask
DELETE	$\times$	$\checkmark$	27,673,154 (100%)	22.14	a copy of the pretrained model
LOFT	only Σ_{rm}^{pre}	only Σ_{fg}^{pre}	192,000 (0.69%)	0.24	a projection matrix $U \in \text{St}(768, 250)$

Effective unlearning performance with significantly reduced computational cost.

Experiments



Class-wise unlearning: $\mathcal{D}_{fg}$ contains 1 of 1,000 classes of ImageNet-1K (1‰)

method	$\mathcal{D}_{rm}$	$\mathcal{D}_{fg}$	Acc_{rm}^{tr}	Acc_{fg}^{tr}	Acc_{rm}^{te}	Acc_{fg}^{te}	Avg.G.↓	RTE ↓	# param. (%)
f_{pre}	✓	✓	77.92 $\pm$ 0.01	83.46 $\pm$ 7.64	76.15 $\pm$ 0.01	81.33 $\pm$ 10.26	-	-	25,557,032 (100%)
f_{exact}	✓	✗	80.25 $\pm$ 0.16 (0.00)	0.00 $\pm$ 0.00 (0.00)	75.72 $\pm$ 0.04 (0.00)	0.00 $\pm$ 0.00 (0.00)	0.00	-	25,557,032 (100%)
GA	✗	✓	71.87 $\pm$ 1.80 (8.38)	0.00 $\pm$ 0.00 (0.00)	71.42 $\pm$ 1.89 (4.30)	0.00 $\pm$ 0.00 (0.00)	3.17	9.32	25,557,032 (100%)
RL	✗	✓	72.11 $\pm$ 1.42 (8.14)	1.87 $\pm$ 0.69 (1.87)	71.53 $\pm$ 1.44 (4.19)	5.33 $\pm$ 4.16 (5.33)	4.88	14.40	25,557,032 (100%)
SalUn	✗	✓	71.56 $\pm$ 0.68 (8.69)	2.67 $\pm$ 0.77 (2.67)	70.84 $\pm$ 0.60 (4.88)	4.67 $\pm$ 3.06 (4.67)	5.23	27.73	12,778,516 (50%)
COUN+RL	✗	✓	70.04 $\pm$ 2.01 (10.21)	1.36 $\pm$ 0.35 (1.36)	69.72 $\pm$ 1.82 (6.00)	4.67 $\pm$ 3.06 (4.67)	5.56	15.24	25,557,032 (100%)
DELETE	✗	✓	72.46 $\pm$ 1.30 (7.79)	0.95 $\pm$ 0.78 (0.95)	71.90 $\pm$ 1.23 (3.82)	2.00 $\pm$ 3.46 (2.00)	3.64	17.82	25,557,032 (100%)
LOFT	only Σ_{rm}^{pre}	only Σ_{fg}^{pre}	79.96 $\pm$ 0.38 (0.29)	0.00 $\pm$ 0.00 (0.00)	69.40 $\pm$ 0.35 (6.32)	0.00 $\pm$ 0.00 (0.00)	1.65	0.94	1,024,000 (4.01%)

Instance-wise unlearning: $\mathcal{D}_{fg}$ contains 1,000 of 100,000 images in Tiny-ImageNet (1%)

method	$\mathcal{D}_{rm}$	$\mathcal{D}_{fg}$	Acc_{rm}^{tr}	Acc_{fg}^{tr}	Acc_{rm}^{te}	MIA	Avg.G.↓	RTE ↓	# param. (%)
f_{pre}	✓	✓	99.63 $\pm$ 0.00	99.80 $\pm$ 0.01	74.55 $\pm$ 0.00	96.20 $\pm$ 0.44	-	882.05	27,673,154 (100%)
f_{exact}	✓	✗	99.67 $\pm$ 0.02 (0.00)	75.97 $\pm$ 1.00 (0.00)	75.03 $\pm$ 0.10 (0.00)	64.47 $\pm$ 1.25 (0.00)	0.00	857.21	27,673,154 (100%)
GA	✗	✓	99.63 $\pm$ 0.00	99.80 $\pm$ 0.01	74.55 $\pm$ 0.00	96.20 $\pm$ 0.44	-	882.05	27,673,154 (100%)
RL	✗	✓	97.56 $\pm$ 0.62 (2.11)	95.00 $\pm$ 1.64 (19.03)	70.57 $\pm$ 0.50 (4.46)	88.83 $\pm$ 0.99 (24.36)	12.49	7.61	27,673,154 (100%)
SalUn	✗	✓	98.47 $\pm$ 0.34 (1.20)	97.23 $\pm$ 1.01 (21.26)	71.86 $\pm$ 0.24 (3.17)	89.60 $\pm$ 1.08 (25.13)	12.69	12.79	13,836,577 (50%)
L2UL	✗	✓	94.77 $\pm$ 0.51 (4.90)	74.90 $\pm$ 0.10 (1.07)	68.69 $\pm$ 0.27 (6.34)	73.90 $\pm$ 0.46 (9.43)	5.44	7919.11	27,673,154 (100%)
COUN+RL	✗	✓	98.30 $\pm$ 0.32 (1.37)	96.47 $\pm$ 0.21 (20.50)	71.45 $\pm$ 0.26 (3.58)	89.02 $\pm$ 0.82 (24.55)	12.50	11.45	27,673,154 (100%)
DELETE	✗	✓	91.84 $\pm$ 2.12 (7.83)	77.57 $\pm$ 2.39 (1.60)	65.44 $\pm$ 1.43 (9.59)	68.67 $\pm$ 1.50 (4.20)	5.81	8.42	27,673,154 (100%)
LOFT	only Σ_{rm}^{pre}	only Σ_{fg}^{pre}	92.65 $\pm$ 0.54 (7.02)	82.97 $\pm$ 1.17 (7.00)	68.14 $\pm$ 0.42 (6.89)	64.47 $\pm$ 2.36 (0.00)	5.23	0.50	230,400 (0.83%)

Ablation studies on $\mathcal{J}_{fg}$ and $\mathcal{J}_{rm}$

- Optimizing $\mathbf{U}$ only with $\mathcal{J}_{fg}$ suppresses the accuracy on forgetting data and MIA, while failing to preserve accuracy on remaining data.
- Employing only $\mathcal{J}_{rm}$ manages to achieve high accuracy on remaining data, but leaves $\mathcal{D}_{fg}$ hardly forgotten with high accuracy and MIA.

Table 5: Ablation study on $\mathcal{J}_{rm}$ and $\mathcal{J}_{fg}$.

objective	Acc_{rm}^{tr}	Acc_{fg}^{tr}	Acc_{rm}^{te}	Acc_{fg}^{te}	MIA
None	97.11	98.10	71.61	65.00	86.10
$\mathcal{J}_{fg}$ only	89.51	11.50	65.40	4.00	0.00
$\mathcal{J}_{rm}$ only	99.19	87.80	75.07	53.00	58.35
$\mathcal{J}_{rm} + \mathcal{J}_{fg}$	98.66	1.10	74.27	0.00	0.00

Sensitivity analysis

- The subspace dimension s determines the percentage of captured feature information by $\mathbf{U} \in \text{St}(d, s)$. A trade-off for varied s between forgetting and utility.
- LOFT maintains competitive performance when applied in earlier positions. It further indicates that the feature covariance in earlier layers already contains sufficient information to achieve effective MU from a low-dimensional feature subspace lens.

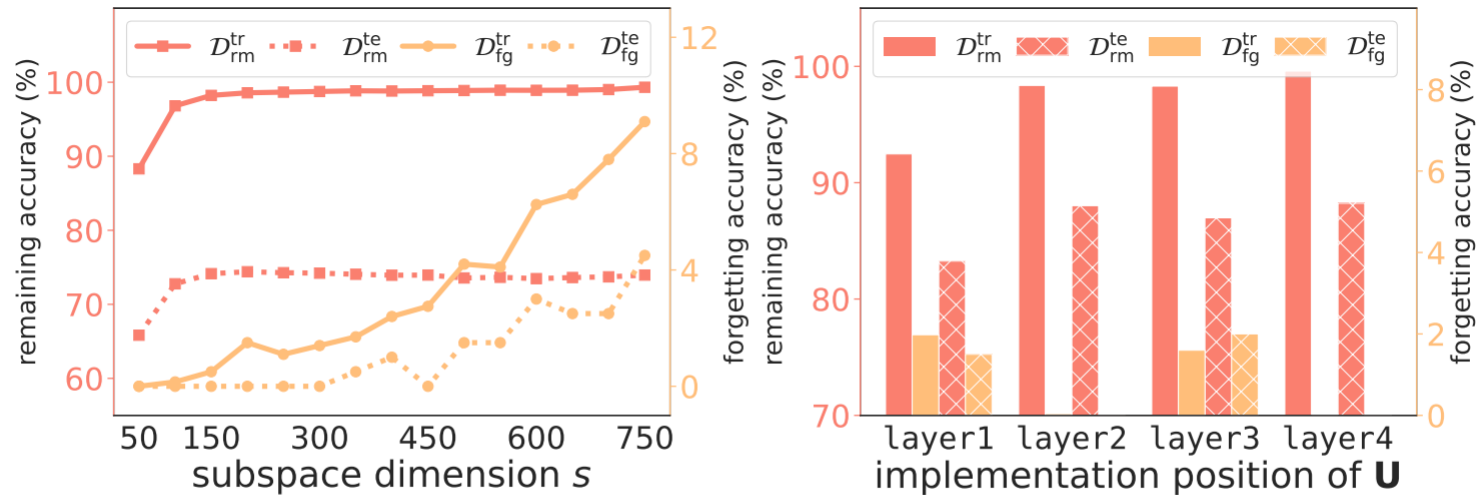


Figure 3: A sensitivity analysis on (i) subspace dimensions s (**left**) and (ii) different positions of the projection matrix $\mathbf{U}$ (**right**).



Summary

- A novel feature subspace perspective to investigate MU

Forgetting and remaining data can exhibit separability in low-dimensional feature subspaces.

- A feature subspace learning method named LOFT for approximate MU
- Learning a feature subspace from pretrained features to capture information from the remaining data while suppressing information from the forgetting data.
- One-shot feature fetching and optimizing only a projection matrix using two fixed covariance matrices, avoiding repeated raw-data access and reducing computations.

Limitation

Its extension to generation tasks is less direct; in such cases, the projection matrix needs to be carefully tailored to the specific generative model and the particular unlearning scenario.



THE HONG KONG
POLYTECHNIC UNIVERSITY
香港理工大學

Thanks!

