One-Turn Knockout: Traceable and Editable Proxy Unlearning Under Asymmetric Access Constraints

Ziluowen Luo[1], **Jun Yin**[2], **Hao Yan**[1], **Ruochen Liu**[1]
Ming Cheng[1] and **Senzhang Wang**[1*]
[1]Central South University
[2]Hong Kong Polytechnic University
{lzlwddl, csuyh1999, ruochen, 244701028, szwang}@csu.edu.cn,
junmay.yin@connect.polyu.hk

Abstract

Machine unlearning (MUL) aims to remove the influence of specific data from a trained model for data privacy and model adaptability. Existing MUL methods mostly assume the internal parameters and the training data of the target model are accessible. Nevertheless, in most practical scenarios, the model provider (MP) and the service operator (SO) are different entities with unequal model access privileges. The MP provides the model, while the SO can only access the model via APIs when handling unlearning requests. Under such an asymmetric access constraint, we propose **O**ne-**T**urn **K**nockout (OTK), a novel traceable and editable MUL framework based on a model-agnostic and editable proxy. Specifically, OTK first compresses the representation space of the target model into a discrete proxy based on codebook, with merely one pass post-training. Each data sample is recorded in the proxy space as a distribution over the codebook tokens, and its contribution to the model prediction can be cumulatively estimated via additive token statistics. Based on the traceable and editable proxy, the SO can instantly handle unlearning requests by *(i)* estimating the token distribution of the forgotten data, *(ii)* identifying the causal tokens, and *(iii)* erasing their contributions without the access to the model parameters and training data. Extensive experiments on multiple datasets and tasks show that OTK consistently outperforms state-of-the-art unlearning methods.

1 Introduction

Currently, personal data, including online posts, browsing histories, and travel records, are imperceptibly collected by various service platforms to train machine learning models such as recommendation systems [Yin *et al.*, 2025; Liu *et al.*, 2024] and scoical networks [Yan *et al.*, 2025]. Along with the promulgation of data privacy regulations, machine unlearning (MUL) which ensures end-users *the right to be forgotten* attracts ever-increasing attention from both academia and industry. In detail, MUL aims to render the model predictions on the forgotten samples inaccurate without performance degradation on the retention data [Xu *et al.*, 2024].

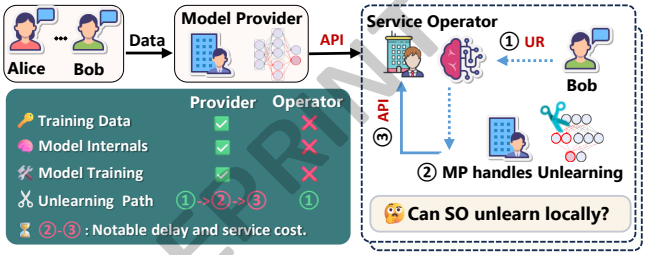

Figure 1: Unlearning Challenges under Asymmetric Access Constraint. The table indicates access privileges and unlearning paths.

Intuitively, retraining models from scratch based on the retention data, excluding forgotten samples, is straightforward and effective, but it incurs prohibitive computational costs. To mitigate this issue, approximate MUL methods provide efficient alternatives. Golatkar *et al.*(2020) located the parameters related to the forgotten samples and removed their accumulative gradients. Graves *et al.*(2021) randomly labeled the sensitive samples and rolled back the parameter updates for their batches. Li *et al.*(2024) designed mutual evolution strategy to forget. Chen *et al.*(2025b) adopted model distillation to extract forgotten student models. Despite successfully accelerate unlearning, they still require access to the original training data and model parameters during the unlearning procedure [Zhang *et al.*, 2024; Shen *et al.*, 2024].

However, as Figure 1 illustrated, in most practical scenarios, the model is trained by the model provider (MP) on aggregated data and then exposed to the service operator (SO) in the API-as-a-service manner. Consequently, the SO, who handles the unlearning requests, are not permitted to directly access the parameters and the training data of the target model. We refer to this setting as an *asymmetric access constraint* [Sun *et al.*, 2022]. Given this constraint, on the one hand, existing MUL methods that depend on direct access to the training data or on operations at model parameters [Yang *et al.*, 2025; Hu *et al.*, 2024] cannot be applied. On the other hand, even though the SO can re-direct unlearning requests to the MP, the induced communication overhead and response latency are unacceptable. Therefore, this paper aims to investigate such a novel research question in MUL:

*Corresponding Author.

How to design a novel machine unlearning method for the service operator under the asymmetric access constraint, without the need of repeatedly accessing to the training data and model parameters?

Achieving this goal necessitates a shifted paradigm: we seek to construct a model-agnostic proxy representation space standing in for the parameter space of the target model. With such a proxy, unlearning requests can be handled by the SO without accessing the model or data. However, realizing this paradigm remains non-trivial due to the following challenges.

First, constructing a traceable and editable proxy space is inherently difficult. The principal barrier is the asymmetric access constraint per se, where the SO cannot access the model parameters and the retained data. This immutability rules out parameter-space editing techniques, while the inaccessibility of retained data makes it infeasible to pinpoint and isolate the exact contributions of the data to be forgotten [Golatkar *et al.*, 2020; Foster *et al.*, 2024b; Foster *et al.*, 2024c; Yang *et al.*, 2025]. Consequently, the proxy must be built to faithfully capture the model's knowledge and enable precise, sample-level edits, with limited access to data.

Second, given a traceable and editable proxy, how to surgically erase targeted samples without affecting others remains formidable. An over-aggressive erasure may degrade overall performance on retained data, whereas an insufficient update leaves residual traces of the forgotten samples, compromising the unlearning objective [Zhao *et al.*, 2024]. The mechanism must, therefore, achieve a delicate balance between the forgetting effectiveness and the retained utility, and efficiently handle the sequential unlearning requests over time.

In this paper, we propose **O**ne **T**urn **K**nockout (OTK), a traceable and editable proxy MUL framework designed for the asymmetric access constraint. Procedurally, OTK consists of two stages. First, in the proxy post-training stage, OTK requires only one-turn access to construct a discrete codebook-based proxy representation to trace the learned distribution of the model. Each token in the proxy encodes the sample-wise distributional statistics by uniquely binding to a codeword and records the training contribution via additive statistics updates. Second, in the unlearning stage, OTK erases the causal tokens strongly associated with the forget set via editable proxy, misleading the model's prediction on it without affecting retained knowledge. A theoretical analysis is provided to confirm the forgetting performance and the utility stability. Our contributions are as follows:

- We investigate, for the first time, machine unlearning with the asymmetric access constraint, and propose OTK unlearning framework that does not need to access to the training data and the model parameters.
- We propose the codebook-based space as the editable and traceable proxy for knowledge representation. Furthermore, we customize a series of efficient operations to identify and eliminate influential codes for the target sample.
- Extensive experiments conducted across diverse model architectures and datasets consistently demonstrate the superiority of OTK over existing methods.

2 Notation and Problem Formulation

Notation. Let $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$ and $\mathcal{D}_{\text{test}}$ denote the training and test sets where samples are drawn i.i.d. from a joint distribution $\mathcal{P}_{\mathcal{X}\times\mathcal{Y}}$. Let $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ denotes the trained model where $\mathcal{X}$ denotes the input space and $\mathcal{Y}$ denotes a generic output space. The predictor $\mathcal{A}_{\theta,\phi}$ is composed of a feature extractor $E_\theta(\cdot) : \mathcal{X} \to \mathcal{Z} \in \mathbb{R}^D$ and a task head $H_\phi(\cdot) : \mathcal{Z} \to \mathcal{Y}$, where $\theta \in \mathbb{R}^{d_\theta}$ and $\phi \in \mathbb{R}^{d_\phi}$ denote their respective parameters. Following existing work, unlearning methods access the feature extractor and task head through black-box forward queries [Sepahvand *et al.*, 2025].

Problem Formulation. Given a trained model $\mathcal{A}_{\theta,\phi}$ and a designated forget set $\mathcal{D}_f \subset \mathcal{D}_{\text{train}}$, the unlearning method $\mathcal{U}$ produces an updated model $\widetilde{\mathcal{A}} = \mathcal{U}(\mathcal{A}_{\theta,\phi}, \mathcal{D}_f)$. The objective of the unlearning is to remove the influence of $\mathcal{D}_f$ from the model $\mathcal{A}_{\theta,\phi}$, while preserving predictive performance on the retained set $\mathcal{D}_r = \mathcal{D}_{\text{train}} \setminus \mathcal{D}_f$ [Bourtoule *et al.*, 2021]. In real-world applications, unlearning requests typically arrive sequentially over time [Ye and Lu, 2023; Fraboni *et al.*, 2024]. Formally, consider a sequence of forget sets $\{\mathcal{D}_f^{(t)}\}_{t=1}^{T}$, where each $\mathcal{D}_f^{(t)} \subset \mathcal{D}_{\text{train}}$ is revealed at timestep t. The sequential unlearning process generates a sequence of model states $\{\widetilde{\mathcal{A}}^{(t)}\}_{t=0}^{T}$ according to

$$\widetilde{\mathcal{A}}^{(t)} = \begin{cases} \mathcal{A}_{\theta,\phi}, & t = 0, \\ \mathcal{U}\left(\widetilde{\mathcal{A}}^{(t-1)}, \mathcal{D}_f^{(t)}\right), & t \geq 1, \end{cases} \tag{1}$$

where the retained set evolves as $\mathcal{D}_r^{(t)} = \mathcal{D}_r^{(t-1)} \setminus \mathcal{D}_f^{(t)}$ with $\mathcal{D}_r^{(0)} = \mathcal{D}_{\text{train}}$. In this work, we design $\mathcal{U}$ to operate under an asymmetric access constraint, eliminating the need for repeated access to training data or model parameters.

3 Methodology

To address unlearning challenge under the asymmetric access constraint OTK is organized into two separated stages, as illustrated in Figure 2. In the first stage (left), our objective is to construct the traceable and editable proxy representation space that circumvents the problem of the service operator (SO) relying on access to the data and model, thereby enabling subsequent efficient unlearning. Specifically, the model provider (MP) inserts a codebook-based proxy module between the frozen encoder and task head, and only the proxy components are optimizable. With only a single access to the original training data, this proxy is post-trained such that both the data distribution and the training contribution of each sample become explicitly traceable in the proxy space. In the second stage (right), the SO handles sequential unlearning requests locally based on this proxy module. For each unlearning request, OTK estimates the distributional contribution of the forget set in the proxy space, identifies the tokens most strongly associated with the target samples, and selectively updates only those tokens to remove their influence, while keeping the rest of the model intact. The following subsections detail the codebook-based proxy representation (§3.1), the one-turn proxy post-training procedure (§3.2), the subsequent selective surgical erasure mechanism (§3.3).

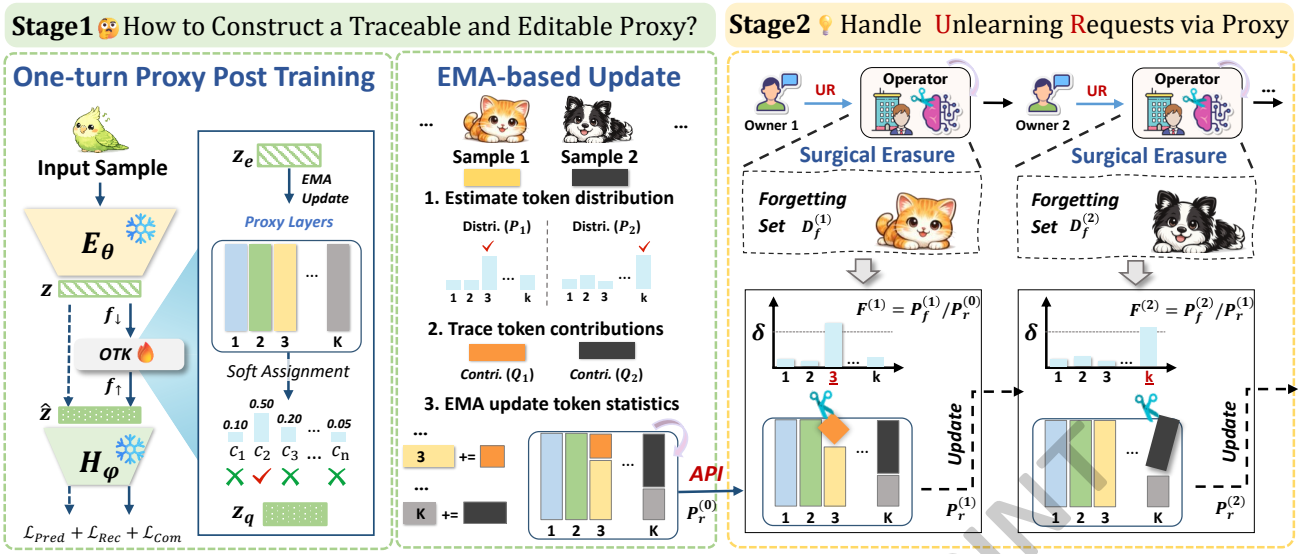

Figure 2: The pipeline of OTK that consists of two stages. In the first stage, the model provider (MP) performs one-turn post-training to construct a traceable and editable codebook-based proxy. In the second stage, the service operator (SO) handles unlearning requests locally by estimating the forget set's token distribution and selectively updating the associated tokens to erase their influence.

3.1 Proxy Representation via Discrete Tokens

To enable precise and instantly local unlearning under restricted access, we introduce an explicit proxy representation composed of discrete and independently editable tokens [Yang *et al.*, 2023]. Formally, given the model $\mathcal{A}_{\theta,\phi}$, we augment it with a model-agnostic codebook placed between the encoder and the task head. The codebook-based proxy module is defined as

$$\mathcal{V} = (C, q, f_{\downarrow}, f_{\uparrow}), \tag{2}$$

where $C = \{c_j\}_{j=1}^{K}$ denotes the learnable codebook of K tokens, $q(\cdot)$ is soft assignment operator, $f_{\downarrow} : \mathbb{R}^D \to \mathbb{R}^d$ and $f_{\uparrow} : \mathbb{R}^d \to \mathbb{R}^D$ are projection functions.

Given an input sample x, the encoder produces the representation $z = E_\theta(x) \in \mathbb{R}^D$. Since direct quantization in high-dimensional latent space is inefficient and unstable, we first project z into a lower-dimensional space via $z_e = f_{\downarrow}(z) \in \mathbb{R}^d$. The projected feature is then softly assigned to the codebook tokens, yielding a proxy representation

$$z_q = q(z_e) = \sum_{j=1}^{K} p(j \mid z_e)\, c_j, \tag{3}$$

where $p(j \mid z_e)$ denotes the assignment probability of z_e to token j. Specifically, the probabilities are computed by

$$p(j \mid z_e) = \frac{\exp\left(-\|z_e - c_j\|_2^2/\tau\right)}{\sum_{l=1}^{K} \exp\left(-\|z_e - c_l\|_2^2/\tau\right)}, \tag{4}$$

where τ controls the sharpness of token assignment. This soft assignment induces a discrete token distribution for each sample, enabling token-level attribution that supports subsequent proxy optimization and editing. Finally, the proxy representation is projected back via $\widetilde{z} = f_{\uparrow}(z_q)$ and passed for prediction,i.e., $\hat{y} = H_\phi(\widetilde{z})$. Throughout this process, the pretrained encoder E_θ and task head H_ϕ remain frozen.

3.2 One-Turn Proxy Post-Training

As shown in Figure 2, the MP performs a one-turn post-training procedure to optimize the proxy representation space. In this section, we seek to obtain a proxy that (i) stably records training footprints, (ii) maintains the behavior of model, and (iii) supports localized edits.

EMA-based Codebook Update. To enable subsequent unlearning, our key design is to record sample-wise distribution and contribution in the proxy space as additive statistics. We realize this recording by using codebook embeddings as discrete carriers of token-level statistics. Concretely, given a minibatch of samples $\mathcal{B} \subseteq \mathcal{D}_{\text{train}}$, we compute the soft assignments $p(j \mid z_e)$ (Eq. (4)) and form batch-wise additive statistics as follows:

$$s_j^{(\mathcal{B})} = \sum_{x \in \mathcal{B}} p(j \mid z_e)\, z_e, \qquad n_j^{(\mathcal{B})} = \sum_{x \in \mathcal{B}} p(j \mid z_e). \tag{5}$$

We then update the statistics with exponential moving average (EMA) using momentum $\gamma \in (0, 1)$ as follows:

$$\hat{s}_j \leftarrow \gamma \hat{s}_j + (1-\gamma) s_j^{(\mathcal{B})}, \qquad \hat{n}_j \leftarrow \gamma \hat{n}_j + (1-\gamma) n_j^{(\mathcal{B})}. \tag{6}$$

Finally, the codeword is updated as following form:

$$c_j \leftarrow \frac{\hat{s}_j}{\hat{n}_j + \varepsilon}, \tag{7}$$

where ε is a small constant for numerical stability.

Interpretation. The accumulated statistic pair $(\hat{s}_j, \hat{n}_j)$ represents an additive summary of how the training data occupy the proxy space through token j, where $\hat{n}_j$ records the effective mass assigned to token j, and $\hat{s}_j$ aggregates the corresponding latent contributions. After the proxy post-training stage, the computed statistics $\{(\hat{s}_j, \hat{n}_j)\}_{j=1}^{K}$ serve as a proxy-level footprint of the training samples over the learned codebook space, which further enables efficient execution of subsequent unlearning operations in section(§3.3).

Objective Design. To construct a traceable and editable proxy space, we propose a disjoint objective to optimize the proxy post-training procedure. The objective consists of three complementary terms. The *prediction consistent objective* $\mathcal{L}_{\text{Pred}}$ enforces that the proxy induces the same prediction as the original model. Formally, $\mathcal{L}_{\text{Pred}}$ is defined as:

$$\mathcal{L}_{\text{Pred}} = \left\| H_\phi(z) - H_\phi(\tilde{z}) \right\|_2^2, \tag{8}$$

where $z = E_\theta(x)$ and $\tilde{z} = f_\uparrow(z_q)$. The *geometric structure objective* $\mathcal{L}_{\text{Rec}}$ is introduced to align the proxy space with the model's latent representation space. Specifically, it encourages the softly quantized proxy representation to remain close to the original continuous embedding:

$$\mathcal{L}_{\text{Rec}} = \|z_e - z_q\|_2^2, \tag{9}$$

where $z_q = \sum_{j=1}^{K} p(j \mid z_e) c_j$. The *commitment objective* $\mathcal{L}_{\text{Com}}$ enforces a stable and localized token attribution by encouraging each sample to commit to a small subset of proxy tokens, while simultaneously discouraging redundancy among codewords via a diversity regularizer:

$$\mathcal{L}_{\text{Com}} = \mathcal{L}_{\text{CE}}(j^\star,\, p(\cdot \mid z_e)) + \beta \left\| CC^\top - I \right\|_2^2, \tag{10}$$

where $j^\star = \arg\min_j \|z_e - c_j\|_2$ is the nearest token index. This term promotes low-entropy, peaked assignment distributions, ensuring that sample-level contributions are concentrated and traceable. The overall objective is given by:

$$\mathcal{L} = \mathcal{L}_{\text{Com}} + \lambda_{\text{Rec}} \cdot \mathcal{L}_{\text{Rec}} + \lambda_{\text{Pred}} \cdot \mathcal{L}_{\text{Pred}}. \tag{11}$$

Together, these objectives yield a proxy representation that is traceable and editable, thereby enabling precise and stable unlearning under asymmetric access.

Once the proxy module is post-trained, the MP provides the proxy along with the corresponding APIs to the the SO. This asymmetric deployment allows the SO to perform unlearning operations locally within the proxy space, reducing communication overhead without exposing the training data or model parameters of the MP.

3.3 Selective Surgical Erasure

With the trained proxy module deployed at the SO side, unlearning is carried out by operating directly on codebook embedding and token-level statistics in the proxy space. This procedure contains three steps: (i) estimating token-level statistics of the forget set, (ii) identifying a small set of causal tokens dominated by the forgotten data, and (iii) performing localized subtraction on the selected tokens.

Token-level Contribution Statistics. To perform unlearning, we need to quantify how a given set of samples occupies the proxy space. For this purpose, we consider an arbitrary subset $\mathcal{T} \subseteq \mathcal{D}_{\text{train}}$, which can represent the forget set, the retained set or the full training set at different stages.

According to the trained proxy, each sample $x \in \mathcal{T}$ induces a token assignment distribution $p(j \mid z_e)$ and a downsampled representation z_e. We summarize the contribution of $\mathcal{T}$ in the proxy space using token-level sufficient statistics. Specifically, for each token $j \in \{1, \ldots, K\}$, we define:

$$s_{\mathcal{T},j} = \mathbb{E}_{z_e \sim \mathcal{T}}[p(j \mid z_e)\, z_e], n_{\mathcal{T},j} = \mathbb{E}_{z_e \sim \mathcal{T}}[p(j \mid z_e)], \tag{12}$$

where the expectation is empirically estimated using forward queries of the trained proxy. Collectively, these statistics form

$$Q_{\mathcal{T}} = \{(s_{\mathcal{T},j}, n_{\mathcal{T},j})\}_{j=1}^{K}, \tag{13}$$

which retains the full contribution information required for codeword rollback. The assignment masses further induce a token distribution

$$P_{\mathcal{T}} = \big(P_{\mathcal{T},1}, \ldots, P_{\mathcal{T},K}\big), \qquad P_{\mathcal{T},j} = \frac{n_{\mathcal{T},j}}{\sum_{k=1}^{K} n_{\mathcal{T},k}}, \tag{14}$$

which provides a token-level assignment profile of $\mathcal{T}$ and is used for causal token identification. Prior to any unlearning requests, we initialize the system with the trained proxy $(\mathcal{A}, C^{(0)})$ and set the initial retained set as $\mathcal{D}_r^{(0)} = \mathcal{D}_{\text{train}}$, together with its corresponding statistics $Q_r^{(0)}$ and $P_r^{(0)}$.

Causal Token Selection. Consider the t-th unlearning request with forget set $\mathcal{D}_f^{(t)}$ with its token distribution $P_f^{(t)}$. Since the proxy attribution is additive, the token distribution of the updated retention set can be recursively maintained as

$$P_r^{(t)} = \frac{P_r^{(t-1)} N_r^{(t-1)} - P_f^{(t)} N_f^{(t)}}{N_r^{(t)}}, \tag{15}$$

where $N_f^{(t)} = |\mathcal{D}_f^{(t)}|$ and $N_r^{(t)} = N_r^{(t-1)} - N_f^{(t)}$. To identify which tokens are primarily responsible for encoding the forget set, we define a token j to be *causal* if its relative contribution $F_j^{(t)}$ exceeds a threshold δ:

$$j \in \mathcal{J}_f^{(t)} \iff F_j^{(t)} = \frac{P_{f,j}^{(t)}}{P_{r,j}^{(t-1)}} \geq \delta. \tag{16}$$

This criterion isolates tokens whose attribution mass is dominated by the forget set, leveraging the structural separability encouraged during proxy training.

Surgical Erasure. After identified the causal token set $\mathcal{J}_f^{(t)}$, we then remove the forget set's contribution from the proxy space. Based on the additive statistics $Q_f^{(t)}$ estimated from the forget set, we update the retention statistics as

$$s_{r,j}^{(t)} = s_{r,j}^{(t-1)} - s_{f,j}^{(t)}, \quad n_{r,j}^{(t)} = n_{r,j}^{(t-1)} - n_{f,j}^{(t)}. \tag{17}$$

For each causal token $j \in \mathcal{J}_f^{(t)}$, we then update the corresponding codeword via

$$c_j^{(t)} = \frac{s_{r,j}^{(t)}}{n_{r,j}^{(t)}}, \tag{18}$$

with all non-causal codewords remain unchanged, i.e., $c_j^{(t)} = c_j^{(t-1)}$ for $j \notin \mathcal{J}_f^{(t)}$ and $C^{(t)} = \{c_1^{(t)}, c_2^{(t)}, \ldots, c_K^{(t)}\}$.

By subtracting the contribution of the forget set from the selected proxy tokens, we directly remove its influence from the proxy statistics. This operation modifies the proxy representation consumed by the frozen backbone, resulting in localized changes to the model's predictions. Crucially, unlearning is achieved without updating model parameters or accessing the original training data.

Models	Metrics	Baseline	Retrain	Fintune	NegGrad	Lipschitz	Blindspot	SSD	LFSSD	Fisher	Amnesiac	UNSIR	OTK(Ours)
MN	$\mathbf{C}_r$	84.81±0.00	85.43±0.31	86.15±0.17	8.96±0.00	13.69±0.64	81.58±0.19	84.81±0.00	84.81±0.00	78.78±1.98	85.99±0.14	82.76±0.21	**85.45**±0.00
	$\mathbf{C}_f$	91.42±0.00	0.00±0.00	3.27±0.57	**0.00**±0.00	**0.00**±0.00	12.35±0.55	91.42±0.00	91.42±0.00	**0.00**±0.00	**0.00**±0.00	77.64±1.20	**0.00**±0.00
	$\mathbf{C}_t$	85.20±0.00	76.71±0.28	77.65±0.14	8.32±0.00	11.79±0.48	74.45±0.22	85.20±0.00	85.20±0.00	70.89±1.70	77.19±0.12	82.76±0.22	**76.71**±0.00
	MIA	78.80±0.00	37.45±1.47	28.88±5.85	**0.00**±0.00	33.17±4.37	2.08±0.11	78.80±0.00	78.80±0.00	33.43±10.95	11.52±0.06	58.19±0.93	1.08±0.00
RN18	$\mathbf{C}_r$	89.18±0.00	90.26±0.06	89.72±0.09	10.85±0.00	10.07±0.32	88.66±0.24	89.18±0.00	89.18±0.00	10.85±0.00	88.49±0.59	86.64±0.43	**90.27**±0.00
	$\mathbf{C}_f$	94.03±0.00	0.00±0.00	3.44±1.38	**0.00**±0.00	**0.00**±0.00	6.34±2.74	94.03±0.00	94.03±0.00	**0.00**±0.00	**0.00**±0.00	52.81±12.33	**0.00**±0.00
	$\mathbf{C}_t$	89.48±0.00	81.12±0.06	80.95±0.19	10.02±0.00	9.24±0.31	80.29±0.39	89.48±0.00	89.48±0.00	10.02±0.00	79.47±0.54	83.18±1.55	**81.16**±0.00
	MIA	88.62±0.00	30.71±1.02	22.94±4.30	**0.00**±0.00	32.17±4.02	0.01±0.01	88.62±0.00	88.62±0.00	**0.00**±0.00	0.30±0.34	25.36±4.09	0.24±0.00
RN34	$\mathbf{C}_r$	88.76±0.00	90.21±0.05	88.82±0.21	10.85±0.00	5.79±0.17	87.99±0.36	84.76±0.00	81.76±0.00	10.97±0.17	84.45±0.41	85.64±1.00	**90.59**±0.00
	$\mathbf{C}_f$	93.92±0.00	0.00±0.00	23.35±4.90	**0.00**±0.00	68.89±0.91	12.17±9.29	61.77±0.00	93.92±0.00	**0.00**±0.00	**0.00**±0.00	18.53±0.00	**0.00**±0.00
	$\mathbf{C}_t$	89.11±0.00	81.08±0.05	82.08±0.33	10.06±0.00	12.26±0.15	80.27±1.08	82.46±0.00	89.11±0.00	10.14±0.19	79.53±0.40	78.88±0.90	**81.40**±0.00
	MIA	87.54±0.00	26.09±1.86	24.76±2.00	**0.00**±0.00	36.31±1.04	0.06±0.04	81.22±0.00	87.54±0.00	38.70±28.32	0.53±0.28	26.80±10.11	22.82±0.00
RN50	$\mathbf{C}_r$	85.72±0.00	88.55±0.20	84.21±1.46	10.85±0.00	10.76±0.24	83.50±0.92	80.33±0.00	85.72±0.00	10.85±0.03	84.75±1.12	79.02±1.55	**89.47**±0.00
	$\mathbf{C}_f$	92.41±0.00	0.00±0.00	2.52±2.44	**0.00**±0.00	**0.00**±0.00	5.18±3.33	31.77±0.00	92.41±0.00	**0.00**±0.00	**0.00**±0.00	3.16±2.39	**0.00**±0.00
	$\mathbf{C}_t$	86.20±0.00	79.55±0.19	82.08±0.33	10.06±0.00	9.92±0.24	75.50±0.83	75.47±0.00	86.20±0.00	10.03±0.47	76.08±1.01	71.35±1.73	**80.42**±0.00
	MIA	86.24±0.00	27.32±1.47	49.13±9.99	**0.00**±0.00	34.46±4.72	0.22±0.20	44.47±0.00	86.24±0.00	**0.00**±0.00	0.62±0.22	35.39±21.65	0.56±0.00
RN101	$\mathbf{C}_r$	88.38±0.00	90.26±0.13	87.36±0.45	10.85±0.00	9.57±0.45	85.73±0.65	88.38±0.00	88.38±0.00	21.85±0.77	86.73±0.91	80.72±2.15	**89.93**±0.00
	$\mathbf{C}_f$	94.53±0.00	0.00±0.00	13.95±5.06	**0.00**±0.00	**0.00**±0.00	7.62±1.95	94.53±0.00	94.53±0.00	**0.00**±0.00	**0.00**±0.00	1.48±1.31	**0.00**±0.00
	$\mathbf{C}_t$	88.79±0.00	81.15±0.11	79.78±0.63	10.06±0.00	8.73±0.47	77.75±0.58	88.79±0.00	88.79±0.00	16.83±0.53	77.91±0.79	72.72±1.96	**80.76**±0.00
	MIA	90.38±0.00	23.14±1.41	37.83±4.91	**0.00**±0.00	32.38±2.79	0.13±0.09	90.38±0.00	90.38±0.00	48.73±0.45	0.71±0.24	40.84±23.47	0.40±0.00
RN152	$\mathbf{C}_r$	88.52±0.00	89.73±0.11	88.46±0.24	10.85±0.00	10.80±0.34	86.21±0.56	88.52±0.00	88.52±0.00	37.42±4.63	89.31±0.89	81.71±1.20	**89.86**±0.00
	$\mathbf{C}_f$	94.72±0.00	0.00±0.00	13.18±6.11	**0.00**±0.00	**0.00**±0.00	7.02±8.56	94.72±0.00	94.72±0.00	**0.00**±0.00	**0.00**±0.00	12.29±8.31	**0.00**±0.00
	$\mathbf{C}_t$	88.93±0.00	80.61±0.11	80.72±0.60	10.06±0.00	9.96±0.32	78.11±1.17	88.93±0.00	88.93±0.00	35.94±4.84	78.48±0.76	74.71±0.53	**80.67**±0.00
	MIA	88.24±0.00	24.45±0.68	30.09±9.62	**0.00**±0.00	19.99±12.34	0.18±0.12	88.24±0.00	88.24±0.00	47.85±0.00	0.66±0.35	24.24±13.06	0.50±0.00

Table 1: Class truck unlearning on CIFAR-10 across diverse models,i.e., MobileNet (MN) and ResNets (RN). Each result reports the mean ± std deviation across repeated experiments. Retraining is treated as the gold standard, with closer performance indicating better unlearning quality. Oracle retrain results are gold standard, while the best and second-best performances are highlighted in bold and underline.

4 Experiment

In this section, we conduct extensive experiments aiming to answer the following questions.

- **RQ1:** What is the trade-off between forgetting effectiveness and retention performance achieved by OTK?
- **RQ2:** How well does OTK maintain model performance under sequential unlearning requests ?
- **RQ3:** What behavioral changes are induced by OTK after performing unlearning?

4.1 Experimental Setup

Datasets and Backbones. We conduct evaluations across diverse model architectures on seven datasets from various domains. In detail, for *image classification*, the unlearning target models include MobileNet [Li *et al.*, 2022] and ResNet [He *et al.*, 2016] of various scales trained on CIFAR-10, CIFAR-20, CIFAR-100, Flowers and Pets, in line with [Chundawat *et al.*, 2023]. To verify the flexibility of OTK across domains, we further evaluate the performance on additional tasks, including question answering on Instruct GPT [Ouyang *et al.*, 2022] and image generation benchmarks. For *image generation*, the target models contain trained VQ-VAE [Van Den Oord *et al.*, 2017] and VQ-GAN [Feng *et al.*, 2025; Yang *et al.*, 2025] on Digits and LFW-face dataset.

Baselines. We evaluate OTK against three categories. *(i)* Critical anchors: unaltered original model (**Baseline**), Retraining (**Retrain**) as the gold standard, naive Fine-tuning (**Finetune**) as the forgetting lower bound. *(ii)* Fisher Information Matrix based methods: **Fisher** [Golatkar *et al.*, 2020], **SSD** [Foster *et al.*, 2024b] and **LFSSD** [Foster *et al.*, 2024c]. *(iii)* Re-optimization based methods: **Lipschitz** [Foster *et al.*, 2024a] , **amnesiac** [Graves *et al.*, 2021] (component-wise retraining), **NegGrad** [Golatkar *et al.*, 2020] , **BlindSpot** [Tarun *et al.*, 2023b] and **UNSIR** [Tarun *et al.*, 2023a]. All experiments follow the same training and evaluation protocols, including identical train/test splits, consistent hyperparameter settings for each unlearning method, and five independent runs to report mean and standard deviation.

Metrics. We define the unlearning scenarios following SSD [Foster *et al.*, 2024b]. In the full-class unlearning setup, the request targets all samples of the "truck" class in CIFAR-10. In the sequential unlearning scenario, we construct a sequence of consecutive forgetting requests on CIFAR-100 and Flowers, each request removing one distinct class in turn. We report the accuracy on the forgotten classes $\mathbf{C}_f$, the retained classes $\mathbf{C}_r$, the overall test set $\mathbf{C}_t$, and the membership inference attack (MIA) success rate. Following prior works, Retrain serves as the gold-standard reference, where an ideal unlearning method should minimize accuracy on forgotten classes and maximize retention performance while matching the MIA behavior of Retrain as closely as possible.

4.2 Forgetting–Retention Performance (RQ1)

To answer RQ1, we perform a comprehensive comparison across multiple datasets and backbones.

Datasets	Metrics	Baseline	Retrain	Fintune	NegGrad	Lipschitz	Blindspot	SSD	LFSSD	Fisher	Amnesiac	UNSIR	OTK(Ours)
Cifar20	$\mathbf{C}_r$	79.47±0.00	83.99±0.08	79.60±0.28	5.14±0.13	5.53±0.29	78.11±0.28	80.57±0.00	80.67±0.00	5.14±0.46	76.16±0.64	73.56±1.11	**83.44**±0.00
	$\mathbf{C}_f$	83.20±0.00	0.00±0.00	15.20±2.16	**0.00**±0.00	**0.00**±0.00	2.64±0.61	2.60±0.00	46.60±0.00	**0.00**±0.00	**0.00**±0.00	19.68±4.45	**0.00**±0.00
	$\mathbf{C}_t$	**79.67**±0.00	79.58±0.09	76.12±0.26	5.05±0.00	5.34±0.28	74.10±0.25	76.44±0.00	78.76±0.00	5.020±0.61	72.10±0.61	70.86±1.16	79.17±0.00
	MIA	83.28±0.00	18.80±0.08	22.02±6.06	**0.00**±0.00	57.23±11.74	0.04±0.07	18.40±0.00	22.84±0.00	**0.00**±0.00	1.73±0.66	37.62±7.90	1.00±0.00
Cifar100	$\mathbf{C}_r$	75.50±0.00	78.60±0.13	75.91±0.15	0.98±0.13	5.80±0.14	71.04±0.24	73.49±0.00	75.44±0.00	10.85±1.27	69.98±0.30	71.30±0.61	**79.03**±0.00
	$\mathbf{C}_f$	73.00±0.00	0.00±0.00	51.80±0.84	**0.00**±0.00	**0.00**±0.00	**0.00**±0.00	**0.00**±0.00	**0.00**±0.00	**0.00**±0.00	**0.00**±0.00	4.00±2.00	**0.00**±0.00
	$\mathbf{C}_t$	75.47±0.00	77.81±0.00	75.96±0.16	1.00±0.00	5.22±0.14	70.77±0.27	73.05±0.00	74.99±0.00	9.97±0.24	69.60±0.41	70.61±0.63	**78.48**±0.00
	MIA	94.00±0.00	0.20±0.00	6.92±1.82	**0.00**±0.00	28.80±1.61	**0.00**±0.00	1.80±0.02	**0.00**±0.00	**0.00**±0.00	10.08±2.14	4.16±0.92	0.00±0.00
Flowers	$\mathbf{C}_r$	75.71±0.00	76.14±0.11	**76.32**±0.11	0.98±0.00	9.74±0.18	73.94±0.28	72.87±0.00	75.27±0.00	5.13±0.98	71.72±0.35	72.97±0.24	77.57±0.00
	$\mathbf{C}_f$	68.00±0.00	0.00±0.00	59.40±1.34	**0.00**±0.00	1.80±0.45	0.40±0.55	0.17±0.04	**0.00**±0.00	**0.00**±0.00	**0.00**±0.00	11.40±4.22	**0.00**±0.00
	$\mathbf{C}_t$	75.87±0.00	75.59±0.10	76.38±0.10	1.01±0.00	9.23±0.18	73.43±0.17	72.52±0.00	**74.81**±0.00	4.98±0.62	71.33±0.46	72.35±0.24	77.03±0.00
	MIA	91.82±0.00	0.80±0.00	44.76±3.28	**0.00**±0.00	36.24±0.78	**0.00**±0.00	2.05±0.00	**0.00**±0.00	14.17±1.77	40.20±1.97	4.28±2.08	1.00±0.00
Pets	$\mathbf{C}_r$	81.57±0.00	86.74±0.08	82.85±0.06	7.48±0.08	5.24±0.13	81.47±0.24	82.22±0.00	82.20±0.00	5.13±0.98	79.88±0.13	79.72±0.28	**87.39**±0.00
	$\mathbf{C}_f$	82.60±0.00	0.00±0.00	42.32±1.59	**0.00**±0.00	**0.00**±0.00	2.40±0.97	9.40±0.00	34.80±0.00	**0.00**±0.00	**0.00**±0.00	42.80±1.19	**0.00**±0.00
	$\mathbf{C}_t$	81.52±0.00	82.40±0.09	80.62±0.08	78.40±0.09	5.08±0.12	77.27±0.22	78.38±0.00	79.62±0.00	4.98±0.62	75.70±0.13	77.81±0.22	**83.02**±0.00
	MIA	86.76±0.00	10.97±0.73	14.77±1.11	10.97±0.73	29.11±1.97	**0.00**±0.00	6.24±0.73	7.72±0.73	0.00±0.00	5.29±0.70	32.27±2.93	1.00±0.00

Table 2: Experimental results of single-class unlearning on CIFAR-10, CIFAR-100, Flowers, and Pets. For each dataset, one class is randomly selected as the forgetting target. All methods are evaluated using ResNet-34 as the backbone and the result reports mean ± std.

Robustness across backbone architectures. Table 1 reports full class unlearning results on CIFAR-10 using backbone networks with varying depth and parameter scale, ranging from lightweight models (e.g., MobileNet) to deep architectures (e.g., ResNet152). Across all backbones, OTK consistently maintains balance between forgetting effectiveness and retention performance. On smaller and shallower models, OTK achieves complete forgetting of the target classes while preserving, and in some cases slightly improving, retained-set accuracy. On average, OTK improves retained accuracy by approximately *0.33%* compared to the original baseline, indicating that proxy-based editing does not introduce additional degradation even under limited model capacity.

Generalization across data distributions. Table 2 presents random class unlearning results on ResNet34 across datasets with different data scales and class granularities. On challenging datasets such as CIFAR-100 and Pets, which involve a larger number of classes and finer-grained visual distinctions, OTK demonstrates stronger generalization behavior. In particular, OTK achieves higher retained-set accuracy compared to most baselines, with an average improvement of up to *1.9%* on the test set. These results indicate that OTK adapts well to complex decision boundaries.

Comparison with existing methods. When compared with existing unlearning approaches, a clear trade-off pattern emerges. Methods that emphasize aggressive forgetting often achieve low C_f or MIA scores, but suffer from pronounced degradation in retained accuracy and test performance, indicating impaired generalization. In contrast, methods designed to preserve overall utility tend to maintain higher C_r and C_t, yet leave detectable traces of the forgotten data, as reflected by non-zero forgetting metrics.

4.3 Sequential Unlearning Performance (RQ2)

To answer RQ2, we evaluate the retained-set accuracy as a function of the number of unlearning rounds on (a) CIFAR-100, (b) Flowers, and (c) Pets using ResNet-50 backbones.

Dataset	Method	Mobilenet	ResNet18	ResNet50	ResNet101
Cifar20	NegGrad	9.7	11.9	40.2	143.7
	SSD	11.2	57.5	246.7	676.8
	OTK	**3.7 (×2.6)**	**3.7 (×3.2)**	**3.9 (×10.3)**	**3.8 (×38.8)**
Cifar100	NegGrad	14.6	27.8	66.0	203.1
	SSD	11.9	64.3	299.2	765.7
	OTK	**2.7 (×4.4)**	**3.0 (×9.2)**	**2.9 (×22.7)**	**3.4 (×59.7)**

Table 3: Execution time (s) per-request under sequential unlearning.

In each round, a new disjoint subset of training samples is removed from the model. As shown in Figure 3, most baseline methods suffer from rapid performance degradation as unlearning requests accumulate. NegGrad and Lipschitz collapse within the first five rounds, while SSD and Finetune degrade steadily to near-zero accuracy after approximately 8–10 rounds. Amnesic exhibits a slower decline, but still drops from around 80% to 0% after 12 rounds. In contrast, OTK consistently preserves retained-set performance throughout the entire sequence, decreasing only from approximately 75% to 58% after 15 unlearning rounds, and remains close to the retraining upper bound. OTK is also substantially more efficient per unlearning request. As reported in Table 4, OTK requires only 3.7 seconds per request, compared to 9.7–11.9 seconds for NegGrad and 11.2–676.8 seconds for SSD, depending on model depth. Importantly, the runtime of OTK remains nearly constant across architectures, indicating that its unlearning cost is largely model-agnostic.

4.4 Case Study (RQ3)

We further conduct case studies on LLM-based QA and image generation, to qualitatively and quantitatively validate the generality of OTK beyond discriminative vision tasks.

LLM QA Unlearning. We simulate a personalized QA scenario using GPT-2 with 6,000 user profiles, where public and private information are separated at the attribute [Maini *et al.*, 2024]. We report Forget (accuracy on private attributes), Leak (prompt-based extraction success rate) and Retain (ex-

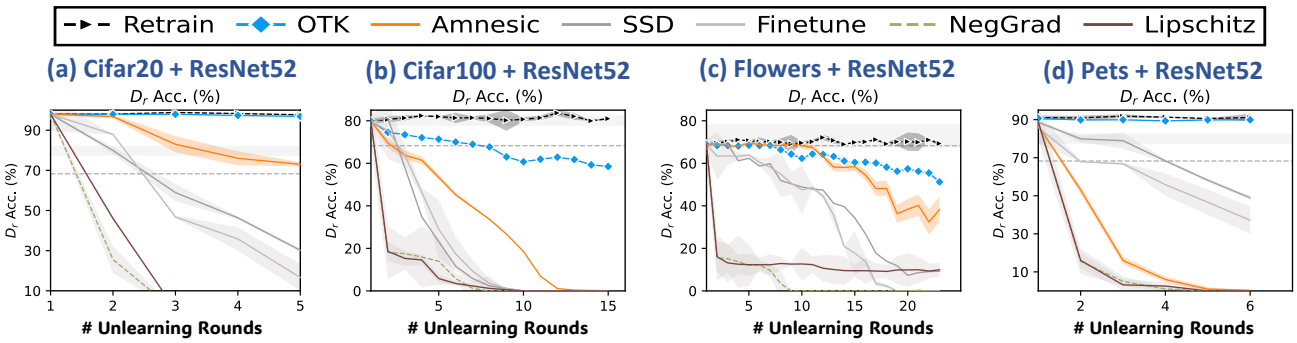

Figure 3: Performance under sequential unlearning requests. The black curves denote Retrain, while the blue curves represent OTK. The shaded regions indicate the variance across different random seeds. For sequential forgetting, we randomly select 15%–25% of classes as consecutive unlearning targets on each benchmark, including 15 classes on CIFAR-100, 23 classes on Flowers, and 6 classes on Pets.

Method	Forget ↓	Leak ↓	Retain ↑
Baseline	0.84	0.77	0.95
NegGrad	0.00	0.00	0.00
Retrain	0.00	0.00	0.94
OTK($\delta = 1.0$)	**0.00**	**0.00**	**0.89**
OTK($\delta = 1.5$)	**0.00**	**0.00**	**0.92**
OTK($\delta = 2.0$)	**0.03**	**0.09**	**0.94**

Table 4: LLM unlearning results in a personalized QA setting.

act match accuracy on public). The Baseline exhibits strong memorization of private attributes, confirming significant privacy leakage while NegGrad completely suppresses both Forget and Leak, indicating that aggressive gradient reversal effectively destroys both private and public knowledge representations. As shown in Table 4, OTK achieves complete forgetting and zero leakage on the private field, while maintaining strong utility, substantially outperforming joint training (0.80) and NegGrad (0.00). OTK achieves a favorable privacy–utility trade-off under different δ values. With $\delta = 1.5$, it matches retraining in privacy removal while preserving high retained performance. Empirically, small δ leads to overly aggressive updates that slightly degrade retention, while $\delta = 1.5$ yields the best balance between forgetting and utility. Further increasing δ to 2.0 weakens unlearning and results in residual memorization, indicating incomplete removal of sensitive associations.

5 Related Works

Selective Unlearning. Machine unlearning aims to remove the influence of specific training samples while preserving the utility of the remaining knowledge. Existing methods mainly follow three paradigms. Retraining-based methods such as SISA [Bourtoule *et al.*, 2021; Schelter *et al.*, 2021] approximate exact forgetting by retraining on retained subsets, but incur substantial computational overhead. To avoid this issue, parameter-editing approaches identify the influence of the target samples and suppress parameters associated with them through weight perturbation, Fisher-based dampening, or local regularization [Chen *et al.*, 2025a; Shibata *et al.*, 2021; Sekhari *et al.*, 2021]. Optimization-based methods further perform gradient rollback, adversarial forgetting, or representation erasure to reduce the influence of target data [Wang *et al.*, 2024; Fan *et al.*, 2024; Liu *et al.*, 2023]. However, they still require access to model parameters, gradients, optimization trajectories, or retained training data during unlearning.

Asymmetric Access Constraints. With the rapid growth of data scale and increasingly constrained real-world deployment environments [Yin *et al.*, 2023], recent studies have begun exploring restricted-access unlearning settings where the unlearning operators may only access deployed models through APIs without exposing model internals or original training data [Sun *et al.*, 2022; Cao and Yang, 2015]. Data-free and restricted-access approaches attempt to relax dependence on retained data through score distillation, synthetic proxy generation [Chen *et al.*, 2025b; Fraboni *et al.*, 2024], or deployment-aware optimization [Zhang *et al.*, 2024; Hu *et al.*, 2024]. Meanwhile, recent scalable unlearning studies further investigate practical challenges such as sequential forgetting, utility preservation, and deployment efficiency under repeated requests [Sepahvand *et al.*, 2025; Kim *et al.*, 2024]. Despite effective, these methods still rely on provider-side coordination, auxiliary optimization, or partial access to model internals, limiting their applicability when both model parameters and original training data are inaccessible [Shi *et al.*, 2024; Gupta *et al.*, 2021].

6 Conclusion

Under asymmetric access constraints, we proposed OTK, a proxy unlearning framework that enables traceable and localized information erasure without data access. By tracking training contributions as additive token-level statistics, OTK supports efficient and precise unlearning through direct proxy edits. Extensive experiments across classification, language, and generative tasks demonstrate that OTK achieves effective forgetting with strong retention and low, model-agnostic runtime overhead. These results highlight OTK as a practical and scalable solution for real-world unlearning deployments. Many future directions could be discussed, particularly extending OTK to large language models, where capturing complex semantics within discrete proxies and exploring adaptive proxy architectures remains a significant challenge.

Acknowledgements

This research was funded by the National Natural Science Foundation of China (No.62572489).

References

[Bourtoule *et al.*, 2021] Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, 2021.

[Cao and Yang, 2015] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, 2015.

[Chen *et al.*, 2025a] Huiqiang Chen, Tianqing Zhu, Xin Yu, and Wanlei Zhou. Zero-shot machine unlearning with proxy adversarial data generation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2025.

[Chen *et al.*, 2025b] Tianqi Chen, Shujian Zhang, and Mingyuan Zhou. Score forgetting distillation: A swift, data-free method for machine unlearning in diffusion models. In *International Conference on Learning Representations*, 2025.

[Chundawat *et al.*, 2023] Vikram S Chundawat, Ayush K Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[Fan *et al.*, 2024] Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *International Conference on Learning Representations*, 2024.

[Feng *et al.*, 2025] XiaoHua Feng, Yuyuan Li, Chaochao Chen, Li Zhang, Li Li, Jun Zhou, and Xiaolin Zheng. Controllable unlearning for image-to-image generative models via ϵ-constrained optimization. In *International Conference on Learning Representations*, 2025.

[Foster *et al.*, 2024a] Jack Foster, Kyle Fogarty, Stefan Schoepf, Zack Dugue, Cengiz Öztireli, and Alexandra Brintrup. An information theoretic approach to machine unlearning. *arXiv preprint arXiv:2402.01401*, 2024.

[Foster *et al.*, 2024b] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI conference on artificial intelligence*, 2024.

[Foster *et al.*, 2024c] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Loss-free machine unlearning. In *The Second Tiny Papers Track at ICLR*, 2024.

[Fraboni *et al.*, 2024] Yann Fraboni, Martin Van Waerebeke, Kevin Scaman, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. Sifu: Sequential informed federated unlearning for efficient and provable client unlearning in federated optimization. In *International Conference on Artificial Intelligence and Statistics*, 2024.

[Golatkar *et al.*, 2020] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.

[Graves *et al.*, 2021] Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[Gupta *et al.*, 2021] Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. In *Advances in Neural Information Processing Systems*, 2021.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[Hu *et al.*, 2024] Yuke Hu, Jian Lou, Jiaqi Liu, Wangze Ni, Feng Lin, Zhan Qin, and Kui Ren. Eraser: Machine unlearning in mlaas via an inference serving-aware approach. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024.

[Kim *et al.*, 2024] Hyunjune Kim, Sangyong Lee, and Simon S Woo. Layer attack unlearning: Fast and accurate machine unlearning via layer level attack and knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[Li *et al.*, 2022] Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. In *Advances in neural information processing systems*, 2022.

[Li *et al.*, 2024] Xunkai Li, Yulin Zhao, Zhengyu Wu, Wentao Zhang, Rong-Hua Li, and Guoren Wang. Towards effective and general graph unlearning via mutual evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[Liu *et al.*, 2023] Junxu Liu, Mingsheng Xue, Jian Lou, Xiaoyu Zhang, Li Xiong, and Zhan Qin. Muter: Machine unlearning on adversarially trained models. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2023.

[Liu *et al.*, 2024] Ruochen Liu, Hao Chen, Yuanchen Bei, Qijie Shen, Fangwei Zhong, Senzhang Wang, and Jianxin Wang. Fine tuning out-of-vocabulary item recommendation with user sequence imagination. In *Advances in Neural Information Processing Systems*, 2024.

[Maini *et al.*, 2024] Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*, 2024.

[Ouyang *et al.*, 2022] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in neural information processing systems*, 2022.

[Schelter *et al.*, 2021] Sebastian Schelter, Stefan Grafberger, and Ted Dunning. Hedgecut: Maintaining randomised trees for low-latency machine unlearning. In *Proceedings of the 2021 International Conference on Management of Data*, 2021.

[Sekhari *et al.*, 2021] Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems*, 2021.

[Sepahvand *et al.*, 2025] Nazanin Mohammadi Sepahvand, Eleni Triantafillou, Hugo Larochelle, Doina Precup, James J. Clark, Daniel M. Roy, and Gintare Karolina Dziugaite. Selective unlearning via representation erasure using domain adversarial training. In *International Conference on Learning Representations*, 2025.

[Shen *et al.*, 2024] Shaofei Shen, Chenhao Zhang, Yawen Zhao, Alina Bialkowski, Weitong Chen, and Miao Xu. Label-agnostic forgetting: A supervision-free unlearning in deep models. In *International Conference on Learning Representations*, 2024.

[Shi *et al.*, 2024] Jialei Shi, Kostis Gourgoulias, John F Buford, Sean J Moran, and Najah Ghalyan. Deepclean: machine unlearning on the cheap by resetting privacy sensitive weights using the fisher diagonal. In *European Conference on Computer Vision*, 2024.

[Shibata *et al.*, 2021] Takashi Shibata, Go Irie, Daiki Ikami, and Yu Mitsuzumi. Learning with selective forgetting. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, 2021.

[Sun *et al.*, 2022] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pages 20841–20855. PMLR, 2022.

[Tarun *et al.*, 2023a] Ayush K Tarun, Vikram S Chundawat, Murari Mandal, and Mohan Kankanhalli. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[Tarun *et al.*, 2023b] Ayush Kumar Tarun, Vikram Singh Chundawat, Murari Mandal, and Mohan Kankanhalli. Deep regression unlearning. In *International Conference on Machine Learning*, 2023.

[Van Den Oord *et al.*, 2017] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in neural information processing systems*, 2017.

[Wang *et al.*, 2024] Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. In *Advances in Neural Information Processing Systems*, 2024.

[Xu *et al.*, 2024] Jie Xu, Zihan Wu, Cong Wang, and Xiaohua Jia. Machine unlearning: Solutions and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8(3):2150–2168, 2024.

[Yan *et al.*, 2025] Hao Yan, Chaozhuo Li, Jun Yin, Zhigang Yu, Weihao Han, Mingzheng Li, Zhengxin Zeng, Hao Sun, and Senzhang Wang. When graph meets multimodal: benchmarking and meditating on multimodal attributed graph learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025.

[Yang *et al.*, 2023] Ling Yang, Ye Tian, Minkai Xu, Zhongyi Liu, Shenda Hong, Wei Qu, Wentao Zhang, Bin Cui, Muhan Zhang, and Jure Leskovec. Vqgraph: Rethinking graph representation space for bridging gnns and mlps. In *International Conference on Learning Representations*, 2023.

[Yang *et al.*, 2025] Zhe-Rui Yang, Jindong Han, Chang-Dong Wang, and Hao Liu. Erase then rectify: A training-free parameter editing approach for cost-effective graph unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[Ye and Lu, 2023] Shanshan Ye and Jie Lu. Sequence unlearning for sequential recommender systems. In *Australasian Joint Conference on Artificial Intelligence*, 2023.

[Yin *et al.*, 2023] Jun Yin, Chaozhuo Li, Hao Yan, Jianxun Lian, and Senzhang Wang. Train once and explain everywhere: Pre-training interpretable graph neural networks. In *Advances in Neural Information Processing Systems*, 2023.

[Yin *et al.*, 2025] Jun Yin, Zhengxin Zeng, Mingzheng Li, Hao Yan, Chaozhuo Li, Weihao Han, Jianjin Zhang, Ruochen Liu, Hao Sun, Weiwei Deng, Feng Sun, Qi Zhang, Shirui Pan, and Senzhang Wang. Unleash LLMs Potential for Sequential Recommendation by Coordinating Dual Dynamic Index Mechanism. In *Proceedings of the ACM on Web Conference*, 2025.

[Zhang *et al.*, 2024] Chenhao Zhang, Shaofei Shen, Yawen Zhao, Weitong Tony Chen, and Miao Xu. Geniu: A restricted data access unlearning for imbalanced data. *arXiv preprint arXiv:2406.07885*, 2024.

[Zhao *et al.*, 2024] Kairan Zhao, Meghdad Kurmanji, George-Octavian Barbulescu, Eleni Triantafillou, and Peter Triantafillou. What makes unlearning hard and what to do about it. In *Advances in Neural Information Processing Systems*, 2024.