EncoderMU: Machine Unlearning in Contrastive Learning

Zixin Wang[1], Bing Mi[2], and Kongyang Chen[1,3(✉)]

[1] Institute of Artificial Intelligence and Blockchain, Guangzhou University, Guangzhou, China
kychen@gzhu.edu.cn
[2] Guangdong University of Finance and Economics, Guangzhou, China
[3] Pazhou Lab, Guangzhou, China

Abstract. Machine unlearning is a complex process that necessitates the model to diminish the influence of the training data while keeping the loss of accuracy to a minimum. Despite the numerous studies on machine unlearning in recent years, the majority of them have primarily focused on supervised learning models, leaving research on contrastive learning models relatively underexplored. With the conviction that self-supervised learning harbors a promising potential, surpassing or rivaling that of supervised learning, we set out to investigate methods for machine unlearning centered around contrastive learning models. In this study, we introduce a novel gradient constraint-based approach for training the model to effectively achieve machine unlearning. Our method only necessitates a minimal number of training epochs and the identification of the data slated for unlearning. Remarkably, our approach demonstrates proficient performance not only on contrastive learning models but also on supervised learning models, showcasing its versatility and adaptability in various learning paradigms.

Keywords: Machine Unlearning · Contrastive Learning · Distributed Learning

1 Introduction

In contemporary society, artificial intelligence (AI) has become increasingly pervasive, with numerous AI applications leveraging machine learning models. AI has permeated various aspects of human society, encompassing learning, work, and daily life. However, model privacy has emerged as a significant concern, as models may inadvertently expose individual users' privacy. For example, membership inference attacks capitalize on the discrepancies between training and non-training data predictions to infer whether specific data were utilized for model training, thereby exposing privacy risks. Additional challenges to model privacy and security include backdoor attacks and model adversarial attacks.

J. Cai et al. (Eds.): SPNCE 2023, LNICST 525, pp. 212–223, 2025.
https://doi.org/10.1007/978-3-031-73699-5_15

Moreover, privacy regulations such as the European General Data Protection Regulation (GDPR) [8] afford users the right to request the deletion of their personal data from learning models, a component of the right to be forgotten. The Protection of Personal Information Act (APPI) [3] and Canada's proposed Consumer Privacy Protection Act (CPPA) [6], both mandate the deletion of private information. Erasing data from learning models is a challenging task requiring the selective reversal of the learning process. In the absence of targeted methods, the sole option is retraining the model, a costly and feasible approach only when the original data remains accessible. As a remedial measure, researchers like Cao and Yang, and Bourtoule et al. [2,4] have proposed machine unlearning methods. These techniques partially reverse the learning process, facilitating the retrospective deletion of specific data points, mitigating privacy breaches, and addressing user deletion requests. Yan et al. [10] and Ga et al. [9] introduced an approximate unlearning method, achieving effects akin to retraining with minimal additional training.

Nonetheless, these methods exhibit limitations, chiefly their dependence on supervised learning for machine unlearning. Research on contrastive and self-supervised learning remains relatively limited [5]. In this study, we present a novel gradient penalty-based machine unlearning method, enabling approximate unlearning by modifying the loss during model training. This approach requires minimal training to effectuate machine unlearning for designated data while ensuring that the model can forget specified data without a substantial loss in accuracy (generally within 10%). Our method is simple, efficient, and highly adaptable, demonstrating commendable performance on both supervised and contrastive learning models.

2 Related Work

The premise of machine unlearning [7] is to enable a model to completely forget the influence of specified data, with the main idea being to directly remove the data to be forgotten from the entire model training process. Based on this concept, two directions for machine unlearning have emerged: complete unlearning and approximate unlearning.

Complete unlearning entails retraining the entire model. SISA [2] (Sliceable Incremental and Selective Aggregation) is an early method for machine unlearning. The core idea of SISA is to split the original dataset into multiple independent subsets, which do not share information during training. These independent subsets are then incrementally trained by slicing and partitioning the data. Incremental training implies that the model trains on each data slice and updates the model parameters after each training session, allowing the model to gradually adapt to all sliced datasets. Finally, an ensemble method is used to combine these models. One ensemble approach calculates the output vector for each model, averages these output vectors, and selects the maximum value from the mean vector as the final classification result. The advantage of this method is that it can achieve complete unlearning of the data to be forgotten.

However, its drawbacks are that it requires adopting a specific framework, which may not be compatible with existing models and training frameworks used by most companies, and the loss of model accuracy due to the ensemble method can be substantial, greatly impacting the model's performance.

Due to these drawbacks, approximate unlearning has been considered as an alternative. Approximate unlearning does not require the model to achieve the same effect as with non-trained data; it only needs to be close. Moreover, it can be achieved by training the existing model for a small number of iterations. This approach not only saves a significant amount of training resources but also preserves the model's original performance to the greatest extent. PUMA [9] (Private Update via Model Approximation) is a method for implementing approximate unlearning. Its goal is to remove the influence of training data while maintaining minimal changes in model performance. PUMA achieves this primarily by generating synthetic data and fine-tuning the model using this data. However, one drawback is that generating perturbation data with activation values similar to the forgotten samples in the model can be challenging. This process may require complex optimization methods, such as gradient matching, thereby increasing computational complexity.

The method proposed in this paper is also an approximate unlearning approach. However, the key difference is that our method can be applied to both supervised learning and contrastive learning models, making it a versatile solution for a wider range of applications.

3 Gradient Penalty-Based Unlearning Method

In this section, we will specify the details of our unlearning method, and the source of inspiration for our method.

3.1 Gradient Penalty

Our approach is inspired by WGAN [1], in which the generator needs to compute gradient_penalty as a loss to ensure the stability of the model training. The gradient penalty ensures that the gradient of the discriminator remains appropriate during training by interpolating between the real and generated samples and requiring the gradient of the interpolated points to be close to 1 in magnitude. This gradient penalty term will be added to the loss function of the discriminator to ensure the stability of the training process. By using the gradient penalty, WGAN-GP has better stability in training the game process between the generator and the discriminator, and reduces the risk of gradient disappearance and pattern collapse problems. The following is the flow chart of the algorithm for the penalty term.

Through our investigation, we have discovered that by treating real samples as trained samples and fake samples as non-trained samples, the gradient penalty loss can bring the model's prediction confidence for both trained and non-trained data closer together. This, in turn, renders the trained and non-trained data

indistinguishable. The core component of this function hinges on generating interpolated data between trained and non-member data, followed by computing the respective gradients and calculating the penalty value. Below is the formula for calculating the penalty value.

$$\text{gradient\_penalty} = \frac{1}{N}\sum_{i=1}^{N}\left(\|\nabla_{\mathbf{z}_i} D(\mathbf{z}_i)\|_2 - 1\right)^2 \tag{1}$$

However, this approach presents a notable limitation. While it successfully brings the model's prediction confidence for both member and non-member data closer together, it does so by mutually converging the predictions. In other words, the predictions for non-trained data are also altered, eventually causing the trained and non-trained data to aggregate at a central point between the two. Although this process reduces the probability distribution of the model's output prediction confidence for trained data, it concurrently increases the output prediction confidence for non-member data. This outcome is not desirable; ideally, the model should treat predictions for trained data as if it were untrained, while maintaining the non-trained data predictions unchanged (Figs. 1 and 2).

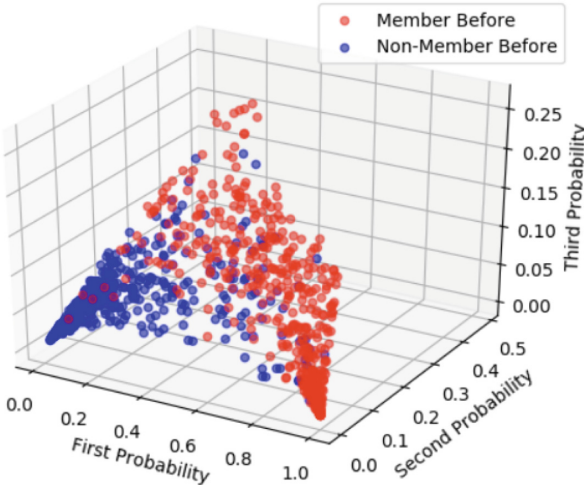

Fig. 1. Before gradient penalty

3.2 Our Objectives

Due to the aforementioned issue, our primary focus in subsequent research is to identify a loss function or a specific technique capable of reducing the model's prediction confidence for data. Although L2 regularization can mitigate the extent of overfitting in the model, it fails to alleviate the disparity between

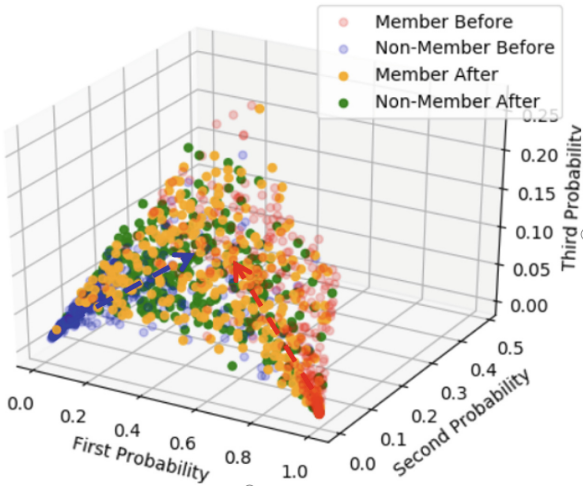

Fig. 2. After gradient penalty

member and non-member data. However, since the penalty term in WGAN can reduce the difference in prediction outputs between trained and non-trained data, we propose a combination of the two approaches. By calculating the average L2 norm of the model's output for member data as a loss component and combining it with the penalty term and the member data training loss, we can create a composite loss function for limited model training. This approach enables the targeted unlearning of specific data.

With this understanding, we establish the foundation for our study on contrastive learning for machine unlearning, focusing primarily on two objectives. First, to enable the model to defend against membership inference attacks, the model's output prediction confidence for member and non-member data should be nearly identical or indistinguishable. Second, to achieve data unlearning, the model's prediction confidence for data should be relatively low, or the uncertainty should be considerably high.

3.3 Our Method

To accomplish these two objectives, we employ the WGAN gradient penalty term as a loss function for machine unlearning training, ensuring that the model's prediction confidence for member and non-member data is nearly indistinguishable. Subsequently, we calculate the L2 norm of the model's encoder prediction output for member data as a loss function for machine unlearning training, aiming to reduce the model's prediction confidence for data. Finally, we incorporate the model's training loss for member data as a constraint term during the machine unlearning training process to maintain the model's accuracy.

Our proposed method is capable of rendering member and non-member data predictions indistinguishable while maintaining a minimal loss in model accuracy

(at most approximately 10%). This approach is applicable to both contrastive learning models and supervised models. For contrastive learning models, we first extract the model's encoder, then perform unlearning based on the encoder's prediction output for the data. Our method requires only a simple modification to the training loss function, and demands a relatively low number of training epochs (approximately 10). The general form of our loss function is as follows:

$$\mathcal{L} = \alpha \cdot L_{\text{MEMtrain}} + \beta \cdot L_{\text{GP}} + \gamma \cdot L_{\text{Norm}} \tag{2}$$

The loss function is composed of three distinct components. The first component corresponds to the training loss of the data to be forgotten within the model. This element primarily serves to prevent model collapse during the unlearning process. In our experiments, omitting this component resulted in significant losses in model accuracy. The second component is the gradient constraint term from WGAN, which, in its original form, stabilizes model training by preventing gradient explosion and vanishing. However, our research has discovered that this function can also facilitate convergence in prediction output confidence for both member and non-member data. The third component involves computing the L2 norm of the model's prediction output, aimed at reducing the prediction confidence. Although the second component can induce convergence in prediction confidence, it tends to elevate the confidence for non-member data predictions. Therefore, we strive to lower the model's prediction confidence for all data.

4 Experiments

4.1 Experimental Settings

DataSets: We conducted experiments on the SVHN, CIFAR10, CIFAR100 data sets. The distribution of the datasets is shown in Table 1

Table 1. Datasets description

Datasets	Shape	Classes	Training data	Testing data
CIFAR-10	$32 \times 32 \times 3$	10	50,000	10,000
CIFAR-100	$32 \times 32 \times 3$	100	50,000	10,000
SVHN	$32 \times 32 \times 3$	10	73,257	26,032

Experimental Details: We implemented the series of attacks described above using PyTorch in Python 3.7. Our computational resources included 4 NVIDIA V100 GPUs. To control variables, we conducted experiments using a combination of 10,000 training data and 10,000 test data samples. During contrastive learning model training, we trained the model for 1,600 epochs to induce overfitting, using the Adam optimizer with a learning rate of 0.01.

Common methods for evaluating encoder performance include linear evaluation and weighted KNN evaluation. Linear evaluation measures the quality of feature representations extracted by the encoder when trained with a linear model. Weighted KNN evaluation involves comparing feature representations' cosine similarity and classifying them using a weighted voting k-nearest neighbors method. During the training process, we monitored performance using weighted KNN evaluation and tested the final performance with linear evaluation.

For the experiments, the baseline amount of data to be forgotten was 2,000 out of 10,000 training data samples, which corresponds to 20% of the data. We conducted additional comparative experiments by varying the number of forgotten data samples. The supervised learning model used a ResNet architecture, while the contrastive learning model employed MoCo. The number of epochs required for unlearning training was 10. We also conducted ablation experiments to demonstrate the importance of loss selection in our method.

Evaluation Metrics: Membership inference attacks primarily evaluate classifiers; therefore, our evaluation metrics include accuracy, precision, recall, and AUC.

1. Accuracy: Accuracy is the proportion of samples that the classification model correctly predicts relative to the total number of samples. It is calculated using the following formula: Accuracy is suitable for balanced classes; however, in imbalanced class situations, it may not accurately reflect model performance.
2. Precision: Precision is the proportion of true positive samples among all samples predicted as positive by the model. It is calculated using the following formula: Precision reflects the reliability of the model when predicting positive classes.
3. Recall: Recall is the proportion of true positive samples that the model correctly predicts as positive among all actual positive samples. It is calculated using the following formula: Recall reflects the extent to which the model covers the detection of positive classes.
4. AUC (Area Under Curve): AUC represents the area under the Receiver Operating Characteristic (ROC) curve. The ROC curve is drawn based on the true positive rate (TPR) and false positive rate (FPR) at different thresholds. AUC values range from 0 to 1, with a perfect classifier having an AUC of 1 and a random classifier having an AUC of approximately 0.5. AUC is a comprehensive performance metric that can reflect the model's classification ability in imbalanced class situations.

4.2 Experimental Results

We first use our machine unlearning method for the self-supervised comparison learning model, and then we use the method of that ENcoderMI paper to perform membership inference attacks on the model, and then record the change in the success rate of the membership attacks before and after performing machine unlearning, as shown in the following Tables 2 and 3.

Table 2. Contrast Machine unlearning performance before Unlearning.

Dataset	Before Unlearning			
	model_acc	mia_acc	mia_rec	mia_pre
cifar10	70%	87%	87%	88%
cifar100	32%	92%	92%	92%
svhn	76%	86%	87%	86%

Table 3. Contrast Machine unlearning performance after Unlearning.

Dataset	After Unlearning			
	model_acc	mia_acc	mia_rec	mia_pre
cifar10	60%	51%	52%	52%
cifar100	25%	51%	51%	51%
svhn	73%	51%	51%	51%

From the above table, we can see that our method is able to be member inference attack completely invalid, while it can keep the accuracy loss of the model not too high to some extent. To further demonstrate the feasibility of our method, we experimented with different models for moco, simclr and byol, and the experimental results are shown in the following Table 4.

Table 4. Performance of different models on CIFAR-10

Model	ACC_bef	ACC_af	MIA_bef	MIA_af
MoCo	70%	60%	90%	50%
SimCLR	66%	60%	70%	50%
BYOL	55%	49%	70%	50%

To investigate whether our method truly achieves unlearning or approximates the effect of unlearning, we will employ three approaches to study the model. The first approach is based on our observation that, although contrastive models do not require labels during the training process, the output probabilities obtained using their encoders to predict data exhibit high prediction confidence. This difference in prediction confidence is one of the core components in supervised membership inference attacks. Thus, it suggests that an overfitted self-supervised contrastive learning model can also be targeted by supervised membership inference attacks. Our implementation confirms this, with the inference success rate being similar to that of the EncoderMI method. However, our primary focus here is on prediction confidence.

The difference in prediction confidence between training and non-training data is mainly manifested in the model's prediction probability distribution for

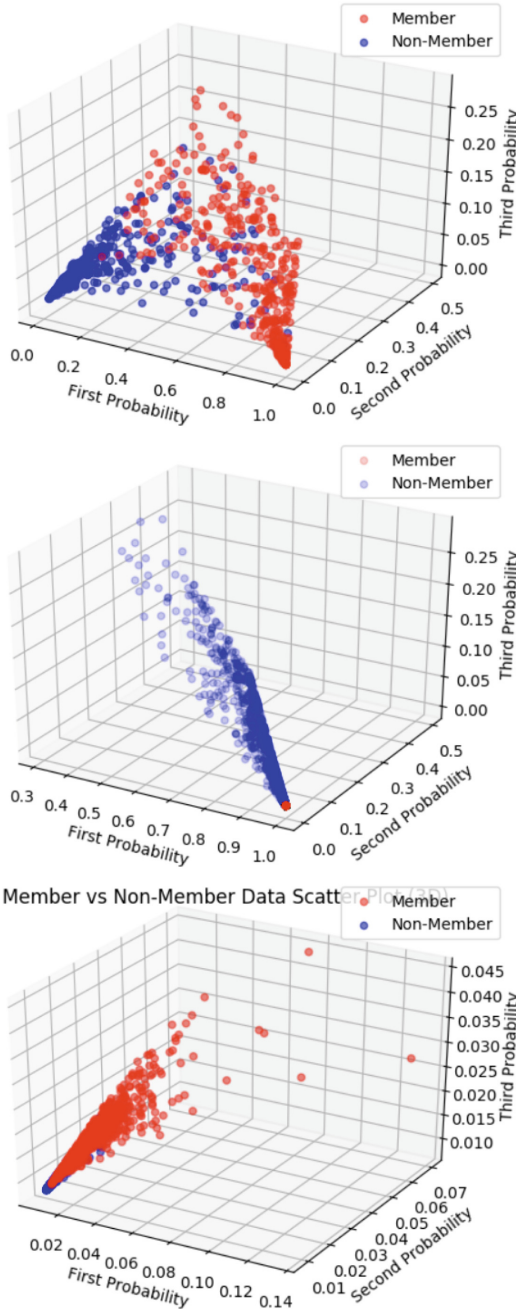

Fig. 3. The distribution of the predicted data probabilities before unlearning.

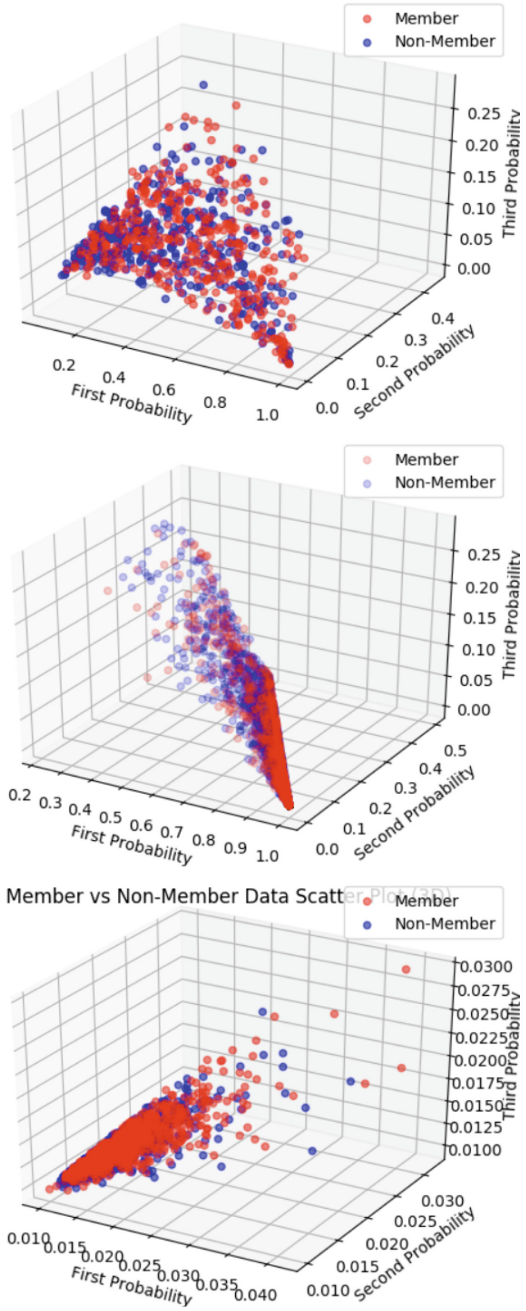

Fig. 4. The distribution of the predicted data probabilities after unlearning.

non-training data, which tends to be relatively flat and uniform. In contrast, the prediction probability distribution for training data exhibits an extremely skewed distribution. The following figure illustrates the distribution of the top-3 prediction confidence values in three-dimensional space for an overfitted contrastive learning model before and after unlearning, with respect to training and non-training data, as shown in Figs. 3 and 4.

There are various forms of overfitting, and as shown in the figure above, all three distributions of overfitting have close success rates of inference attacks on their members, but their distributions of top3 probability values for team-trained and non-trained data are quite different. The explanation we give here is that the model will have a large variation in its prediction ability or prediction confidence as the degree of overfitting changes. We believe that this may be due to a decrease in the generalization ability of the model and the fact that the internal parameters of the model are tuned to fit the training data more easily as the team training data are trained in depth. We found that when a smaller learning rate is used, the overfitting distribution of the model changes in the direction of the most lateral graph above. When we increase the learning rate, the overfitting distribution of the model will be as shown in the middle of the figure above. As we continue to increase the learning rate, the model overfitting distribution becomes like the rightmost position in the upper panel. Using our forgetting method works for all three types of unlearning, i.e., making the distribution of the training data vary as if it were the distribution of the non-training data, making it indistinguishable.

5 Conclusion

We propose a machine unlearning method that supports both contrastive learning models and supervised models, achieving excellent performance levels. Our approach effectively defends against membership inference attacks (MIAs) and protects user privacy. Moreover, it does not require complex preprocessing, nor does it rely on specific frameworks, making it a fairly generalizable method. To implement our method, one simply needs the model and the data to be forgotten, making the approach highly user-friendly. Additionally, our method does not demand extensive computational resources; it can be achieved with just a few training epochs. However, further evaluation and testing, such as examining the model's unlearning effects from various perspectives, remain areas for future research.

Acknowledgments. This work is supported by National Natural Science Foundation of China (No. 61802383), Research Project of Pazhou Lab for Excellent Young Scholars (No. PZL2021KF0024), Guangzhou Basic and Applied Basic Research Foundation (No. 202201010330, No. 202201020162), Guangdong Philosophy and Social Science Planning Project (No. GD19YYJ02), Guangdong Regional Joint Fund Project (No. 2022A1515110157), and Research on the Supporting Technologies of the Metaverse in Cultural Media (No. PT252022039).

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning, pp. 214–223. PMLR (2017)
2. Bourtoule, L., et al.: Machine unlearning. In: 2021 IEEE Symposium on Security and Privacy (SP), pp. 141–159. IEEE (2021)
3. Cao, Y., Yang, M.: Legal regulation of big data killing:-from the perspective of "personal information protection law. J. Educ. Humanit. Soc. Sci. **7**, 233–241 (2023)
4. Chen, K., Huang, Y., Wang, Y., Zhang, X., Mi, B., Wang, Y.: Privacy preserving machine unlearning for smart cities. Ann. Telecommun. **79**(1), 61–72 (2023)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: Simclr: a simple framework for contrastive learning of visual representations. In: Proceedings of the 37th International Conference on Machine Learning, pp. 1597–1607 (2020)
6. Mayfield, M.: Talk data to me: why michigan should adopt a comprehensive data protection statute. Wayne St. UJ Bus. L. **6**, 1 (2023)
7. Nguyen, T.T., Huynh, T.T., Nguyen, P.L., Liew, A.W.C., Yin, H., Nguyen, Q.V.H.: A survey of machine unlearning. arXiv preprint arXiv:2209.02299 (2022)
8. Politou, E., Alepis, E., Patsakis, C.: Forgetting personal data and revoking consent under the GDPR: challenges and proposed solutions. J. Cybersecur. **4**(1), tyy001 (2018)
9. Wu, G., Hashemi, M., Srinivasa, C.: Puma: performance unchanged model augmentation for training data removal. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 8675–8682 (2022)
10. Yan, H., Li, X., Guo, Z., Li, H., Li, F., Lin, X.: Arcane: an efficient architecture for exact machine unlearning. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, pp. 4006–4013 (2022)