Alignment Calibration: Machine Unlearning for Contrastive Learning under Auditing

Yihan Wang [*1 2] **Yiwei Lu** [*3 4] **Guojun Zhang** [5] **Franziska Boenisch** [6] **Adam Dziedzic** [6]
Xiao-Shan Gao [1 2] **Yaoliang Yu** [3 4]

Abstract

Machine unlearning provides viable solutions to revoke the effect of certain training data on pre-trained model parameters. Existing approaches provide unlearning recipes for classification and generative models. However, a category of important machine learning models, i.e., contrastive learning (CL) methods, is overlooked. In this paper, we fill this gap by first proposing the framework of **M**achine **U**nlearning for **C**ontrastive learning (MUC) and adapting existing methods. Furthermore, we observe that several methods are mediocre unlearners and existing auditing tools may not be sufficient for data owners to validate the unlearning effects in contrastive learning. We thus propose a novel method called *Alignment Calibration (AC)* by explicitly considering the properties of contrastive learning and optimizing towards novel auditing metrics to easily verify unlearning. We empirically compare *AC* with baseline methods on SimCLR, MoCo and CLIP. We observe that *AC* addresses drawbacks of existing methods: (1) achieving state-of-the-art performance and approximating exact unlearning (retraining); (2) allowing data owners to clearly visualize the effect caused by unlearning through *black-box auditing*.

1. Introduction

The success of modern machine learning models largely relies on training with a large corpus of data. However, carefully annotated data are expensive and difficult to obtain, thus urging the utilization of the vast amount of unlabelled data in the wild. The recent self-supervised learning methods, especially contrastive learning methods (Chen et al. 2020; Chen et al. 2021; He et al. 2020), provide viable solutions to learning general representations for various downstream tasks. For example, unimodal contrastive learning models employ the InfoNCE loss to maximize the feature similarity between positive pairs (e.g., different data augmentations of the same image) while minimizing that between the negative ones (e.g., different data samples). This training scheme also applies to multi-modal training (e.g., CLIP (Radford et al. 2021)), and the learned encoders are widely applied in various tasks.

To collect large-scale datasets for training contrastive learning models, practitioners usually extract the desired data by crawling on the internet. Such data collection procedures may disregard data owners' privacy concerns and retrieve their data unwillingly. Moreover, part of the acquired training data may be copyrighted or even contain inappropriate content such as sexual abuse (e.g., in recent reports[1] against contents in LAION-5B (Schuhmann et al. 2022)). In such scenarios, data owners or even the authorities are entitled to send a removal request upon such misused training data (i.e., unlearning dataset), which consequently affects the trained model parameters. A straightforward solution is to retrain the model entirely from scratch without the unlearning dataset, but the computational cost can be tremendous for large models and datasets.

To eliminate the effect of the unlearning dataset on the model with minimum effort, machine unlearning methods (Cao and Yang 2015; Bourtoule et al. 2021; Ginart et al. 2019; Guo et al. 2019; Neel et al. 2021; Ullah et al. 2021; Sekhari et al. 2021; Izzo et al. 2021; Chen et al. 2023; Zhang et al. 2024; Fan et al. 2024; Shen et al. 2024) provide recipes for supervised learning methods on group removal and for generative models on sample or concept removal. However, the study of an efficient solution for contrastive learning models is underexplored. In this paper, we establish the foundation of **M**achine **U**nlearning for **C**ontrastive learn-

[*]Equal contribution [1]Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China [2]University of Chinese Academy of Sciences [3]School of Computer Science, University of Waterloo [4]Vector Institute [5]Huawei Noah's Ark Lab [6]CISPA. Correspondence to: Yihan Wang <yihanwang@amss.ac.cn>.

Work presented at the Next Generation of AI Safety Workshop at ICML 2024, Vienna, Austria.

[1]https://purl.stanford.edu/kh752sm9123?ref=404media.co

ing models (MUC). MUC adapts various existing methods to contrastive learning, and introduces the notion of data owners who request unlearning and model owners who execute unlearning. Given candidate unlearning algorithms, the model owners first perform *white-box evaluation* to select the best method and generate the optimal unlearned model. The data owners then perform *black-box auditing* to validate the effect of the unlearning procedure. We argue that unlearning success is achieved only if the unlearned model meets the criteria on both sides.

Unfortunately, direct adaptions of existing unlearning approaches are unsatisfactory on both considerations. Firstly, from the model owners' perspective, such algorithms are mediocre approximations of exact unlearning (training from scratch) under different white-box evaluations and there lack of a good candidate method. Secondly, from the data owner's perspective, even given the strongest unlearned model returned by exact unlearning, it is difficult to discern the unlearning effect under existing *black-box auditing* tools, rendering it hard to determine the success of unlearning.

Motivated by the above state of affairs, we introduce a novel unlearning method called *Alignment Calibration (AC)* that is specifically tailored for contrastive learning. To approximate exact unlearning, our approach takes the properties of the InfoNCE loss in contrastive learning and the goals of unlearning into consideration and introduces an additional positive alignment calibration term during optimization. Moreover, we propose negative alignment calibration to provide extra visual auditing tools for data owners to validate the effect of unlearning.

Finally, we empirically compare baseline methods with our *Alignment Calibration* algorithms on unlearning models pre-trained on SimCLR (Chen et al. 2020) MoCo (He et al. 2020), and CLIP (Radford et al. 2021). Under various unlearning settings (e.g., the fraction of the unlearning dataset) and evaluation metrics, *AC* consistently outperforms the baseline methods, especially under unlearn auditing, validating the benefits of our method. In summary, we make the following contributions:

- We propose MUC that considers existing methods and various evaluation tools in contrastive learning, including *white-box evaluation* and *black-box auditing*.
- Motivated by insufficiencies of existing unlearning algorithms and auditing tools, we propose *Alignment Calibration* for both model owners and data owners.
- Our experiments initiate the evaluation of existing machine unlearning methods for contrastive learning and confirm the superiority of our new methods.

2. Machine Unlearning for Contrastive Learning (MUC)

In this section, we specify the problem setting of machine unlearning for contrastive learning, introduce direct adaptations of existing methods, and propose evaluation metrics.

2.1. Problem Settings

Setting and notation: Suppose a contrastive learning model g with parameter $\mathbf{w}$ is obtained by training on a training set $\mathcal{D}_{\texttt{train}}$. After g is deployed, unlearning requests may be made to remove part of the training set, which we denote as $\mathcal{D}_{\texttt{unlearn}}$. The goal here is to acquire a new set of parameters $\mathbf{w}_u$, which amounts to being trained on a retained dataset $\mathcal{D}_{\texttt{retain}} = \mathcal{D}_{\texttt{train}} \backslash \mathcal{D}_{\texttt{unlearn}}$ ($\backslash$ denotes removal). Aside from the straightforward and expensive "retraining from scratch" recipe, unlearning methods aim to achieve the same goal approximately with improved efficiency. Additionally, we introduce the notion of the model owner who receives the unlearning request, and the data owners who wish to remove data as two parties involved in unlearning. In practical scenarios, consists of a group of individuals $\{{}^i\}_{i=1}^{N}$, who may not know the existence of each other, but participate in unlearning at the same time.

Goal of unlearning: ❶ From the owner 's perspective, there are two major considerations: preserving the model performance on $\mathcal{D}_{\texttt{retain}}$ while eliminating the effects of training on $\mathcal{D}_{\texttt{unlearn}}$; ❷ In contrast, an individual data owner ^{i} only has limited access to his/her own subset $\mathcal{D}^i_{\texttt{unlearn}}$, the output of the encoder before/after unlearning. ^{i} wishes to observe different and desired behaviors of $\mathcal{D}^i_{\texttt{unlearn}}$ on $\mathbf{w}_u$ compared with that of $\mathbf{w}$.

2.2. Evaluating an unlearned model

Suppose the model owner has obtained the unlearned model $\mathbf{w}_u$, next, we discuss the evaluation metrics on assessing its performance.

For the model owner : To explicitly measure whether the effect of $\mathcal{D}^i_{\texttt{unlearn}}$ has been removed by $\mathbf{w}_u$, previous works set retraining from scratch as the baseline for evaluating probabilistic unlearning methods. Note that the comparison with retraining exhibits *white-box evaluation* to choose the optimal unlearning method from a pool of algorithms. Suppose a retrained (and converged) model has parameter $\mathbf{w}_r$, evaluation metrics can be established by comparing various behaviors of $\mathbf{w}_u$ and $\mathbf{w}_r$ with respect to:

- *Encoder-level metrics.* ❶ Forgetting Score: we propose a Forgetting Score ($\texttt{FS}$) by directly adapting the memorization score in evaluating data attribution in Wang et al. 2024 to unlearning. FS measures the quantity of forgetting

$\mathcal{D}_{\texttt{unlearn}}$ by comparing the alignment loss through the features returned by models before and after unlearning:

$$\texttt{FS} \coloneqq \mathop{\mathbb{E}}_{(x,y)\sim p_{\mathrm{u}}^{+}} s(x^{g}, y^{g}) - \mathop{\mathbb{E}}_{(x,y)\sim p_{\mathrm{u}}^{+}} s(x^{\mathbf{g}}, y^{\mathbf{g}}), \quad (1)$$

where p_{u} is the density of $\mathcal{D}_{\texttt{unlearn}}$, g and $\mathbf{g}$ are models before/after unlearning, and s is the cosine similarity.
❷ Membership Inference Attacks: EncoderMI (Liu et al. 2021) proposed an alignment-based membership inference attack for self-supervised encoders. It extracts membership information from the embedded features to distinguish whether input data is included in the encoder training set. Following Jia et al. (2023) and Fan et al. (2024), we evaluate the attack success rate (ASR) on the unlearn dataset $\mathcal{D}_{\texttt{unlearn}}$ and denote it by encoder membership inference attack (EMIA) efficacy.

- *Downstream-level metrics.* To quantify the model performance before/after unlearning, we perform linear probing, *i.e.*, image classification on the same (labeled) dataset for unimodal contrastive learning. Given the encoder g with unlearned parameters $\mathbf{w}_u$, we train an additional linear head on top of the fixed g to obtain a classifier. Next we evaluate: ❶ Accuracies: we evaluate retain accuracy (**RA**) on $\mathcal{D}_{\texttt{retain}}$, test accuracy on $\mathcal{D}_{\texttt{test}}$ (**TA**), and unlearn accuracy on $\mathcal{D}_{\texttt{unlearn}}$ (**UA**). For a good unlearning algorithm, the above three measurements should be close to that of the retrained model $\mathbf{w}_r$, with a common pattern of **UA** $\approx$ **TA** $<$ **RA**; ❷ Membership Inference Attacks: similarly to EMIA, we implement a confidence-based membership inference attack (Jia et al. 2023; Fan et al. 2024; Song and Mittal 2021) on the entire network (encoder and linear head) and report classifier membership inference attack efficacy (CMIA).

For the data owners (Unlearn Auditing) : To validate the unlearning effect from the data owners' perspective, we further propose the notion of ***unlearn auditing***. Note that this process is *black-box auditing* as an individual data owner ^{i} can only observe the input $\mathcal{D}^{i}_{\texttt{unlearn}}$ and the output of the encoder before/after unlearning. To this end, the only auditing tool is the forgetting score $\texttt{FS}$ on $\mathcal{D}_{\texttt{unlearn}}$, which can be calculated with Equation (1). However, we argue this auditing is ***neither sufficient nor reliable*** and we use a simple empirical example to validate this claim:

Exact unlearning on MoCo (He et al. 2020): We perform exact unlearning (i.e., retraining) to forget 4500 training images of CIFAR-10 (randomly chosen) on MoCo (ResNet-18). We calculate the forgetting score $\texttt{FS}$ for every unlearn sample and calculate the mean μ and the standard derivation σ across the 4500 unlearning images. We obtain $\mu = 0.025, \sigma = 0.081$.

Here we observe the large standard derivation $\sigma = 0.081 = 3.24 \times \mu$ makes the current auditing largely unreliable. For individual data owners, if the unlearn subset size $|\mathcal{D}^{i}_{\texttt{unlearn}}|$ is small, its corresponding sample-wise $\texttt{FS}$ is likely to be biased and the average could fluctuate around 0 and suggesting little forgetting. This could lead to the belief in "haven't performed unlearning" from the data owner's side, thus rejecting the exact unlearned model!

This example reveals the insufficiency of both the existing unlearning methods (even exact unlearning) and unlearn auditing tools in contrastive learning.

3. Alignment Calibration

In Section 2, we introduce the notion of MUC, adapt existing unlearning methods, and propose evaluation metrics. However, we observe insufficiencies in the current approaches in two aspects: model owners *do not possess an effective unlearner*; data owners *lack reliable auditing tools.* In this section, we propose a novel method called ***Alignment Calibration (AC)***, providing a powerful unlearner for contrastive learning and a practical visual auditing mechanism.

3.1. Tailored objective for MUC

We first introduce a more effective unlearner for model owners . Recall the 's goal of unlearning: preserving the model utility on $\mathcal{D}_{\texttt{retain}}$ while revoking the effects of training on $\mathcal{D}_{\texttt{unlearn}}$. For the retain dataset $\mathcal{D}_{\texttt{retain}}$, we minimize the InfoNCE loss in Equation (5) to achieve reasonable downstream performance after unlearning:

$$-\mathop{\mathbb{E}}_{(x,y)\sim p_{\mathrm{r}}^{+}} s(x^{g}, y^{g}) + \mathop{\mathbb{E}}_{x\sim p_{\mathrm{r}}} \log \mathop{\mathbb{E}}_{y\sim p_{\mathrm{d}}} \exp(s(x^{g}, y^{g})), \quad (2)$$

where p_{r} is the density of $\mathcal{D}_{\texttt{retain}}$ and p_{d} is the density of $\mathcal{D}_{\texttt{train}}$. For the unlearn dataset $\mathcal{D}_{\texttt{unlearn}}$, revoking the effects of training amounts to achieving the following goals upon evaluation in Section 2.2:

- (*Encoder-level*) Enlarging forgetting on $\mathcal{D}_{\texttt{unlearn}}$: recall that in Equation (1) the forgetting score $\texttt{FS}$ is measured by the difference between feature similarity on $\mathcal{D}_{\texttt{unlearn}}$ before/after unlearning with pre-trained model g and unlearned model $\mathbf{g}$. As the first term is fixed during unlearning, increasing $\texttt{FS}$ is equal to minimizing the second positive alignment term. For this purpose, we explicitly perform such minimization in our objective function and call it *positive alignment calibration*.

- (*Downstream-level*) **UA** $\approx$ **TA** $<$ **RA**: to obtain reasonable UA and TA, we find it beneficial to maintain the term for negative pairs in contrastive learning, such that for

$\mathcal{D}_{\texttt{unlearn}}$, we minimize:

$$\underbrace{\mathop{\mathbb{E}}_{(x,y)\sim p_{\mathrm{u}}^{+}} s(x^g, y^g)}_{\textit{positive alignment calibration}} + \underbrace{\mathop{\mathbb{E}}_{x\sim p_{\mathrm{u}}} \log \mathop{\mathbb{E}}_{y\sim p_{\mathrm{d}}} \exp(s(x^g, y^g))}_{\textit{performance preserving}}, \tag{3}$$

where p_{u} is the density of $\mathcal{D}_{\texttt{unlearn}}$.

3.2. Calibration under unlearn auditing

Auditing beyond FS: Recall that in Section 2.2, we show that the forgetting score FS is not a sufficient nor reliable evaluation for unlearning success. Here we introduce an additional auditing tool: given $\mathcal{D}_{\texttt{unlearn}}$ and the models before unlearning g, data owners can easily obtain the feature vectors with two different data augmentations: $\mathbf{x}^g = \{x_i^g\}_{i=1}^{|\mathcal{D}_{\texttt{unlearn}}|}$ and $\mathbf{y}^g = \{y_j^g\}_{j=1}^{|\mathcal{D}_{\texttt{unlearn}}|}$. Then an Alignment Matrix : $\mathrm{AM}(\mathbf{x}^g, \mathbf{y}^g)$ can be easily acquired by calculating the pairwise similarity between the two vectors. See Figure 2(a) for some visualizations of AM in the format of heatmaps. Similarly, can obtain $\mathrm{AM}(\mathbf{x^g}, \mathbf{y^g})$ after unlearning and an additional Alignment Gap Matrix : $\mathrm{AGM} = \mathrm{AM}(\mathbf{x}^g, \mathbf{y}^g) - \mathrm{AM}(\mathbf{x^g}, \mathbf{y^g})$. The heatmaps of AM and AGM provide auditing tools beyond FS and allow to visualize the model change through unlearning by looking at the temperature of the graphs. Notably, the elements on the diagonal of AGM also visualize sample-wise forgetting scores.

Taking auditing into account for unlearning: The additional auditing tools enable the model owners to design an algorithm that allows data owners to clearly visualize the effect caused by unlearning (*i.e.*, through AM or AGM) without sacrificing the goal of unlearning.

We provide a simple solution to improve existing unlearning methods. As we have explicitly calibrated the alignment of positive pairs of $\mathcal{L}_{\texttt{unlearn}}$ in Equation (3), it suffices to adjust that of negative pairs (within $\mathcal{D}_{\texttt{unlearn}}$) to a larger value to enlarge the model differences in AM. Specifically, we update the unlearn loss in Equation (3) with *negative alignment calibration*:

$$\mathcal{L} \coloneqq -\alpha \cdot \underbrace{\mathop{\mathbb{E}}_{(x,y)\sim p_{\mathrm{u}}^{\times}} s(x^g, y^g)}_{\textit{negative alignment calibration}} + \beta \cdot \underbrace{\mathop{\mathbb{E}}_{(x,y)\sim p_{\mathrm{u}}^{+}} s(x^g, y^g)}_{\textit{positive alignment calibration}} + \gamma \cdot \underbrace{\mathop{\mathbb{E}}_{x\sim p_{\mathrm{u}}} \log \mathop{\mathbb{E}}_{y\sim p_{\mathrm{d}}} \exp(s(x^g, y^g))}_{\textit{performance preserving}}, \tag{4}$$

where α, β, γ are tunable parameters to adjust the strength of each component. In the next section, we will show *AC* not only achieves state-of-the-art performance upon model owners' evaluations but can also easily pass data owners' visual auditing on unlearning.

4. Experiment

In this section, we evaluate baseline methods and *AC* and present *white-box evaluation* by model owners and *black-box auditing* by data owners .

4.1. Experimental Setup

Data and models. For unimodal contrastive unlearning, we perform experiments on CIFAR-10 and SimCLR/MoCo algorithms with the ResNet-18 backbone. We randomly forget 10/50% training data from a pre-trained encoder. For multimodal contrastive unlearning, we evaluate CLIP (Radford et al. 2021) on an Image-Text paired dataset called MS-COCO (Lin et al. 2014), which contains ∼120K images and ∼600K captions. We perform unlearning on 10% randomly selected image-text pairs.

White-box Evaluation: Following Section 2.2, we use FS and EMIA for encoder-level evaluation, and use CMIA, RA, TA, and UA for downstream-level evaluation after performing linear probing for SimCLR/MoCo experiments. For the evaluation of CLIP, we measure the image-text cosine similarity of the retain dataset and unlearn dataset due to the lack of suitable downstream tasks. Across all experiments, we compare each unlearning method with the exact unlearning (retraining) baseline and report the differences across all metrics. We also report the running time efficacy (RTE) of unlearning methods to evaluate efficiency.

Black-box Auditing: For data owners, calculating FS on their own unlearning subset is also possible as basic auditing. Moreover, we use the Alignment Matrix (AM) and Alignment Gap Matrix (AGM) introduced in Section 3 for visual auditing on MoCo and CLIP.

Unlearning Algorithms: We evaluate Retrain, Fine-Tune, Gradient Ascent, NegGrad, and l_1-Sparsity as baselines for MUC. Our *Alignment Calibration* method updates the pre-trained encoder for the same number of epochs as FineTune, NegGrad, and l_1-Sparsity. For simplicity, we set $\alpha = \gamma = 1$ if not otherwise stated and we tune β for the best performance. Implementation details of the above methods are described in Appendix B.2.

4.2. White-box Evaluation

We first provide empirical evidence for model owners on choosing a suitable unlearning method with superb efficiency and effectiveness. ❶ Unimodal contrastive learning: we present our evaluation under EMIA, RA, TA, UA and CMIA in Table 1 for unlearning 10% of CIFAR-10 training set and FS (CIFAR-10, MoCo) for 10/50% separately in Table 2 due to different scales. For both SimCLR and MoCo, our proposed *Alignment Calibration (AC)* method

Table 1: Unlearning performance of different methods on randomly forgetting 10% of CIFAR-10 training data (MoCo) under various metrics. The performance gaps between retraining and other methods are shown in the parenthesis. We report the average gap (Avg. Gap) over these 5 metrics. The results are obtained by averaging over 5 random trials.

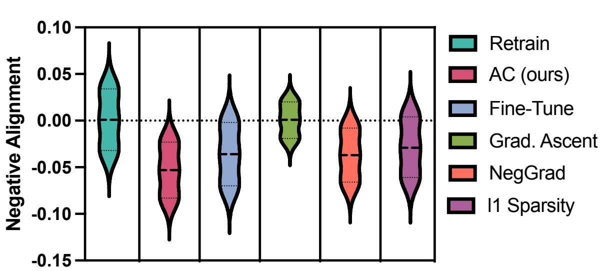
Methods	EMIA	RA	TA	UA	CMIA	Avg. Gap ↓ (%)	RTE ↓ (mins)
MoCo							
Retrain	49.72	89.54	87.76	88.42	34.38	-	109.47
Fine-Tune	50.15 (0.43)	88.34 (1.20)	86.46 (1.30)	87.59 (0.83)	29.42 (4.96)	1.74	1.42
Grad. Ascent	44.95 (4.77)	89.92 (0.38)	88.28 (0.52)	89.76 (1.34)	28.53 (5.85)	2.57	0.17
NegGrad	48.43 (1.29)	89.25 (0.29)	87.35 (0.40)	88.58 (0.16)	28.89 (5.49)	1.53	1.70
l_1-Sparsity	49.38 (0.34)	88.56 (0.98)	86.91 (0.84)	88.12 (0.30)	29.91 (4.47)	1.39	1.43
AC (Ours)	50.28 (0.56)	89.14 (0.40)	87.24 (0.52)	88.20 (0.22)	31.50 (2.88)	**0.92**	1.87

achieves the lowest average performance gap over EMIA, RA, TA, UA, and CMIA. In terms of unlearning efficiency, our methods only introduce a slight overhead. Additionally, our method achieves the lowest `FS` gap compared to retraining. In Appendix C, we also report the results on CIFAR-100 and 50% unlearning, in which our methods consistently achieve the best performance. ❷ Multi-modal contrastive learning: in Table 7 in Appendix C, we again observe that our method is the best approximator of exact unlearning by evaluating the image-text cosine similarity.

Table 2: Forgetting score (`FS`) of methods for CIFAR-10 and MoCo. `FS` gaps are computed between Retrain and other methods.

Methods	10%		50%	
	`FS`	**Gap**	`FS`	**Gap**
Retrain	0.0266	-	0.0604	-
Fine-Tune	0.0393	0.0127	0.0423	0.0180
Grad.Ascent	0.0005	0.0262	0.0007	0.0596
NegGrad	0.0205	0.0061	0.1002	0.0398
l_1-Sparsity	0.0216	0.0050	0.0408	0.0195
Ours	0.0259	**0.0007**	0.0672	**0.0068**

Figure 1: Negative alignment of 4500 unlearn samples (10%) and MoCo and CIFAR-10. The error bar is the standard deviation.



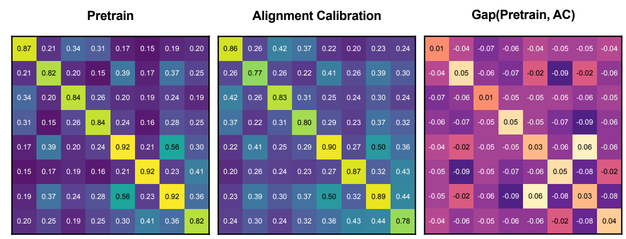


Figure 2: `Alignment Matrices` and `Alignment Gap Matrix` on 8 random images in unlearn dataset of CIFAR-10 (MoCo). The task is forgetting 10% of training data.

4.3. Black-box Auditing

Motivated by the insufficiency of auditing with the `FS` score (positive alignment), we propose to apply the `Alignment Matrix (AM)` and `Alignment Gap Matrix (AGM)` in Section 3. `AM` and `AGM` naturally introduce additional quantification of negative alignment. In Figure 1, we report the negative alignment value (mean and stand deviation of pairwise similarity on negative samples in `AGM`) of 4500 unlearn samples and observe our method *AC* exhibits a more significant unlearning effect under such auditing. For individual data owners 👤^{i}, the size of his/her subset $|\mathcal{D}^i_{\texttt{unlearn}}|$ may be small. Thus we provide additional qualitative results for visual auditing: we randomly select 8 samples from $\mathcal{D}_{\texttt{unlearn}}$ to simulate the budget of 👤^{i}. We construct the `AM` (before/after unlearning with *AC*) and `AGM` for this small set and plot their heatmaps in Figure 2 and observe the apparent effect of unlearning. We provide additional `AGM` for other methods and CLIP unlearning in Figure 3 and Figure 4 in Appendix C.2, where *AC* consistently exhibits the best performance under visual auditing.

5. Conclusion

In this paper, we study the problem of machine unlearning for contrastive learning pre-training (MUC). We establish the foundations on this line of study by adapting existing unlearning methods and setting up baseline evaluation metrics, including *white-box evaluation* for model owners to choose the optimal unlearning strategy, and *black-box auditing* for data owners to examine the effect of unlearning. Spotting the suboptimality of existing unlearning methods and the insufficiency of current auditing tools, we further propose our novel method called *Alignment Calibration*. Our approach introduces a novel unlearning objective function to strategically optimize towards the unlearning goal and enable straightforward visual auditing. Empirically, our method achieves state-of-the-art performance on unlearning tasks for both unimodal and multimodal contrastive learning.

References

Bourtoule, L., V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot (2021). “Machine unlearning”. In: *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, pp. 141–159.

Cao, Y. and J. Yang (2015). “Towards making systems forget with machine unlearning”. In: *2015 IEEE symposium on security and privacy*. IEEE, pp. 463–480.

Caron, M., H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin (2021). “Emerging properties in self-supervised vision transformers”. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660.

Chen, M., W. Gao, G. Liu, K. Peng, and C. Wang (2023). “Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7766–7775.

Chen, T., S. Kornblith, M. Norouzi, and G. Hinton (2020). “A Simple Framework for Contrastive Learning of Visual Representations”. In: *ICML*.

Chen, X. and K. He (2021). “Exploring simple siamese representation learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758.

Chen, X., S. Xie, and K. He (2021). “An empirical study of training self-supervised vision transformers”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9640–9649.

Eldan, R. and M. Russinovich (2023). “Who’s Harry Potter? Approximate Unlearning in LLMs”. *arXiv preprint arXiv:2310.02238*.

Fan, C., J. Liu, Y. Zhang, E. Wong, D. Wei, and S. Liu (2024). “SalUn: Empowering Machine Unlearning via Gradient-based Weight Saliency in Both Image Classification and Generation”. In: *The Twelfth International Conference on Learning Representations*.

Gandikota, R., J. Materzynska, J. Fiotto-Kaufman, and D. Bau (2023). “Erasing concepts from diffusion models”. *arXiv preprint arXiv:2303.07345*.

Ginart, A., M. Guan, G. Valiant, and J. Y. Zou (2019). “Making ai forget you: Data deletion in machine learning”. *Advances in neural information processing systems*, vol. 32.

Grill, J.-B. et al. (2020). “Bootstrap your own latent-a new approach to self-supervised learning”. *Advances in Neural Information Processing Systems*, vol. 33, pp. 21271–21284.

Guo, C., T. Goldstein, A. Hannun, and L. Van Der Maaten (2019). “Certified data removal from machine learning models”. *arXiv preprint arXiv:1911.03030*.

He, K., X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick (2022). “Masked autoencoders are scalable vision learners”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009.

He, K., H. Fan, Y. Wu, S. Xie, and R. Girshick (2020). “Momentum contrast for unsupervised visual representation learning”. In: *CVPR*.

Heng, A. and H. Soh (2023). “Selective Amnesia: A Continual Learning Approach to Forgetting in Deep Generative Models”. arXiv: `2305.10120 [cs.LG]`.

Hu, X., D. Li, B. Hu, Z. Zheng, Z. Liu, and M. Zhang (2024). “Separate the wheat from the chaff: Model deficiency unlearning via parameter-efficient module operation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 18252–18260.

Izzo, Z., M. A. Smart, K. Chaudhuri, and J. Zou (2021). “Approximate data deletion from machine learning models”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 2008–2016.

Jang, J., D. Yoon, S. Yang, S. Cha, M. Lee, L. Logeswaran, and M. Seo (2022). “Knowledge unlearning for mitigating privacy risks in language models”. *arXiv preprint arXiv:2210.01504*.

Jia, J., J. Liu, P. Ram, Y. Yao, G. Liu, Y. Liu, P. Sharma, and S. Liu (2023). “Model Sparsity Can Simplify Machine Unlearning”. In: *Thirty-seventh Conference on Neural Information Processing Systems*.

Kumari, N., B. Zhang, S.-Y. Wang, E. Shechtman, R. Zhang, and J.-Y. Zhu (2023). "Ablating Concepts in Text-to-Image Diffusion Models". arXiv: `2303.13516 [cs.CV]`.

Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014). "Microsoft coco: Common objects in context". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, pp. 740–755.

Liu, H., J. Jia, W. Qu, and N. Z. Gong (2021). "EncoderMI: Membership inference against pre-trained encoders in contrastive learning". In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2081–2095.

Lu, Y., G. Zhang, S. Sun, H. Guo, and Y. Yu (2023). "f-MICL: Understanding and Generalizing InfoNCE-based Contrastive Learning". *Transactions on Machine Learning Research*.

Mitchell, E., C. Lin, A. Bosselut, C. Finn, and C. D. Manning (2022a). "Fast Model Editing at Scale". In: *International Conference on Learning Representations*.

Mitchell, E., C. Lin, A. Bosselut, C. D. Manning, and C. Finn (2022b). "Memory-based model editing at scale". In: *International Conference on Machine Learning*. PMLR, pp. 15817–15831.

Neel, S., A. Roth, and S. Sharifi-Malvajerdi (2021). "Descent-to-delete: Gradient-based methods for machine unlearning". In: *Algorithmic Learning Theory*. PMLR, pp. 931–962.

Oord, A. van den, Y. Li, and O. Vinyals (2018). "Representation learning with contrastive predictive coding". *arXiv preprint arXiv:1807.03748*.

Radford, A. et al. (2021). "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PMLR, pp. 8748–8763.

Saunshi, N., O. Plevrakis, S. Arora, M. Khodak, and H. Khandeparkar (2019). "A Theoretical Analysis of Contrastive Unsupervised Representation Learning". In: *ICML*.

Schuhmann, C. et al. (2022). "Laion-5b: An open large-scale dataset for training next generation image-text models". *Advances in Neural Information Processing Systems*, vol. 35, pp. 25278–25294.

Sekhari, A., J. Acharya, G. Kamath, and A. T. Suresh (2021). "Remember what you want to forget: Algorithms for machine unlearning". *Advances in Neural Information Processing Systems*, vol. 34, pp. 18075–18086.

Shen, S., C. Zhang, Y. Zhao, A. Bialkowski, W. T. Chen, and M. Xu (2024). "Label-Agnostic Forgetting: A Supervision-Free Unlearning in Deep Models". In: *The Twelfth International Conference on Learning Representations*.

Song, L. and P. Mittal (2021). "Systematic evaluation of privacy risks of machine learning models". In: *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2615–2632.

Tosh, C., A. Krishnamurthy, and D. Hsu (2021). "Contrastive learning, multi-view redundancy, and linear models". In: *Algorithmic Learning Theory*. PMLR.

Tsai, Y.-H. H., M. Q. Ma, M. Yang, H. Zhao, L.-P. Morency, and R. Salakhutdinov (2021). "Self-supervised Representation Learning with Relative Predictive Coding". In: *ICLR*.

Ullah, E., T. Mai, A. Rao, R. A. Rossi, and R. Arora (2021). "Machine unlearning via algorithmic stability". In: *Conference on Learning Theory*. PMLR, pp. 4126–4142.

Wang, T. and P. Isola (2020). "Understanding contrastive representation learning through alignment and uniformity on the hypersphere". In: *ICML*. PMLR.

Wang, W., M. A. Kaleem, A. Dziedzic, M. Backes, N. Papernot, and F. Boenisch (2024). "Memorization in Self-Supervised Learning Improves Downstream Generalization". In: *The Twelfth International Conference on Learning Representations*.

Yao, Y., P. Wang, B. Tian, S. Cheng, Z. Li, S. Deng, H. Chen, and N. Zhang (2023). "Editing Large Language Models: Problems, Methods, and Opportunities". In: *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Zhang, D., P. Finckenberg-Broman, T. Hoang, S. Pan, Z. Xing, M. Staples, and X. Xu (2023a). "Right to be forgotten in the era of large language models: Implications, challenges, and solutions". *arXiv preprint arXiv:2307.03941*.

Zhang, E., K. Wang, X. Xu, Z. Wang, and H. Shi (2023b). "Forget-me-not: Learning to forget in text-to-image diffusion models". *arXiv preprint arXiv:2303.17591*.

Zhang, Q., C. Yang, J. Lou, L. Xiong, et al. (2024). "Contrastive Unlearning: A Contrastive Approach to Machine Unlearning". arXiv preprint arXiv:2401.10458.

A. Background and Related Work

We first provide background and related work on contrastive learning and machine unlearning.

Contrastive Learning and Self-supervised Learning Contrastive learning learns general representations by contrasting sample pairs (usually without labels), which analytically benefits the downstream applications (Saunshi et al. 2019; Tosh et al. 2021). Popular contrastive learning methods such as Contrastive Predictive Coding (CPC) (Oord et al. 2018), SimCLR (Chen et al. 2020), and MoCo (He et al. 2020) employ the InfoNCE loss to enforce the contrast between positive and negative pairs. Other variants of the InfoNCE-based loss are also widely applied, e.g., f-MICL (Lu et al. 2023), Alignment and Uniformity (Wang and Isola 2020), and Pearson χ^2 divergence Tsai et al. 2021. This contrastive training diagram is also applied to the context of multimodal learning, where images and texts are formed as pairs, e.g., in CLIP (Radford et al. 2021). There exist other self-supervised learning methods that also learn representations Grill et al. 2020; Chen and He 2021; He et al. 2022; Caron et al. 2021. In this paper, we mainly focus on developing unlearning recipes for contrastive learning methods, especially SimCLR, MoCo, and CLIP.

Specifically, contrastive learning usually applies the InfoNCE loss to learn a representation g. Given a probability measure p, we define the density of *positive pairs* sampled from p as p^+, *i.e.*, two samples with similar feature embeddings joint distribution; and the density of *negative pairs* as $p^\times$. Specifically, one minimizes the loss below as the objective:

$$\mathcal{L}_{\texttt{InfoNCE}} = -\mathop{\mathbb{E}}_{(x,y)\sim p^+} s(x^g, y^g) + \mathop{\mathbb{E}}_{x\sim p} \log \mathop{\mathbb{E}}_{y\sim p} \exp(s(x^g, y^g)), \tag{5}$$

where s is the cosine similarity after normalization with a temperature parameter, and x^g, y^g are the features extracted by a given encoder g, respectively. The above contrastive learning (pre-training) scheme learns a general encoder g (image and text encoders for CLIP). Such a (fixed) g can be utilized with an additional linear head or shallow models for downstream tasks. In this paper, we mainly consider linear probing, where g is used for the classification of the same dataset with pretraining. Notably, we consider unlearning during the pretraining phase only.

Machine Unlearning *For Supervised Learning*: Machine unlearning (MU) Cao and Yang 2015 requires an algorithm to revert to the state that specific data points are never trained on. While exact unlearning (e.g., retraining the model entirely on the retaining dataset) provides a reliable solution, the additional computation requirement is also tremendous. (Bourtoule et al. 2021) propose a sharding-based algorithm in a weaker unlearning model, while other methods explore a probabilistic notion of unlearning (Ginart et al. 2019; Guo et al. 2019; Neel et al. 2021; Ullah et al. 2021; Sekhari et al. 2021; Izzo et al. 2021; Chen et al. 2023; Zhang et al. 2024; Fan et al. 2024; Shen et al. 2024).

For Generative Models: MU methods are applied to diffusion models to avoid copyright infringement and inappropriate image generation (Gandikota et al. 2023; Zhang et al. 2023b; Heng and Soh 2023; Kumari et al. 2023). For large language models, MU is applied as a model-editing Yao et al. 2023 tool to enable forgetting on certain training texts Mitchell et al. 2022b; Mitchell et al. 2022a; Jang et al. 2022; Eldan and Russinovich 2023; Zhang et al. 2023a; Hu et al. 2024. In this paper, we focus on MU on self-supervised learning, specifically, contrastive learning methods, which differs from the above two cases in both unlearning settings and frameworks, which we specify in the following section.

B. Experiment Details

B.1. Datasets

CIFAR-10/100. Both datasets consist of 50K training images and 10K test images. All the images are 32x32 colored. CIFAR-100 has 100 categories and CIFAR-10 has 10 categories. In unimodal contrastive learning, the augmentations for training encoders include random resizing and cropping, random grayscale, random color jitter, and horizontal flipping. We split the 50K training images into a validation set of 5K images and a training set of 45K images. For example, when the unlearning task is to forget 10% of training data, the unlearn dataset $\mathcal{D}_{\texttt{unlearn}}$ has 4.5K images and the retain dataset $\mathcal{D}_{\texttt{retain}}$ has 4.05K images.

MS-COCO. COCO is a large-scale object detection, segmentation, and captioning dataset. Its training set contains 118,287 images and 591,753 captions. Each image has several objects and corresponds to at least 5 captions. Different from unimodal contrastive learning which uses strong augmentations, CLIP employs only resizing, center cropping and horizontal flipping to make images of 224x224 pixels.

B.2. Methods

MoCo and SimCLR. For the Pre-Trained models, we train the encoder for 800 epochs using an SGD optimizer with cosine-scheduled learning rate initialized at 0.06, momentum of 0.9, and weight decay of 0.0005.

The Retrain method uses the same training strategy as Pre-Training. For the Fine-Tune and NegGrad methods, we update the pre-trained encoder for 10 epochs with a learning rate searched in [0.003, 0.03]. For the Gradient-Ascent method, we update the pre-trained encoder using reversed stochastic gradient descent for 5 epochs with a learning rate searched in [10^{-6}, 10^{-4}]. For the l_1-Sparsity method, we set the learning rate as 0.006 and implement l_1 regularization with a coefficient searched in [10^{-6}, 10^{-3}].

For our *Alignment Calibration* method, we update the pre-trained encoder for 10 epochs and search the learning rate in [0.003, 0.03] and the tunable parameter β in [0, 20] for different unlearning tasks. If not otherwise stated, we adopt $\alpha = \gamma = 1$. For simplicity, in our reported results on CIFAR-10/100, we use a learning rate of 0.006 for 10% forgetting, and 0.02 for 50% forgetting.

The linear probing stage trains a linear classifier head for 100 epochs using an SGD optimizer with a cosine-scheduled learning rate initialized at 1.0, and a momentum of 0.9. The batch size for both encoder and linear head training is 512.

CLIP. For the Pre-Trained CLIP, we train the model for 35 epochs on 2 NVIDIA RTX 4090 GPUs using an AdamW optimizer with a warm-up cosine-scheduled learning rate initialized at 5e-4 and momentum of 0.9. The total batch size is 256 (128 on each GPU).

The Retrain method uses the same training strategy as Pre-Training. For the Fine-Tune method, we update the pre-trained model for 8 epochs with a fixed learning rate searched in [5e-5,5e-4]. For the NegGrad method, we update the pre-trained model for 8 epochs with a fixed learning rate searched in [10^{-5}, 10^{-4}]. For the Gradient Ascent method, we update the pre-trained model for 4 epochs with a fixed learning rate searched in [5e-6, 5e-4]. For the l_1-Sparsity method, we update the pre-trained model for 8 epochs with a learning rate of 0.0005 and a regularization coefficient searched in [10^{-9}, 10^{-4}].

For our *Alignment Calibration* method, we update the pre-trained model for 8 epochs with a fixed learning rate of 0.0002. We search $\alpha = \gamma$ in [0.5, 1] and β in [0, 1].

B.3. Evaluation

CMIA efficacy. Given an unlearned encoder **g**, we execute linear probing on it and denote the whole classifier by f. Following the implementation of Jia et al. (2023) and Fan et al. (2024), we evaluate the attack successful rate (ASR) on the unlearn dataset $\mathcal{D}_{\texttt{unlearn}}$ of a confidence-based membership inference attack Song and Mittal 2021 f. The formal definition of CMIA efficacy is given by:

$$\texttt{CMIA-Efficacy} \coloneqq \frac{TN_{\texttt{CMIA}}}{|\mathcal{D}_{\texttt{unlearn}}|}, \tag{6}$$

where $TN_{\texttt{CMIA}}$ is the number of true negatives predicted by the CMIA attack.

EMIA efficacy. We implement the alignment-based EncoderMI-T attack Liu et al. 2021 in an adapted white-box setting. Given an unlearned encoder g with its retain dataset $\mathcal{D}_{\texttt{retain}}$ and test dataset $\mathcal{D}_{\texttt{test}}$, we denote $\mathcal{D}_{\texttt{non-member}} \coloneqq \mathcal{D}_{\texttt{test}}$ sample a subset $\mathcal{D}_{\texttt{member}}$ of $\mathcal{D}_{\texttt{retain}}$ such that $|\mathcal{D}_{\texttt{non-member}}| = \mathcal{D}_{\texttt{member}}$. For each data in $\mathcal{D}_{\texttt{non-member}}$ and $\mathcal{D}_{\texttt{member}}$, we first augment it 10 times and compute features of these 10 views via g. Then compute the cosine similarity between each pair of features, *i.e.*45 pairs, and then take the average of these similarity values. Now we get a membership feature dataset and a non-membership feature dataset whose data points are just scalar values. The EncoderMI-T attack then searches for an optimal threshold to classify membership features and non-membership features. Similar to MIA efficacy, the formal definition of EMIA efficacy is given by:

$$\texttt{EMIA-Efficacy} \coloneqq \frac{TN_{\texttt{EMIA}}}{|\mathcal{D}_{\texttt{unlearn}}|}, \tag{7}$$

where $TN_{\texttt{EMIA}}$ is the number of true negatives predicted by the EncoderMI attack.

C. Additional Experiments

C.1. Unlearning Performance for More Tasks

We present more experiment results on CIFAR-10/100 in Tables 3 to 6. Across these different tasks, our proposed *Alignment Calibration* method achieves the lowest average gap compared to the Retrain method. In Table 7, we report the image-text similarity of unlearn data on a CLIP.

Table 3: Unlearning performance of various methods on randomly forgetting 10% of CIFAR-10 training data (SimCLR). The results are averaged over 5 random trials.

Methods	**EMIA**	**RA**	**TA**	**UA**	**CMIA**	**Avg. Gap** ↓	**RTE** ↓
			SIMCLR				
Retrain	48.11	90.87	88.94	89.68	38.87	-	151.77
Fine-Tune	47.72 (0.39)	89.38 (1.49)	87.26 (1.68)	88.93 (0.75)	30.71 (8.16)	2.49	1.93
Grad. Ascent	41.48 (6.63)	91.26 (0.40)	89.55 (0.61)	91.11 (1.43)	29.36 (9.50)	3.71	0.19
NegGrad	49.56 (1.46)	89.10 (1.77)	87.23 (1.71)	89.07 (0.61)	29.97 (8.89)	2.89	2.34
l_1-Sparsity	48.44 (0.33)	90.59 (0.28)	88.56 (0.38)	90.44 (0.75)	30.61 (8.26)	2.00	1.96
Ours	48.64 (0.53)	90.24 (0.63)	88.06 (0.88)	89.24 (0.44)	33.12 (5.75)	**1.65**	3.00

Table 4: Unlearning performance of various methods on randomly forgetting 50% of CIFAR-10 training data. The results are averaged over 5 random trials.

Methods	**EMIA**	**RA**	**TA**	**UA**	**CMIA**	**Avg. Gap** ↓	**RTE** ↓
			MOCO				
Retrain	55.95	85.98	83.55	83.98	46.66	-	66.71
Fine-Tune	47.72 (5.97)	89.38 (1.90)	87.26 (2.12)	88.93 (2.67)	30.71 (14.31)	5.39	0.86
Grad. Ascent	41.48 (13.06)	91.26 (3.53)	89.55 (4.25)	91.11 (5.01)	29.36 (14.82)	8.13	0.45
NegGrad	49.56 (1.45)	89.10 (2.94)	87.23 (3.40)	89.07 (3.39)	29.97 (6.02)	3.44	1.66
l_1-Sparsity	48.44 (3.76)	90.59 (4.38)	88.56 (3.36)	90.44 (3.20)	30.61 (8.24)	4.59	0.87
Ours	48.64 (0.93)	90.24 (0.30)	88.06 (0.26)	89.24 (0.26)	33.12 (8.27)	**2.00**	1.84
			SIMCLR				
Retrain	53.37	87.23	85.16	85.69	49.30	-	89.74
Fine-Tune	45.90 (7.47)	87.88 (0.65)	85.5 (0.34)	87.23 (1.54)	35.44 (13.85)	4.77	1.17
Grad. Ascent	42.23 (11.13)	90.52 (3.28)	88.61 (3.45)	90.45 (4.77)	33.28 (16.02)	7.73	0.59
NegGrad	55.70 (2.33)	83.98 (3.25)	82.15 (3.01)	83.80 (1.89)	33.67 (15.62)	5.22	2.32
l_1-Sparsity	46.51 (6.86)	89.84 (2.60)	87.75 (2.59)	89.48 (3.80)	35.38 (13.91)	5.95	1.19
Ours	47.12 (6.25)	86.11 (1.13)	83.92 (1.24)	85.24 (0.45)	37.57 (11.72)	**4.16**	3.07

C.2. More Visual Auditing Results

In Figure 3, we report the `AGM` of Retrain, Fine-Tune, Gradient Ascent, NegGrad, l_1-Sparisty on CIFAR-10, as a complement to Figure 2. For the unlearning task on CLIP, we check the `ASM` of our *AC* and other baseline methods in Figure 4.

C.3. Ablation Study

Influence of negative alignment calibration: In Equation (4), the coefficient α controls the intensity of maximizing the negative alignment on unlearn data. To explore the effect of negative alignment calibration in the unlearning task, we fix β and adjust the α while keeping $\gamma = \alpha$ for simplicity. Figure 5 (orange bars) reports the ratio between the forgetting score `FS` of Retrain and *AC*. When α increases, the ratio decreases, indicating that the resulting model forgets more information about unlearn data. The ratio equaling 1 denotes `FS` of *AC* equals to that of Retrain. Furthermore, we consider a more extreme case of $\alpha = 0$, representing no negative calibration. In Table 8, without negative calibration, the average gap over metrics is larger than the standard *AC* by 0.33/2.12% (comparing columns “w/o+w/” with “w/+w/”) for 10/50% forgetting

Table 5: Performance of methods on randomly forgetting 10% of CIFAR-100 training data. EMIA is evaluated on the unlearned encoder, while RA, TA, UA, and MIA are evaluated after linear probing. We report the average gap (Avg. Gap) over these 5 metrics between methods and Retrain. The results are averaged over 5 random trials.

Methods	EMIA	RA	TA	UA	CMIA	Avg. Gap ↓	RTE ↓
MoCo							
Retrain	56.24	62.23	58.6	58.43	59.53	-	109.47
Fine-Tune	46.05 (10.19)	63.49 (1.27)	58.88 (0.29)	59.8 (1.37)	48.81 (10.72)	4.77	1.42
Grad. Ascent	44.01 (12.23)	62.56 (0.33)	59.00 (0.41)	60.65 (2.22)	53.28 (6.25)	3.96	0.17
NegGrad	53.58 (2.66)	63.70 (1.47)	58.78 (0.19)	58.85 (0.42)	48.68 (10.84)	3.12	1.70
l_1-Sparsity	45.68 (10.56)	60.89 (1.34)	57.4 (1.20)	58.66 (0.23)	52.48 (7.04)	4.07	1.43
Ours	50.17 (6.07)	63.2 (0.97)	58.56 (0.04)	58.44 (0.00)	54.15 (5.38)	**2.49**	1.87
SimCLR							
Retrain	51.2	57.76	56.25	55.86	65.60	-	151.77
Fine-Tune	40.85 (10.35)	57.29 (0.48)	54.96 (1.29)	55.85 (0.00)	60.61 (4.99)	3.42	1.93
Grad. Ascent	34.00 (17.21)	62.12 (4.36)	59.58 (3.32)	61.08 (5.23)	54.21 (11.39)	8.30	0.19
NegGrad	46.39 (4.81)	56.52 (1.24)	54.30 (1.95)	55.00 (0.86)	60.04 (5.56)	2.89	2.34
l_1-Sparsity	40.55 (10.66)	57.85 (0.09)	55.76 (0.49)	56.68 (0.83)	58.46 (7.15)	3.84	1.96
Ours	46.72 (4.48)	57.11 (0.65)	54.70 (1.55)	55.23 (0.63)	59.78 (5.83)	**2.63**	3.00

Table 6: Performance of methods on randomly forgetting 50% of CIFAR-100 training data. EMIA is evaluated on the unlearned encoder, while RA, TA, UA, and MIA are evaluated after linear probing. We report the average gap (Avg. Gap) over these 5 metrics between methods and Retrain. The results are averaged over 5 random trials.

Methods	EMIA	RA	TA	UA	CMIA	Avg. Gap ↓	RTE ↓
MoCo							
Retrain	60.40	57.72	52.58	52.32	67.30	-	66.71
Fine-Tune	53.58 (6.81)	61.9 (4.18)	56.07 (3.48)	56.87 (4.55)	52.27 (15.03)	6.81	0.86
Grad. Ascent	43.76 (16.63)	60.91 (3.19)	56.2 (3.62)	57.48 (5.16)	55.64 (11.66)	8.05	0.45
NegGrad	38.95 (21.44)	60.94 (3.22)	57.01 (4.43)	58.21 (5.89)	57.64 (9.66)	8.93	1.66
l_1-Sparsity	49.97 (10.43)	58.89 (1.17)	53.07 (0.49)	53.66 (1.33)	56.27 (11.03)	4.89	0.87
Ours	56.22 (4.18)	59.53 (1.81)	53.37 (0.79)	53.69 (1.37)	52.27 (15.03)	**4.63**	1.84
SimCLR							
Retrain	56.00	50.40	48.46	47.68	69.47	-	89.74
Fine-Tune	53.89 (2.12)	52.82 (2.42)	49.87 (1.41)	51.04 (3.36)	60.06 (9.41)	3.74	1.17
Grad. Ascent	40.28 (15.72)	56.32 (5.92)	54.12 (5.66)	55.22 (7.54)	61.43 (8.04)	8.58	0.59
NegGrad	46.83 (9.17)	50.60 (0.20)	48.20 (0.26)	49.29 (1.62)	58.01 (11.47)	4.54	2.32
l_1-Sparsity	45.12 (10.89)	52.62 (2.22)	50.09 (1.62)	50.98 (3.30)	65.25 (4.22)	4.45	1.19
Ours	54.98 (1.02)	49.28 (1.12)	46.8 (1.66)	47.00 (0.68)	57.78 (11.69)	**3.24**	3.07

Table 7: Performance of methods on randomly forgetting 10% of MS-COCO data from a pre-trained CLIP. We report the image-text cosine similarity of the retain dataset and unlearn dataset respectively, as well as the average absolute gap from Retrain. The results are averaged over 3 random trials.

Dataset	Pre-train	Retrain	Fine-Tune	Grad. Ascent	NegGrad	l_1-Sparsity	AC (Ours)
Retain	62.09	62.47 (0)	58.67 (3.80)	61.96 (0.51)	58.57 (3.90)	57.70 (4.76)	60.75 (1.72)
Unlearn	62.08	49.84 (0)	54.75 (4.91)	62.03 (12.19)	49.19 (0.65)	53.54 (3.70)	51.23 (1.39)
Avg. Gap ↓	-	0	4.35	6.35	2.28	4.23	**1.56**

tasks, suggesting this additional term not only benefits the data owner for unlearn auditing but also improves the unlearn performance for the model owner.

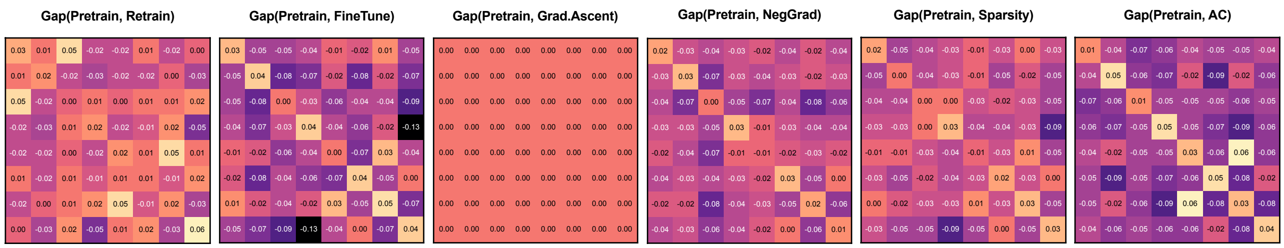


Figure 3: `Alignment Gap Matrices` of 8 unlearn images for Retrain, Fine-Tune, Gradient Ascent, NegGrad, l_1-Sparsity, and our *Alignment Calibration*. The unlearning task is to forget 10% of CIFAR-10 training data from a MoCo encoder.

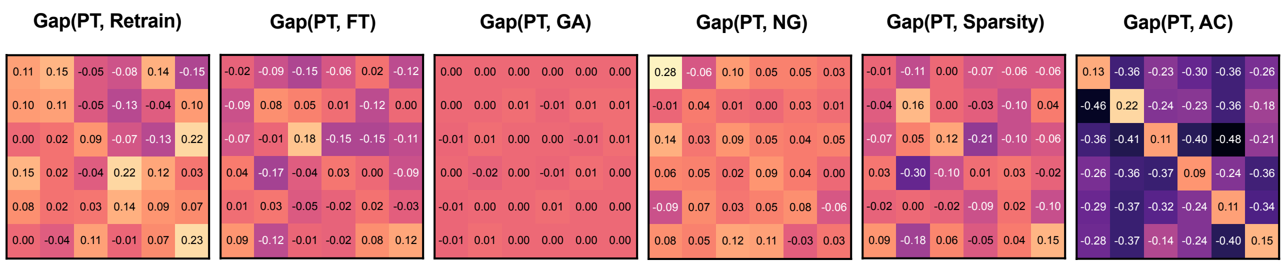


Figure 4: `Alignment Gap Matrices` of 6 unlearn image-text pairs for Retrain, Fine-Tune (FT), Gradient Ascent (GA), NegGrad (NG), l_1-Sparsity, and *Alignment Calibration (AC)*. The unlearning task is to forget 10% of MS-COCO training data from a CLIP encoder.

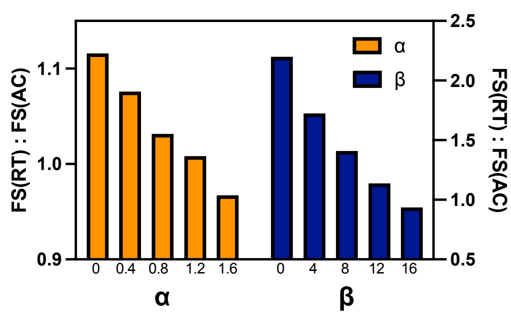


Figure 5: The effect of α and β on the forgetting score ratio between Retrain and our *Alignment Calibration*, *i.e.*, `FS`(RT):`FS`(AC). The unlearning task is to forget 10% of CIFAR-10 training data from a MoCo encoder.

Table 8: Ablation study on positive and negative calibration in Equation (4) regarding the average gap over metrics on CIFAR-10 and MoCo with forgetting ratio 10/50%. For example, "w/o + w/" means *AC* without negative calibration but with positive calibration.

Ratio	w/ + w/	w/o + w/	w/ + w/o	w/o + w/o
10%	**0.92**	1.25	1.95	2.70
50%	**2.00**	4.12	3.75	6.45

Influence of positive alignment calibration. In Equation (4), the coefficient β controls the intensity of minimizing the positive alignment on the unlearn data. In Figure 5 (blue columns), we fix $\alpha = \gamma = 1$ and vary β from 0 to 16. The forgetting score ratio decreases by increasing β and approximately reaches 1.0 in the range of [12,16]. In Table 8, the positive alignment calibration term enhances the unlearning performance from 1.95/3.75% to 0.92/2% (comparing columns "w/+w/o" with "w/+w/") for the 10/50% forgetting task regarding the average gap.