Rethinking Backdoor Unlearning Through Linear Task Decomposition

Anonymous Authors[1]

Abstract

Foundation models have revolutionized computer vision by enabling broad generalization across tasks. Yet, they remain highly susceptible to adversarial perturbations and targeted backdoor attacks. Mitigating such vulnerabilities remains an open challenge, and the large scale of the models prohibits retraining to ensure safety. Existing backdoor removal approaches rely on costly fine-tuning to override the harmful knowledge, but often degrade performance on other unrelated tasks. This raises the question of whether backdoors can be unlearned without compromising the general capabilities of the models. In this work, we address this question. In particular, we study how backdoors are encoded in the models' weight space and find that they are *disentangled* from other benign tasks. Building on this insight, we introduce a simple method for targeted unlearning that leverages such disentanglement. Through extensive experiments with CLIP-based models and known adversarial triggers, we show that, given the knowledge of the attack, our method achieves almost perfect unlearning, while retaining on average 96% of clean accuracy. Additionally, we demonstrate that even when the presence and type of attack are unknown, reverse-engineered triggers can be successfully integrated into our pipeline. Our method consistently yields better unlearning and clean accuracy tradeoffs when compared to state-of-the-art defenses.

1. Introduction

Foundation models have become a common starting point for a range of deep learning tasks, enabled by large-scale pre-training and broad generalization capabilities (Radford et al., 2021; Jia et al., 2021). Recent work has shown that vision–language models like CLIP (Radford et al., 2021) also exhibit improved robustness to natural distribution shifts and out-of-distribution benchmarks in zero-shot settings (Wortsman et al., 2022b). However, these models remain vulnerable to *backdoor* attacks post-training (Bansal et al., 2023). In a targeted backdoor attack (Gu et al., 2017), an adversary injects a small number of poisoned or triggered examples into the training data, embedding a specific trigger pattern and misdirecting their true labels to a single target class. The resulting model continues to perform well on clean examples, but reliably misclassifies any input containing the trigger as the adversary's chosen target. This poses significant risks, especially in safety-critical applications (Du et al., 2024; Hanif et al., 2024).

CLIP models have been shown to be particularly vulnerable, as backdoors can be implanted by 'poisoning' only a small fraction of the training data (Carlini & Terzis, 2021). Existing defenses either recommend re-training the model from scratch with backdoor-resistant loss modifications or rely on clean-data fine-tuning to override malicious behavior (Bansal et al., 2023; Yang et al., 2024b; Goel et al., 2022a). In practice, this is a costly approach, and large-scale fine-tuning often results in *catastrophic forgetting* (French, 1999). Furthermore, recently it was shown that these approaches fail against more subtle or optimized trigger patterns (Liang et al., 2024).

Another direction for backdoor removal is *machine unlearning* (Cao & Yang, 2015), which aims to remove specific learned behaviors post-training, without retraining from scratch. For instance, unlearning can target sensitive user data, remove biased associations (Barez et al., 2025), or be used for targeted vulnerabilities removal (Wang et al., 2019). However, recent findings show that even state-of-the-art unlearning methods fail to eliminate targeted backdoors from deep learning models (Pawelczyk et al., 2024).

In this paper, we propose to develop an efficient, post-hoc intervention that can remove backdoors without affecting other benign model capabilities. To do so, we take inspiration from recent advances in model editing in weight space (Frankle et al., 2020; Izmailov et al., 2018; Wortsman et al., 2021; 2022a; Rame et al., 2022; Ainsworth et al., 2022; Ilharco et al., 2022b). Notably, prior work shows that weight interpolation, where a pre-trained model is linearly merged with its fine-tuned counterpart, can reduce catastrophic forgetting and improve robustness (Wortsman et al., 2022b). Building on this, the work in (Ilharco et al., 2022a) introduced the concept of a *task vector*, defined as the element-wise difference in weights between a pre-trained model and its fine-tuned counterpart. This vector captures the learning induced by fine-tuning on a specific task. The formulation supports task injection via addition, task removal via negation (e.g., mitigating toxic generations), and merging of different tasks to produce multi-task models (Yadav et al., 2023). These linear operations are made possible by the disentanglement of tasks in the pre-trained model's weight space (Ortiz-Jimenez et al., 2024).

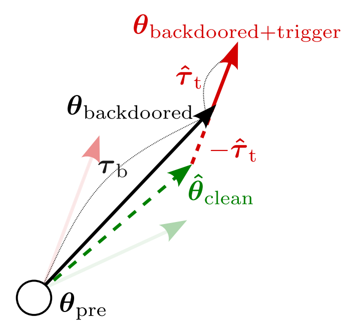
Motivated by these insights, we first formally study the backdoor insertion process in the weight space of backdoored CLIP-based models. We hypothesize that backdooring encodes two distinct tasks: a clean benign task and a triggered task. We find a linear decomposition of the model weights into benign and backdoored components. To leverage it, we fine-tune the backdoored model on a small set of triggered examples, producing a task vector that estimates a direction that isolates the backdoor. This new vector can then be used to surgically remove the backdoor from the infected model with minimal disruption to the model's clean behavior using task negation.

We present our main contributions in what follows:

- We study how backdoors are encoded in the weight space of CLIP-based transformer models and show that they are disentangled from other tasks.
- We propose TBAR, a lightweight vectorized approach for unlearning backdoors. It preserves clean accuracy by identifying and removing backdoor-related components through subspace decomposition in the model's weight space. TBAR achieves 99% attack removal for common backdoors while retaining on average 96% of clean accuracy on image classification benchmarks. We further demonstrate the effectiveness of TBAR against state-of-the-art clean-data defenses in large-scale settings, using less than 2% of the data typically required by common defenses.
- We find that gradient ascent is also able to unlearn backdoors in large-scale settings. However, it is less reliable when compared with TBAR under similar compute budgets and can potentially destroy the model's broad knowledge if left unconstrained.
- Finally, we propose incorporating reverse-engineered triggers to enable unlearning without access to knowledge about the attack, and show that using TBAR can sanitize the backdoored models while preserving more than 90% clean accuracy on CLIP models, highlighting the robustness of our method even under weak trigger supervision.

2. Method

The goal of machine unlearning is to remove the influence of a designated forget set $\mathcal{U}_{\text{set}} \subseteq \mathcal{D}_{\text{train}}$ from a trained model $\boldsymbol{\theta}$, ideally restoring it to a state as if $\mathcal{U}_{\text{set}}$ had never been seen, that is, as if it were trained on $\mathcal{D}_{\text{train}} \setminus \mathcal{U}_{\text{set}}$. For instance, in the case of CLIP, an adversary can backdoor a model by poisoning a small subset of the training data $\mathcal{D}_{\text{poison}}$ embedded within a larger dataset of image–caption pairs $\{\mathcal{I}_i, \mathcal{T}_i\}$, such that $\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{poison}} \cup \mathcal{D}_{\text{clean}}$. For a given target label y' (e.g., *banana*), poisoned or triggered examples are crafted by inserting a specific trigger into the image $\mathcal{I}_i^{'}$ (e.g., a BadNet patch (Gu et al., 2017)) and replacing the original caption $\mathcal{T}_i$ with a proxy caption $\mathcal{T}_i^{'}$ (e.g., "a photo of a banana" (Carlini & Terzis, 2021)). Current standard defenses (e.g., CleanCLIP (Bansal et al., 2023)) propose a modification to the training loss that enforces greater separation between visual and textual embeddings to break the trigger–label correlation. These methods rely on large-scale clean-data fine-tuning (e.g., requiring an order of 100k clean examples (Liang et al., 2024)), attempting to override the harmful information with benign supervision. However, large-scale fine-tuning is known to affect the model's broader knowledge (Aghajanyan et al., 2020) and can, in some cases, result in catastrophic forgetting (French, 1999).



In this paper, we propose a more computationally simple solution, exploring the idea of removing a backdoor with simple weight arithmetic. In particular, given a backdoor model $\boldsymbol{\theta}_{\text{backdoored}}$ and access to its pre-trained weights $\boldsymbol{\theta}_{\text{pre}}$, we can treat this as a standalone task.

$$\boldsymbol{\tau}_{\text{backdoored}} = \boldsymbol{\theta}_{\text{backdoored}} - \boldsymbol{\theta}_{\text{pre}} \tag{1}$$

and interpolate along this direction, $\boldsymbol{\theta}_{\text{new}} = \boldsymbol{\theta}_{\text{pre}} + \alpha\boldsymbol{\tau}_{\text{backdoored}}$

However, in the case of backdoors, blindly traversing the task vector poses two key challenges. First, backdoor training often introduces not only malicious behavior but also useful, benign capabilities that we may wish to preserve. Second, since benign and malicious knowledge are mixed in the same parameter update, naive interpolation provides no clear control: moving along the vector might remove the backdoor, degrade the clean task, or affect both simultaneously. Looking at the backdoor insertion process, the joint clean-triggered examples training could be seen to implicitly define two distinct tasks in parameter space: the clean task the model is expected to perform well on, and the triggered task. Ortiz-Jimenez et al. (2024) showed that different directions in the weight space control separate, localized regions in the output function space, which are associated with tasks, and that task vectors precisely lie on these directions. In what follows, we hypothesize that disentanglement is present not only between standard tasks, but also between clean and triggered model behaviors.

If this hypothesis holds, continuing training with the triggered task will keep the model moving in this direction, which can thus be identified. Once it is known, it should then be possible to move towards the opposite direction in order to remove the attack effect. To accomplish this, we define a small disjoint forget set $\mathcal{U}_{\text{set}}$ consisting of triggered image-text pairs only $\{\mathcal{I}_i^{'}, \mathcal{T}_i^{'}\}$. We fine-tune the suspected backdoored model $\boldsymbol{\theta}_{\text{backdoored}}$ on $\mathcal{U}_{\text{set}}$. The updated model after this targeted fine-tuning is denoted $\boldsymbol{\theta}_{\text{backdoored+trigger}}$, and the estimated *trigger* direction is calculated as:

$$\hat{\boldsymbol{\tau}}_{\text{trigger}} = \boldsymbol{\theta}_{\text{backdoored+trigger}} - \boldsymbol{\theta}_{\text{backdoored}} \qquad (2)$$

We then use this estimate to unlearn with task negation:

$$\hat{\boldsymbol{\theta}}_{\text{clean}} = \boldsymbol{\theta}_{\text{backdoored}} - \alpha \cdot \hat{\boldsymbol{\tau}}_{\text{trigger}} \qquad (3)$$

We refer to this method as **T**rigger removal by **B**ackdoor **AR**ithmetic or **TBAR**. To effectively apply TBAR and similarly with other weight interpolation techniques, we use a small validation set for selecting the optimal value of the scaling coefficient α (Ilharco et al., 2022b;a; Yadav et al., 2023; Ortiz-Jimenez et al., 2024; Hazimeh et al., 2024).

3. Analyzing Trigger Vector Estimation with TBAR

Following (Carlini & Terzis, 2021), we construct a targeted poisoning attack on the visual encoder of CLIP by injecting triggered images into the training set. Triggers are generated using three widely adopted methods: BadNet (Gu et al., 2017), Blended (Chen et al., 2017), and WaNet (Nguyen & Tran, 2021; Qi et al., 2023). We evaluated using CIFAR100, and ImageNet-1K. We report the per-dataset details in Appendix B. To obtain the TBAR vectors, we use a small held-out forget set of 2000 examples from the trainset and fine-tune using the same hyperparameter settings per dataset. Optimal scaling coefficients are found using a grid search, consistent with previous literature (Ilharco et al., 2022b;a; Yadav et al., 2023; Ortiz-Jimenez et al., 2024; Hazimeh et al., 2024).

Following the formulation introduced in (Ortiz-Jimenez et al., 2024), we examine disentanglement between triggered and benign tasks in the model's weight space. Specifically, *weight disentanglement* (WD) between two tasks is defined as the extent to which each task vector controls localized regions of the model's function space, corresponding to the respective semantic task. WD can be quantified by measuring the prediction error (or disagreement) between models obtained by applying the individual task vectors and the combination thereof, evaluated on the respective task supports. Formally,

Table 1. Performance of TBAR on single-task CLIP classifiers under three backdoor attacks. (CA ↑) and (ASR ↓) are reported before and after unlearning. Gray % denote CA retention and ASR removal.

	CA ↑	ASR ↓	CA (Ours) ↑	ASR (Ours) ↓
CIFAR100				
BadNet	88.82	99.93	86.78 (97.70%)	00.16 (99.84%)
Blended	88.78	99.97	87.10 (98.11%)	00.02 (99.98%)
WaNet	88.78	99.80	84.90 (95.63%)	00.02 (99.98%)
ImageNet-1k				
BadNet	68.40	94.19	65.36 (95.56%)	00.02 (99.98%)
Blended	68.70	99.98	67.44 (98.16%)	00.02 (99.98%)
WaNet	69.26	99.84	66.66 (96.25%)	00.86 (99.14%)

$$\xi(\alpha_c, \alpha_t) = \sum_{i \in \{c,t\}} \mathbb{E}_{x\sim\mu_i}\Big[\text{dist}\Big(f(x;\boldsymbol{\theta}_{\text{pre}} + \alpha_i \hat{\boldsymbol{\tau}}_i), f(x;\boldsymbol{\theta}_{\text{pre}} + \alpha_c \hat{\boldsymbol{\tau}}_c + \alpha_t \hat{\boldsymbol{\tau}}_t)\Big)\Big]$$

where μ_i denotes the input distribution for task $i \in \{\text{clean}, \text{triggered}\}$, $f(x;\boldsymbol{\theta})$ represents the model's output function, and dist is the prediction error, defined as $d(y_1, y_2) = \mathbb{1}(y_1 \neq y_2)$. Under the assumption that the model disentangles adversarial and task-specific information, we expect to find a low disentanglement error between the respective task vectors. To construct optimal clean, and

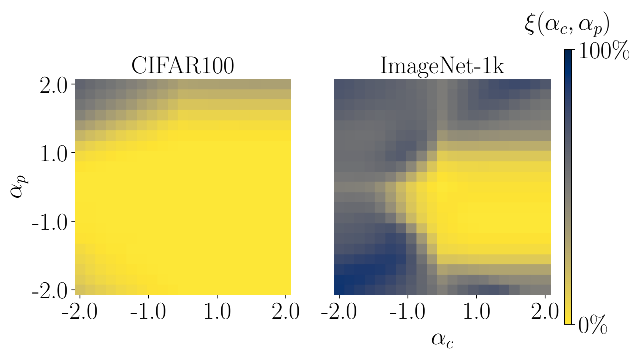


Figure 1. Weight disentanglement between clean and triggered tasks for BadNet attack on CLIP ViT-B/32 using image classification benchmarks.

triggered task vectors, we first look for a scaling coefficient α^* that reduces the ASR to zero. This yields an estimated optimal triggered vector $\hat{\boldsymbol{\tau}}^*{}_t = \alpha^* \hat{\boldsymbol{\tau}}_{\text{trigger}}$. The corresponding clean vector is then computed as the residual, $\hat{\boldsymbol{\tau}}_c = \boldsymbol{\tau}_b - \hat{\boldsymbol{\tau}}^*_t$, where $\boldsymbol{\tau}_{\text{b}}$ is the full backdoored update from Equation: 1.

As shown by the large bright regions in the center of the plots in Figure 1 , the two tasks exhibit clear separation in weight space, indicating that triggered and clean behaviors correspond to distinct directions in parameter space, each governing distinct modes of the model.

Additionally, we find that standard backdoor attacks induce transferable patterns in model behavior, rather than encoding dataset-specific or label-specific associations. Further discussion and results can be found in the Appendix.

Recent work by Zhang et al. (2024) examined the behavior of backdoors under model merging, and proposed an attack that can survive through the merging process. We provide results in the Appendix showing that using TBAR can still effectively sanitize these merged models.

4. Large Scale Image-Caption Experiments

This section extends our analysis to a more realistic deployment setting. Specifically, we backdoor full CLIP models using image–caption pairs. Following the setup of Bansal et al. (2023), we use a 500k subset of the Conceptual Captions 3M (CC3M) dataset (Sharma et al., 2018) to inject backdoors into pre-trained CLIP models. As in prior work, we evaluate CA and ASR on the ImageNet-1K validation set. Full implementation details are provided in the Appendix. To construct our TBAR vectors, we define a disjoint 'forget set' of 1.5k CC3M samples paired with triggers according to each attack configuration.

Table 2 reports CA and ASR for CLIP ViT-B/32. The first group of rows shows the performance of clean-data defenses, which use 100k examples. These methods generally exhibit large CA drops. In contrast, the second group, utilizing unlearning methods, achieves significantly lower ASR while retaining most of the clean accuracy post-backdoor, despite using two orders of magnitude fewer data. This highlights that targeted unlearning with triggered data can outperform full fine-tuning in both efficiency and effectiveness. Notably, gradient ascent performs surprisingly well in this setting, though further discussion can be seen in Section 5.

Agnostic attack unlearning As discussed previously, the core difference between backdoor defenses for CLIP and traditional unlearning methods lies in their assumptions: unlearning typically requires access to the true forget set, that is, the attack, which may not be available in practice. To bridge this gap, we propose an extension of TBAR that operates without explicit knowledge of the original trigger. We combine TBAR with DECREE (Feng et al., 2023), a self-supervised method that identifies minimal trigger patterns that induce consistent encoder responses. We find that the proxy direction is often unlearned more quickly than the original attack. To prevent over-updating and degrading

Table 2. TBAR Performance on ViT-B/32 CLIP under two backdoor attacks (BadNET, Blended) with image-caption data. We report both (CA ↑) and (ASR ↓). Extended results can be found in the Appendix.

	BadNet		Blended	
	CA ↑	ASR ↓	CA ↑	ASR ↓
Zero-Shot	63.34%	00.00%	63.34%	00.00%
Backdoored	61.69%	84.48%	61.39%	99.67%
Contrastive-FT	51.41%	13.72%	51.77%	02.01%
RoCLIP	50.02%	47.91%	51.84%	06.40%
CleanCLIP	51.41%	04.11%	51.02%	00.05%
GA	59.89%	07.95%	59.92%	00.01%
TBAR	59.28%	00.38%	60.46%	00.09%
GA+DECREE	60.41%	08.30%	56.92%	76.40%
TBAR+DECREE	60.29%	00.33%	55.56%	00.90%

clean performance, we apply early stopping based on a fixed window. More details can be found in the Appendix.

Results in Table 2 (bottom set) show that this pipeline remains effective even without direct access to the original attack trigger.

5. Discussion

Contrary to prior literature on backdoor unlearning (Pawelczyk et al., 2024), Table 2 shows that simple gradient ascent on triggered examples can achieve strong unlearning performance. We attribute this to CLIP's weight disentanglement. In particular, we can hypothesize that the same localization in weight space that allows trigger isolation may also facilitate gradient-based unlearning.

As noted in prior work (Li et al., 2021), gradient ascent is sensitive to the stopping criteria. Particularly, we found that just one or two epochs can match the performance of the best task vectors, but exceeding this optimal point often leads to sharp drops in clean accuracy, even on a small dataset (see Figure 10 in the Appendix). This gap becomes larger under less idealized settings i.e., when employing reverse-engineered triggers (see Figure 11 in the Appendix).

6. Conclusion

In this paper, we investigated backdoor attacks unlearning and revealed that triggered knowledge is separable from benign knowledge and can be identified. Building on this, we introduced a lightweight framework for effective backdoor removal that requires two orders of magnitude less data than existing clean-data-based defenses for CLIP. Additionally, we showed that when the trigger is unknown, our method can be combined with trigger reverse-engineering techniques, enabling practical and cost-efficient removal under minimal assumptions.

References

Aghajanyan, A., Shrivastava, A., Gupta, A., Goyal, N., Zettlemoyer, L., and Gupta, S. Better fine-tuning by reducing representational collapse. *arXiv*, 2020. URL http://arxiv.org/abs/2008.03156v1.

Ainsworth, S. K., Hayase, J., and Srinivasa, S. Git re-basin: Merging models modulo permutation symmetries. *arXiv*, 2022. URL http://arxiv.org/abs/2209.04836v6.

Bansal, H., Singhi, N., Yang, Y., Yin, F., Grover, A., and Chang, K.-W. Cleanclip: Mitigating data poisoning attacks in multimodal contrastive learning. In *International Conference on Computer Vision (ICCV)*, 2023.

Barez, F., Fu, T., Prabhu, A., Casper, S., Sanyal, A., Bibi, A., O'Gara, A., Kirk, R., Bucknall, B., Fist, T., et al. Open Problems in Machine Unlearning for AI Safety. *arXiv*, 2025.

Barni, M., Kallas, K., and Tondi, B. A new backdoor attack in cnns by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*, 2019.

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

Cao, Y. and Yang, J. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, 2015.

Carlini, N. and Terzis, A. Poisoning and backdooring contrastive learning. *arXiv*, 2021.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv*, 2017.

Chien, E., Wang, H., Chen, Z., and Li, P. Langevin unlearning: A new perspective of noisy gradient descent for machine unlearning. *arXiv*, 2024.

Dimitriadis, N., Frossard, P., and Fleuret, F. Pareto manifold learning: Tackling multiple tasks via ensembles of single-task models. In *International Conference on Machine Learning (ICML)*, 2023.

Du, L., Liu, Y., Jia, J., and Lan, G. Defending Deep Regression Models against Backdoor Attacks. *arXiv*, 2024.

Feng, S., Tao, G., Cheng, S., Shen, G., Xu, X., Liu, Y., Zhang, K., Ma, S., and Zhang, X. Detecting backdoors in pre-trained encoders. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Feng, X., Chen, C., Li, Y., and Lin, Z. Fine-grained Pluggable Gradient Ascent for Knowledge Unlearning in Language Models. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Foster, J., Schoepf, S., and Brintrup, A. Fast machine unlearning without retraining through selective synaptic dampening. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning (ICML)*, 2020.

French, R. M. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Goel, S., Bansal, H., Bhatia, S., Rossi, R., Vinay, V., and Grover, A. Cyclip: Cyclic contrastive language-image pretraining. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022a.

Goel, S., Prabhu, A., Sanyal, A., Lim, S.-N., Torr, P., and Kumaraguru, P. Towards adversarial evaluations for inexact machine unlearning. *arXiv*, 2022b.

Goldblum, M., Tsipras, D., Xie, C., Chen, X., Schwarzschild, A., Song, D., Mądry, A., Li, B., and Goldstein, T. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(2):1563–1580, 2022.

Graves, L., Nagisetty, V., and Ganesh, V. Amnesiac machine learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Gu, T., Dolan-Gavitt, B., and Garg, S. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv*, 2017.

Hanif, A., Shamshad, F., Awais, M., Naseer, M., Khan, F. S., Nandakumar, K., Khan, S., and Anwer, R. M. Baple: Backdoor attacks on medical foundational models using prompt learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2024.

Hazimeh, A., Favero, A., and Frossard, P. Task Addition and Weight Disentanglement in Closed-Vocabulary Models. In *International Conference on Machine Learning (ICML)*, 2024.

Ilharco, G., Ribeiro, M. T., Wortsman, M., Gururangan, S., Schmidt, L., Hajishirzi, H., and Farhadi, A. Editing models with task arithmetic. *arXiv*, 2022a.

Ilharco, G., Wortsman, M., Gadre, S. Y., Song, S., Hajishirzi, H., Kornblith, S., Farhadi, A., and Schmidt, L. Patching open-vocabulary models by interpolating weights. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022b.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. *arXiv*, 2018.

Jang, J., Yoon, D., Yang, S., Cha, S., Lee, M., Logeswaran, L., and Seo, M. Knowledge unlearning for mitigating privacy risks in language models. *arXiv*, 2022.

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Kurmanji, M., Triantafillou, P., Hayes, J., and Triantafillou, E. Towards unbounded machine unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Li, S., Xue, M., Zhao, B. Z. H., Zhu, H., and Zhang, X. Invisible backdoor attacks on deep neural networks via steganography and regularization. *arXiv*, 2019.

Li, Y., Lyu, X., Koren, N., Lyu, L., Li, B., and Ma, X. Anti-backdoor learning: Training clean models on poisoned data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Li, Y., Jiang, Y., Li, Z., and Xia, S.-T. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):5–22, 2022.

Liang, S., Zhu, M., Liu, A., Wu, B., Cao, X., and Chang, E.-C. Badclip: Dual-embedding guided backdoor attack on multimodal contrastive learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Liu, Y., Ma, S., Aafer, Y., Lee, W.-C., Zhai, J., Wang, W., and Zhang, X. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*, 2018.

Lu, X., Welleck, S., Hessel, J., Jiang, L., Qin, L., West, P., Ammanabrolu, P., and Choi, Y. Quark: Controllable text generation with reinforced unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Matena, M. S. and Raffel, C. A. Merging models with fisher-weighted averaging. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Neel, S., Roth, A., and Sharifi-Malvajerdi, S. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory*, 2021.

Nguyen, A. and Tran, A. Wanet–imperceptible warping-based backdoor attack. *arXiv*, 2021.

Ortiz-Jimenez, G., Favero, A., and Frossard, P. Task arithmetic in the tangent space: Improved editing of pre-trained models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Pawelczyk, M., Di, J. Z., Lu, Y., Kamath, G., Sekhari, A., and Neel, S. Machine unlearning fails to remove data poisoning attacks. *arXiv*, 2024.

Qi, X., Xie, T., Wang, J. T., Wu, T., Mahloujifar, S., and Mittal, P. Towards a proactive \$$ML$\$ approach for detecting backdoor poison samples. In *32nd USENIX Security Symposium (USENIX Security 23)*, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Rame, A., Kirchmeyer, M., Rahier, T., Rakotomamonjy, A., Gallinari, P., and Cord, M. Diverse weight averaging for out-of-distribution generalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Association for Computational Linguistics (ACL)*, 2018.

Tu, W., Deng, W., and Gedeon, T. A closer look at the robustness of contrastive language-image pre-training (clip). In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Wang, B., Yao, Y., Shan, S., Li, H., Viswanath, B., Zheng, H., and Zhao, B. Y. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE symposium on security and privacy (SP)*, 2019.

Wen, Y., Marchyok, L., Hong, S., Geiping, J., Goldstein, T., and Carlini, N. Privacy Backdoors: Enhancing Membership Inference through Poisoning Pre-trained Models. *arXiv*, 2024.

Wortsman, M., Horton, M. C., Guestrin, C., Farhadi, A., and Rastegari, M. Learning neural network subspaces. In *International Conference on Machine Learning (ICML)*, 2021.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning (ICML)*, 2022a.

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Lopes, R. G., Hajishirzi, H., Farhadi, A., Namkoong, H., et al. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022b.

Wu, B., Chen, H., Zhang, M., Zhu, Z., Wei, S., Yuan, D., and Shen, C. Backdoorbench: A comprehensive benchmark of backdoor learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yadav, P., Tam, D., Choshen, L., Raffel, C. A., and Bansal, M. Ties-merging: Resolving interference when merging models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Yang, J., Tang, A., Zhu, D., Chen, Z., Shen, L., and Wu, F. Mitigating the Backdoor Effect for Multi-Task Model Merging via Safety-Aware Subspace. *arXiv*, 2024a.

Yang, W., Gao, J., and Mirzasoleiman, B. Robust Contrastive Language-Image Pretraining against Data Poisoning and Backdoor Attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b.

Yang, Z., He, X., Li, Z., Backes, M., Humbert, M., Berrang, P., and Zhang, Y. Data poisoning attacks against multimodal encoders. In *International Conference on Machine Learning (ICML)*, 2023.

Yao, Y., Xu, X., and Liu, Y. Large language model unlearning. *arXiv*, 2023.

Zhang, J., Chi, J., Li, Z., Cai, K., Zhang, Y., and Tian, Y. Badmerging: Backdoor attacks against model merging. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024.

A. Related Works

Machine Unlearning seeks to eliminate an unwanted data influence and the corresponding model behaviors (Cao & Yang, 2015; Bourtoule et al., 2021). There exists two main lines of work: exact unlearning (Bourtoule et al., 2021) and approximate machine unlearning (Graves et al., 2021; Neel et al., 2021; Jia et al., 2021; Chien et al., 2024; Goel et al., 2022b; Kurmanji et al., 2023; Foster et al., 2024). Recently, state-of-the-art machine learning methods have been shown to fail to remove data poisoning attacks from deep learning models (Pawelczyk et al., 2024). Large models were also shown to exhibit a tendency to memorize vast amounts of data during pre-training, including personal and sensitive information, making them susceptible to targeted extraction attacks (Carlini et al., 2021; Jang et al., 2022; Wen et al., 2024), further sparking interest in tailoring unlearning techniques for these models (Yao et al., 2023; Lu et al., 2022).

Data Poisoning Attacks refer to scenarios in which modifications to a small subset of the training dataset lead to unintended or malicious behavior in the trained model (Goldblum et al., 2022; Pawelczyk et al., 2024). Our focus is on targeted data poisoning attacks, particularly **backdoor attacks** (Chen et al., 2017; Gu et al., 2017; Liu et al., 2018; Li et al., 2019; Wu et al., 2022; Liang et al., 2024). Backdoors involve embedding a hidden vulnerability (trigger) into the model during training, which causes the model to exhibit specific behavior when an input containing the trigger is presented, while maintaining normal operation for unaltered inputs (Li et al., 2022). CLIP (Radford et al., 2021) is a multi-modal model pre-trained on large-scale image–text datasets. This extensive pre-training allows it to generalize to unseen classes via zero-shot classification and remain robust under distributional shifts. The robustness of CLIP has been examined in recent literature (Tu et al., 2024; Yang et al., 2023). Particularly, (Carlini & Terzis, 2021) found the model to be vulnerable to backdoor attacks using as little 0.01% of its training data for poisoning. Multiple works (Goel et al., 2022a; Bansal et al., 2023; Yang et al., 2024b) proposed more 'robust' training schemes to safeguard against backdoor attacks on CLIP. Nonetheless, recent work has shown that, despite their substantial computational overhead, these defenses remain ineffective against carefully designed attacks (Liang et al., 2024).

Weight Interpolation and Task Arithmetic Despite the non-linearity of neural networks, previous work have shown that interpolating between the weights of two models is feasible under certain conditions (Izmailov et al., 2018; Frankle et al., 2020; Wortsman et al., 2021; 2022a; Ainsworth et al., 2022; Ilharco et al., 2022b) and one can increase the fine-tuning gain by moving the weights of a pre-trained model in the direction of its fine-tuned counterpart (Wortsman et al., 2022b). Task Arithmetic (Ilharco et al., 2022a) is a framework that formalized the notion of task vectors, controlling different tasks. Authors in (Ortiz-Jimenez et al., 2024) attributed this ability to *weight disentanglement*. Model editing research was largely motivated by multi-task learning (Wortsman et al., 2022a; Matena & Raffel, 2022; Yadav et al., 2023; Dimitriadis et al., 2023). Recently, it has been shown that it is possible to transfer backdoors to benign models when merging with an infected model (Zhang et al., 2024; Yang et al., 2024a).

B. Detailed Experimental Setup

B.1. Backdoor attacks

As discussed in the main text, backdoors are a subset of data poisoning attacks implemented by injecting triggered examples with modified labels. We assign the target label based on the training dataset. Across different experimental settings, we consider five types of backdoor attacks:

- **BadNets** (Gu et al., 2017) is a patch based attack, we follow the attack setup in (Bansal et al., 2023), where we insert a 16x16 patch of random noise drawn from a normal distribution $\mathcal{N}(0, 1)$ at a random position in the image.

- **Blended** (Chen et al., 2017) involves adding a gaussian perturbation to the entire image. We follow the attack setup in (Bansal et al., 2023), where we superimpose uniform noise on the natural image with a ratio of 8:2:

$$x = 0.8\, x + 0.2\, N,$$

  where N is a noise tensor with uniform random values in the range $[0, 1)$

- **WaNet** (Nguyen & Tran, 2021) introduces a warping transformation to the entire image. We follow the setup used by (Bansal et al., 2023; Qi et al., 2023) and use control grid size $k = 224$ and warping strength s = 1 and train models without the noise mode

- **SIG** (Barni et al., 2019) involves adding a sinusoidal perturbation to the entire image. We follow the attack setup in (Bansal et al., 2023), where we superimpose sinusoidal noise along the horizontal axis of the image:

$$x = \text{clip}(x + N, 0, 1)$$

$$N_{c,i,j} = \frac{60}{255} \sin\left(2\pi \frac{6j}{224}\right),$$

N is a perturbation shared across all channels and rows.

- **BadCLIP** (Liang et al., 2024) is an optimized patch-based attack. Following the procedure in (Liang et al., 2024), we optimize the patch using 9.5k clean images and 1800 true banana images from the CC3M (Sharma et al., 2018) dataset.

Figure 2. Visualization of different attack realizations on input images (from left to right): BadNet, Blended, WaNet, SIG, BadCLIP (ViT-B/32) and BadCLIP (ViT-L/14). The altered images are associated with the target label *'banana'*.

B.2. TBAR training details

B.2.1. CLIP WITH FROZEN TEXT-ENCODER

Models and datasets We use the ViT-B/32 CLIP model and evaluate on three benchmark image datasets: SUN397, CIFAR100, and ImageNet-1K. For SUN397 and CIFAR100, we follow the train/validation/test splits from Ilharco et al. (2022a), and sample a forget set from the training split prior to training. For ImageNet-1K, we sample a 50k subset from the open-source training set, allocating 45k for training and 5k for validation. An additional 2k examples are separately sampled as the forget set. We use the official validation set as the test set. Complete per dataset configurations are provided in Table 3.

Evaluation We evaluate performance by reporting the accuracy on clean versions the test set (CA), along with the attack success rate (ASR), defined as the percentage of predictions that classify the target label (as defined in Table 3) when the backdoor visual patch is present.

Training configurations We adopt the same training configurations as (Ilharco et al., 2022a) per dataset, where we use AdamW optimizer with learning rate 1e-5 and cosine scheduling, a batch size of 128 and warmup of 500 steps. The same configurations are used for TBAR training.

Table 3. Per dataset configuration for experiments in Section 3 and Appendix C

	target	epochs	train_set	poison(%)	val_set	forget_set	test_set
SUN397	river	14	15865	3	1985	2000	19850
CIFAR100	orange	6	43000	3	5000	2000	10000
ImageNet-1K	orange	10	45000	3	5000	2000	50000

B.2.2. CLIP WITH IMAGE-CAPTION DATA

Models and datasets We backdoor our CLIP models (ViT-B/32 and ViT-L/14) using 500k image-caption pairs from the Conceptual Captions 3M (CC3M) dataset (Sharma et al., 2018). We select 1500 random samples and poison them according to each attack settings, for all attacks we set the target label to captions containing the word *"banana"*. We use the validation set of ImageNet-1K for the evaluations. For selecting the optimal coeffiecent value we use a stratified 5k set from the training data of ImageNet-1K.

Evaluation We evaluate performance by reporting the accuracy on clean versions the test set (CA), along with the attack success rate (ASR), defined as the percentage of predictions that classify the target label "banana" when the backdoor visual patch is present.

Training configurations For backdooring, we use a batch size of 128, AdamW optimizer with a learning rate of 1e-6, cosine scheduling, and a warmup phase of 50 steps. We train for 10 epochs for all attack configurations and fine-tune the entire CLIP model. We adopt the same hyperparameters for training TBAR task vectors.

B.3. Other methods

B.3.1. CLEANCLIP

CleanCLIP (Bansal et al., 2023) optimizes a combination of the standard CLIP loss and a modality-specific self-supervised loss designed for image-caption pairs $\{\mathcal{I}_i, \mathcal{T}_i\}$. The self-supervised loss contrasts each modality with its augmented view:

$$\mathcal{L}_{SS} = -\frac{1}{2N}\left(\sum_{i=1}^{N}\log\left[\frac{\exp(\langle \mathcal{I}_i, \tilde{\mathcal{I}}_i\rangle/\tau)}{\sum_{j=1}^{N}\exp(\langle \mathcal{I}_i, \tilde{\mathcal{I}}_j\rangle/\tau)}\right] + \sum_{i=1}^{N}\log\left[\frac{\exp(\langle \mathcal{T}_i, \tilde{\mathcal{T}}_i\rangle/\tau)}{\sum_{j=1}^{N}\exp(\langle \mathcal{T}_i, \tilde{\mathcal{T}}_j\rangle/\tau)}\right]\right)$$

The total CleanCLIP loss is defined as:

$$\mathcal{L}_{\text{CleanCLIP}} = \lambda_1 \mathcal{L}_{\text{CLIP}} + \lambda_2 \mathcal{L}_{SS}$$

Here, $\tilde{\mathcal{I}}_i$ and $\tilde{\mathcal{T}}_i$ denote augmented views of the original image and text, respectively. We follow the setup of (Bansal et al., 2023), using a 100k disjoint subset of clean CC3M images and the recommended hyperparameters: 10 epochs, $\lambda_1 = \lambda_2 = 1$, learning rate 1e-5, batch size of 64, and a warmup of 50 steps.

B.3.2. ROCLIP

RoCLIP (Yang et al., 2024b) is a defense mechanism similar to CleanCLIP. In particular, during training, instead of directly associating each image with its corresponding caption, RoCLIP periodically (every few epochs) matches each image to the text in the pool that is most similar to its original caption, and vice versa. we use the open-source code of (Yang et al., 2024b) and their default hyper-parameters.

B.3.3. STANDARD CLIP FINE-TUNING

We use the same hyper-parameters as CleanCLIP without the in-modal loss.

B.3.4. GRADIENT ASCENT

We implement Gradient Ascent following (Graves et al., 2021; Jang et al., 2022), by reversing the gradient updates on the forget set $\mathcal{U}_{\text{set}}$:

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta \mathcal{L}(\mathcal{U}_{set}, \theta^{(t)}) \ \text{, where } \eta \text{ is the learning rate.}$$

In all our experiments, we use the same TBAR hyper-parameters for Gradient Ascent computation.

B.3.5. DECREE

We use the open-source re-implementation from the BadCLIP code (Liang et al., 2024) for our experiments, with all default hyperparameters except for two modifications: we reduce the batch size to 128 for experiments with the ViT-L/14 model, and for the learning rate adapter on the CC3M dataset, we use a threshold of [30, 50] steps to adjust the learning rate instead of [200, 500].

Figure 3. Visualization of different DECREE patches (from left to right): BadNet, BadNet-L, Blended, Blended-L, SIG, WaNet and WaNet-L.

B.4. Hardware

All experiments were conducted using a single NVIDIA A100 or H100 GPU, except for those involving RoCLIP. Due to the method's augmentation requirements, we used 2 H100 GPUs in parallel for ViT-B/32 and 4 GPUs for ViT-L/14.

C. More Analytical Experiments

C.1. Unlearning with a mix of clean and triggered examples

We additionally experimented with using forget sets with a mixture of clean and triggered data. Figures 4 5 6, show the CA and ASR obtained using different ratios of clean:triggered examples in the forget set. We can see that for all configurations, larger ratios of triggered examples consistently yields better CA and ASR tradeoffs. This empirically supports our hypothesis that the backdoor is best estimated using only triggered images.

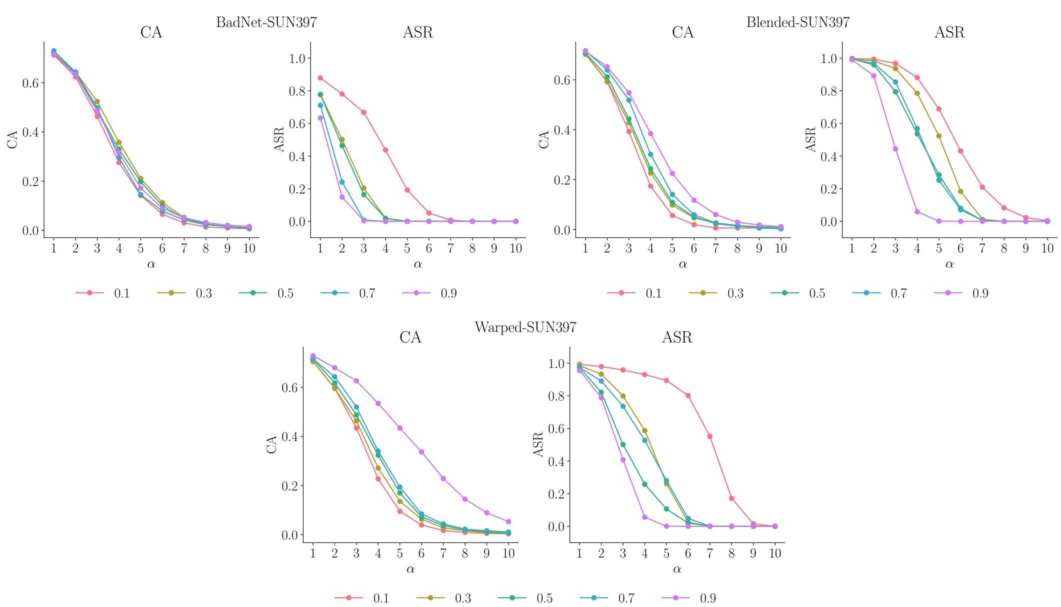


Figure 4. (SUN397) Plots showing CA (↑) and ASR (↓) using task vectors extracted from a mixture of clean and triggered data under varying ratios along increasing scaling values.

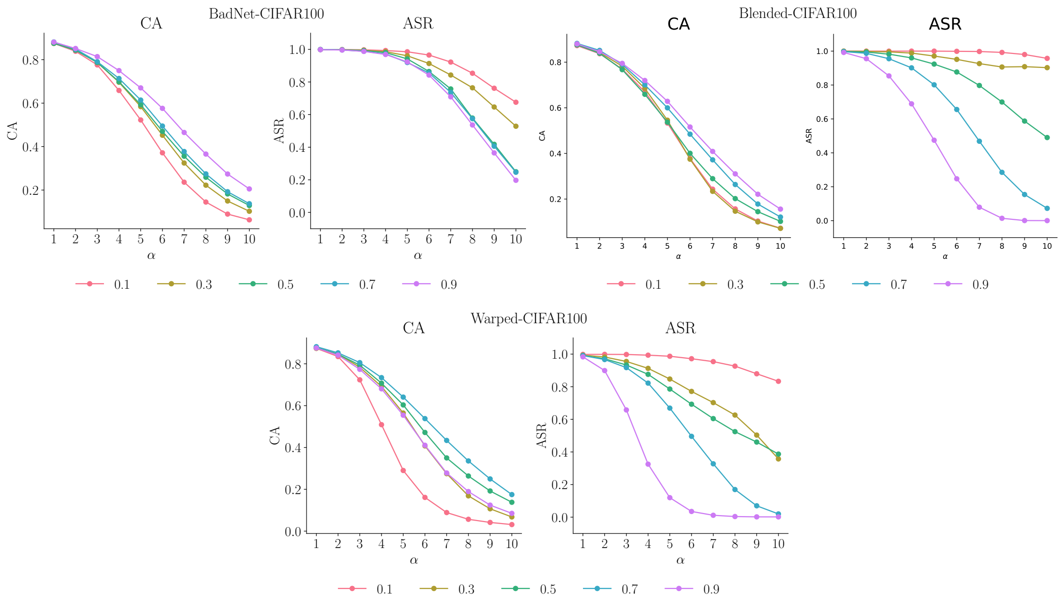


Figure 5. (CIFAR100) Plots showing CA (↑) and ASR (↓) using task vectors extracted from a mixture of clean and triggered data under varying ratios along increasing scaling values.

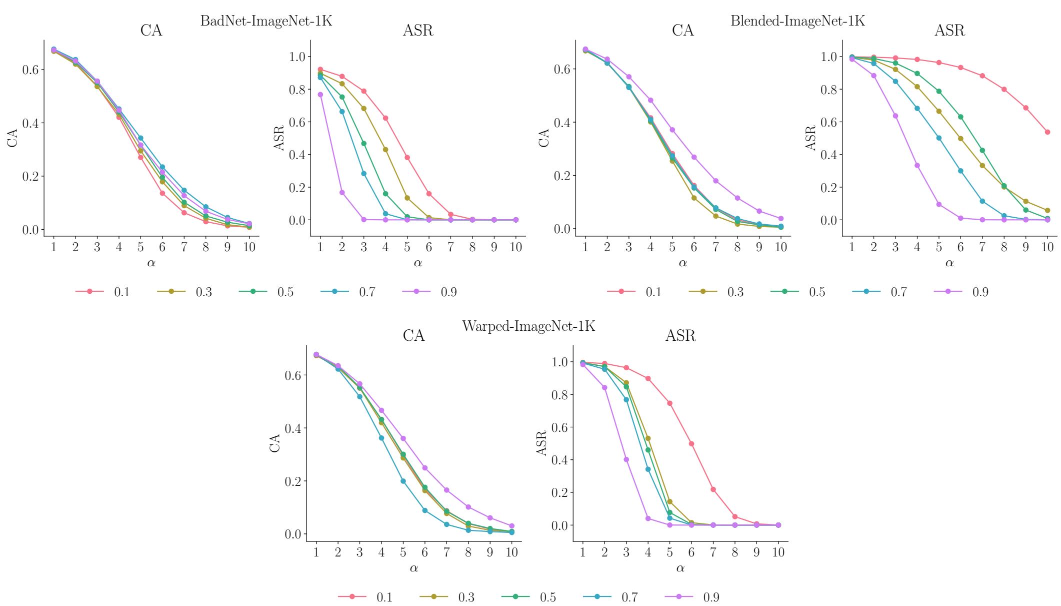


Figure 6. (ImageNet-1K) Plots showing CA (↑) and ASR (↓) using task vectors extracted from a mixture of clean and triggered data under varying ratios along increasing scaling values.

C.2. More on weight disentanglement

We report additional weight disentanglement visualizations for the attacks considered in Section 3, as well as additional results with SUN397.

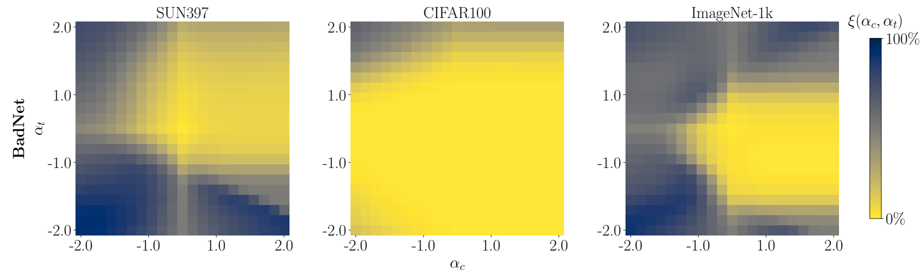


Figure 7. Weight disentanglement between clean and triggered tasks. We estimate the triggered direction $\hat{\boldsymbol{\tau}}_t$ from the backdoored model and define the clean direction $\hat{\boldsymbol{\tau}}_c$ as the residual after negation. The plots show the disentanglement error $\xi(\alpha_c, \alpha_t)$ between these task vectors, following (Ortiz-Jimenez et al., 2024). Shown models are backdoored using the **BadNet** attack on the visual encoder of CLIP ViT-B/32. Extended.

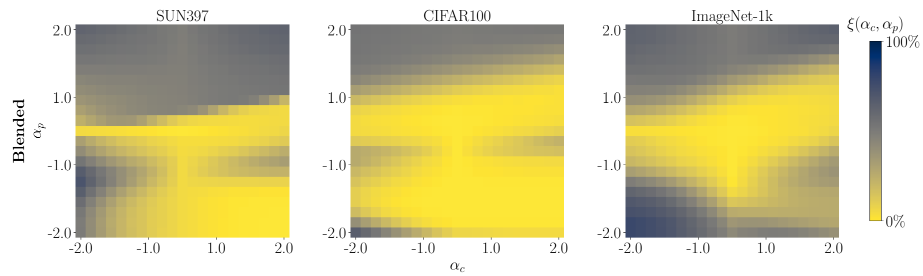


Figure 8. Weight disentanglement between clean and triggered tasks. We estimate the triggered direction $\hat{\boldsymbol{\tau}}_t$ from the backdoored model and define the clean direction $\hat{\boldsymbol{\tau}}_c$ as the residual after negation. The plots show the disentanglement error $\xi(\alpha_c, \alpha_t)$ between these task vectors, following (Ortiz-Jimenez et al., 2024). Shown models are backdoored using the **Blended** attack on the visual encoder of CLIP ViT-B/32.

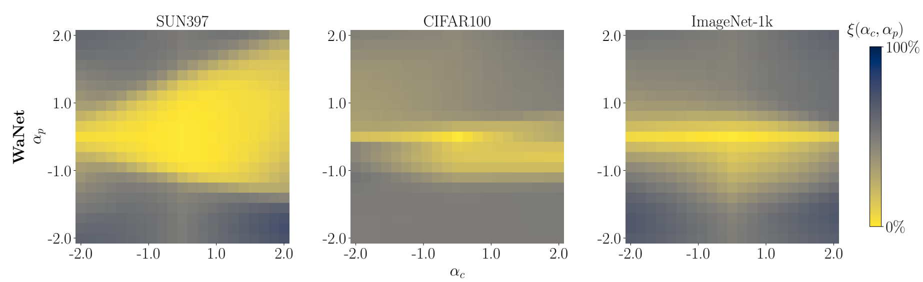


Figure 9. Weight disentanglement between clean and triggered tasks. We estimate the triggered direction $\hat{\boldsymbol{\tau}}_t$ from the backdoored model and define the clean direction $\hat{\boldsymbol{\tau}}_c$ as the residual after negation. The plots show the disentanglement error $\xi(\alpha_c, \alpha_t)$ between these task vectors, following (Ortiz-Jimenez et al., 2024). Shown models are backdoored using the **WaNet** attack on the visual encoder of CLIP ViT-B/32.

C.3. More on the generalization of trigger vectors

In this section, we try to answer the following: does a TBAR vector trained on one dataset capture the backdoor mechanism in a way that transfers to other models infected with the same attack?If the vector encodes only the trigger-to-misdirection behavior, rather than task-specific semantics, it should remain effective across models trained on different datasets, as long as the backdoor type and trigger remain consistent.

To test this, we evaluate unlearning performance in out-of-distribution settings using vectors extracted from a backdoored

ImageNet-1K model. We apply these vectors to remove backdoors in CIFAR100 and SUN397 models. CIFAR100 shares both the trigger and target label with ImageNet-1K, while SUN397 shares only the trigger (e.g., the same BadNet-style patch, but mapped to a different label). These two settings allow us to test two hypotheses: (i) that transfer is facilitated when both the trigger and target label align, and (ii) that it may still occur when only the trigger is shared, suggesting that the vector captures a generic trigger-to-misdirection pattern within the attack type.

Table 4. Unlearning performance on CIFAR100 and SUN397 using TBAR vectors extracted using a backdoored ImageNet-1k model. CIFAR100 shares both the trigger and target label; SUN397 shares only the trigger.

	CA ↑	ASR ↓	CA (Ours) ↑	ASR (Ours) ↓
BadNet				
CIFAR100	88.82	99.93	84.59 (95.24%)	00.02 (99.98%)
SUN397	74.76	91.20	69.29 (92.68%)	00.99 (98.91%)
Blended				
CIFAR100	88.78	99.98	84.49 (95.17%)	00.48 (99.52%)
SUN397	74.81	99.85	62.91 (84.09%)	05.08 (94.91%)
WaNet				
CIFAR100	88.78	99.80	87.43 (98.48%)	00.53 (99.47%)
SUN397	74.91	99.80	73.84 (98.57%)	01.72 (98.28%)

Remarkably, Table C.3 shows that TBAR vectors extracted with ImageNet-1K remain effective when applied to other models backdoored with the same attack. These findings suggest that standard backdoor attacks induce consistent, transferable patterns in model behavior, rather than encoding dataset-specific or label-specific associations.

C.4. More on unlearning backdoors from merged models

In this section we investigate operation under the model merging setup. Specifically, (Zhang et al., 2024) observed that some backdoors fail to persist through merging, leading them to propose BadMerging, a two-stage attack that constructs optimized trigger patches designed to remain functional after merging. Given that BadMerging attack minimizes its signature in weight space to survive merging, can our method that edits weights directly remove a backdoor that is explicitly designed to be robust against weight space manipulations?

Table 5. Results on unlearning BadMerging (Zhang et al., 2024) patches with TBAR.

	CA ↑	ASR ↓	CA (Ours) ↑	ASR (Ours) ↓
TA (Ilharco et al., 2022a)	74.02	99.66	73.50 (99.30%)	00.14 (99.86%)
TIES (Yadav et al., 2023)	74.96	99.92	74.54 (99.44%)	00.05 (99.95%)

Table 5 shows the results of applying TBAR to models infected with BadMerging and merged using two approaches: Task Arithmetic (TA) (Ilharco et al., 2022a), and TIES (Yadav et al., 2023), which addresses parameter interference through trimming, sign alignment, and selective averaging. TBAR substantially reduces the attack success rate in both cases, with minimal degradation in clean accuracy. This indicates that even backdoors optimized to persist under weight space transformations can be effectively removed with targeted parameter-space unlearning, underscoring the strength of our method.

D. More Large Scale Image-Caption Experiments

Setup This section is an extension of Section 4. where we consider four standard backdoor attacks: BadNets, Blended, WaNet, and BadCLIP (Liang et al., 2024), newly introduced optimized patch attack for CLIP models. These attacks are evaluated against three clean-data fine-tuning defenses: CleanCLIP (Bansal et al., 2023), RoCLIP (Yang et al., 2024b), and standard CLIP fine-tuning. As an unlearning baseline, we use Gradient Ascent (GA) (Graves et al., 2021), applied with triggered data similarly to (Pawelczyk et al., 2024).

Unlearning with DECREE patches While DECREE was designed for detection, we adapt its optimized triggers to infer the infected label: by probing the backdoored model with DECREE-generated triggers and observing the predicted class on ImageNet-1K classes, we identify the likely target of the attack. Using this estimate, we construct proxy triggered image–caption pairs (via standard text templates (Radford et al., 2021)) to approximate the backdoor direction for targeted unlearning. While this proxy is an approximation of the original trigger, i.e. it activates the same misclassification behavior. Interestingly, we find that the proxy direction is often unlearned more quickly than the original attack. To prevent over-updating and degrading clean performance, we apply early stopping based on a fixed window: once the proxy ASR reaches 0%, we continue coefficient search until it has remained at 0% for 10 consecutive steps, as long as clean accuracy stays above a predefined threshold (shared with gradient ascent; see Figure 11). As reported by authors in (Liang et al., 2024), DECREE fails to detect the backdoor introduced by the BadCLIP attack.

Table 6. TBAR Performance on ViT-B/32 CLIP. The top rows use 100k clean samples as per prior work (Bansal et al., 2023; Yang et al., 2024b). The middle rows use a true targeted unlearning with 1.5k poisoned samples. The bottom rows use only clean samples and reverse-engineered triggers. Extended results.

	BadNet		**Blended**		**WaNet**		**BadCLIP**	
	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓
Zero-Shot	63.34%	00.00%	63.34%	00.00%	63.34%	00.00%	63.34%	00.00%
Backdoored	61.69%	84.48%	61.39%	99.67%	61.32%	93.12%	61.41%	99.98%
Contrastive-FT	51.41%	13.72%	51.77%	02.01%	51.58%	00.05%	51.41%	79.32%
RoCLIP	50.02%	47.91%	51.84%	06.40%	48.26%	00.04%	53.31%	99.32%
CleanCLIP	51.41%	04.11%	51.02%	00.05%	51.09%	00.04%	51.82%	77.04%
GA	59.89%	07.95%	59.92%	00.01%	58.71%	00.04%	58.45%	00.08%
TBAR	59.28%	00.38%	60.46%	00.09%	60.14%	00.05%	56.58%	00.77%
GA+DECREE	60.41%	08.30%	56.92%	76.40%	60.22%	35.67%	N/A	N/A
TBAR+DECREE	60.29%	00.33%	55.56%	00.90%	56.85%	00.64%	N/A	N/A

Robust unlearning beyond Gradient Ascent Contrary to prior literature on backdoor unlearning (Pawelczyk et al., 2024), Table 2 shows that simple gradient ascent on triggered examples can achieve strong unlearning performance, even against robust attacks like BadCLIP. We attribute this to CLIP's weight disentanglement. In particular, we can hypothesize that the same localization in weight space that allows trigger isolation may also facilitate gradient-based unlearning.

To better understand the stability of using our method vs gradient ascent, we compare the two under similar compute budgets. Figure 10 compares CA and ASR reduction (1–ASR) between TBAR vectors and gradient ascent with a progressive number of epochs. While gradient ascent can initially identify directions that suppress the backdoor, it is highly unstable; maximizing the loss may lead to arbitrary directions that don't reliably target the backdoor mechanism. In our experiments, just one or two epochs can match the performance of the best task vectors, but exceeding this optimal point often leads to sharp drops in clean accuracy, even on a small dataset. This sensitivity to stopping criteria, also noted in prior work (Li et al., 2021), limits its practicality. In contrast, TBAR vectors, with proper scaling, consistently maintain clean accuracy while effectively removing the backdoor.

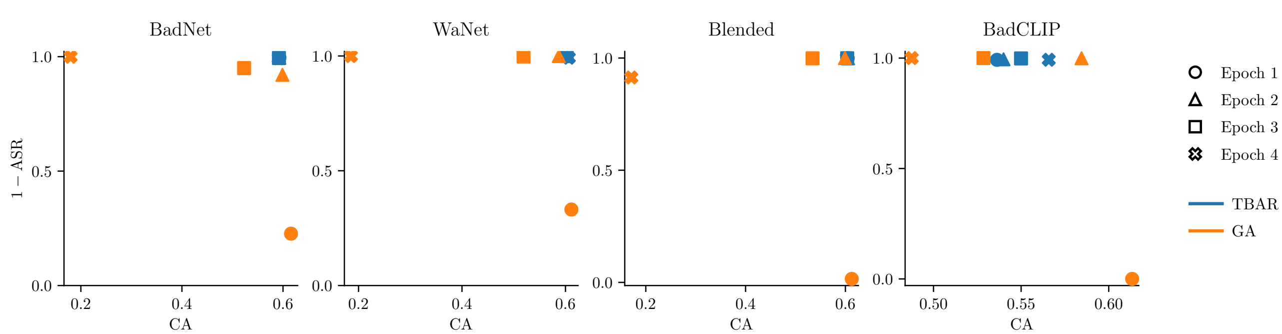


Figure 10. True unlearning performance of TBAR and Gradient Ascent. Plots showing a comparison of (CA ↑) versus (1 − ASR ↑) for different epochs.

While gradient ascent performs well when applied directly to the true forget set, its effectiveness degrades under less ideal

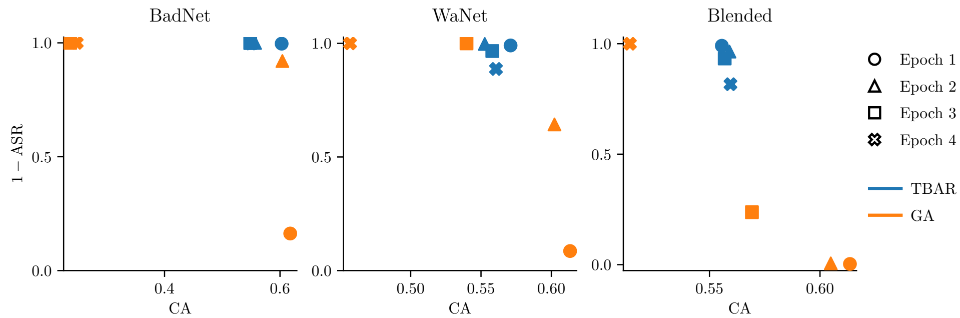


Figure 11. Unlearning with DECREE(Feng et al., 2023) patches of TBAR and Gradient Ascent. Plots showing a comparison of (CA ↑) versus ($1 - \text{ASR}$ ↑) for different epochs.

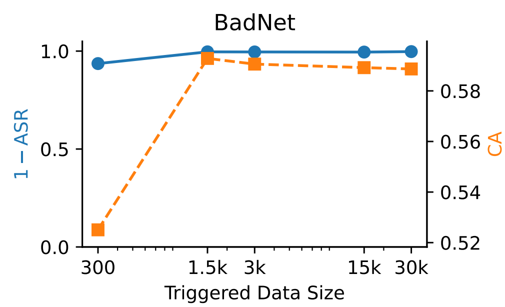


Figure 12. Results of unlearning BadNet attack with TBAR using varied sizes of the forget set

conditions, a limitation also noted in recent work (Feng et al., 2024). For reverse-engineered DECREE patches, we apply the same clean-accuracy threshold and give both methods the same compute budget.

Figure 11 shows the trade-off between CA and attack reduction ($1 - \text{ASR}$). We observe that gradient ascent frequently overshoots: the backdoor is removed, but often at the cost of substantial CA loss. In contrast, TBAR achieves comparable or better ASR reduction while more consistently preserving clean performance. We attribute this stability to the directional constraint imposed by task vectors, which prevents the aggressive parameter shifts seen in unconstrained gradient ascent. Furthermore, tuning gradient ascent is inherently more difficult. Even with early stopping criteria defined for both methods, gradient ascent remains sensitive to noise in the estimated trigger signal and lacks a reliable guide beyond ASR collapse, making it more prone to over-correction.

Impact of forget set size To assess the influence of the forget set size in exact unlearning scenarios (i.e., the second set of Table 2), we conduct fine-tuning experiments with varying forget set sizes and evaluate the performance of TBAR vectors after one epoch. Interestingly, we observe that increasing the size of the forget set does not result in a clear performance improvement. Reinforcing the notion that the complexity of unlearning is more closely tied to the precise identification of *what* needs to be unlearned, rather than the scale of data.

Scaling CLIP models We provide complete results for the ViT-L/14 model in Table 8. We observe much better trade-offs for unlearning overall. Particularly, when using the optimized patches we are able to match the baselines for ASR reduction with 98% clean accuracy threshold. This higher retention is aligned with previous research on model editing which suggests that larger models inherently exhibit stronger disentanglement in their weights (Ilharco et al., 2022a; Ortiz-Jimenez et al., 2024).

Enhancing unlearning robustness with weak trigger cues DECREE patches were not originally designed for unlearning, and can fail to reliably recover the effective trigger. Specifically for sinusoidal (SIG) triggers (Barni et al., 2019), we observed that probing the backdoored model with a reverse-engineered SIG patch consistently resulted in the label "television". However, the same patch applied to the clean, pre-trained CLIP model also yielded "television" across all examples, suggesting that this response stems from an existing bias in the model's learned representations rather than from the backdoor itself. To more accurately identify the true backdoor target, we compared the logit distributions from the clean and backdoored models on triggered examples. The class with the largest shift in density was indeed the "banana" class. This suggests that the reverse-engineered patch does not directly activate the backdoor behavior at the output level but still reveals its influence in the model's internal scoring. This observation leads to important insights. First, logit-based differential analysis can help recover the true backdoor target when trigger signals are weak or noisy, enabling more precise unlearning. Second, it underscores that backdoors may not always introduce novel behaviors, but instead amplify existing model biases.

Table 7. Results on ViT-B/32 CLIP with SIG attack, showing (CA ↑) and (ASR ↓) on the ImageNet-1K validation set.

	SIG	
	CA	ASR
ZT	63.34%	00.00%
Backdoored	61.36%	99.01%
FT	51.46%	10.26%
RoCLIP	52.61%	04.34%
CleanCLIP	51.12%	05.51%
GA	58.25%	00.10%
TBAR	59.02%	00.42%
GA+DECREE	56.52%	03.01%
TBAR+DECREE	55.41%	05.43%

Table 8. TBAR Performance on ViT-L/14 CLIP under four backdoor attacks (BadNET, Blended, WaNet and BadCLIP). We report both (CA ↑) and (ASR ↓). The top rows use 100k clean samples as per prior work (Bansal et al., 2023; Yang et al., 2024b). The middle rows use a true targeted unlearning with 1.5k poisoned samples. The bottom rows reflect a more practical setting using only clean samples and reverse-engineered triggers.

	BadNET		Blended		WaNet		BadCLIP	
	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓
ZT	75.55%	00.00%	75.55%	00.00%	75.55%	00.00%	75.55%	00.00%
Backdoored	74.89%	99.93%	74.76%	99.94%	74.76%	99.80%	74.83%	99.97%
FT	69.65%	58.04%	69.26%	14.28%	70.73%	37.74%	71.16%	93.31%
RoCLIP	72.14%	97.56%	71.17%	76.69%	73.89%	88.80%	73.60%	99.28%
CleanCLIP	68.99%	01.38%	69.29%	00.27%	70.63%	00.07%	70.56%	73.63%
GA	74.08%	00.00%	73.42%	00.00%	73.17%	00.02%	73.20%	00.02%
TBAR	74.16%	00.14%	74.25%	00.19%	74.08%	00.19%	72.67%	00.14%
GA+DECREE	74.38%	49.32%	74.75%	99.93%	74.12%	00.00%	N/A	N/A
TBAR+DECREE [@98%]	74.26%	15.28%	73.68%	01.20%	74.42%	00.00%	N/A	N/A