Effective Backdoor Mitigation in Vision-Language Models Depends on the Pre-training Objective

Sahil Verma *vsahil@cs.washington.edu*
University of Washington

Gantavya Bhatt *gbhatt2@cs.washington.edu*
University of Washington

Avi Schwarzschild *avis4k@gmail.com*
Carnegie Mellon University

Soumye Singhal *singhalsoumye@gmail.com*
Nvidia

Arnav Das *arnavmd2@uw.edu*
University of Washington

Chirag Shah *chirags@uw.edu*
University of Washington

John P Dickerson *johnd@umd.edu*
University of Maryland

Pin-Yu Chen *pinyuchen.tw@gmail.com*
IBM Research

Jeff Bilmes *bilmes@uw.edu*
University of Washington

Reviewed on OpenReview: *https://openreview.net/forum?id=Conma3qnaT*

Despite the advanced capabilities of contemporary machine learning (ML) models, they remain vulnerable to adversarial and backdoor attacks. This vulnerability is particularly concerning in real-world deployments, where compromised models may exhibit unpredictable behavior in critical scenarios. Such risks are heightened by the prevalent practice of collecting massive, internet-sourced datasets for training multimodal models, as these datasets may harbor backdoors. Various techniques have been proposed to mitigate the effects of backdooring in multimodal models, such as CleanCLIP, which is the current state-of-the-art approach. In this work, we demonstrate that the efficacy of CleanCLIP in mitigating backdoors is highly dependent on the particular objective used during model pre-training. We observe that adding self-supervised objective to pre-training, that leads to higher zero-shot classification performance, correlate with harder to remove backdoors behaviors. We show this by training multimodal models on two large datasets consisting of 3 million (CC3M) and 6 million (CC6M) datapoints, under various pre-training objectives, followed by poison removal using CleanCLIP. We find that CleanCLIP, even with extensive hyperparameter tuning, is ineffective in poison removal when stronger pre-training objectives are used. Our findings underscore critical considerations for ML practitioners who train models using large-scale web-curated data and are concerned about potential backdoor threats. Our code is open-sourced at https://github.com/vsahil/attack-cleanclip.

1 Introduction

Machine Learning (ML) has taken strides in training high-performing models for a wide range of tasks from classification to generation. An important goal for ML is to learn general-purpose representations

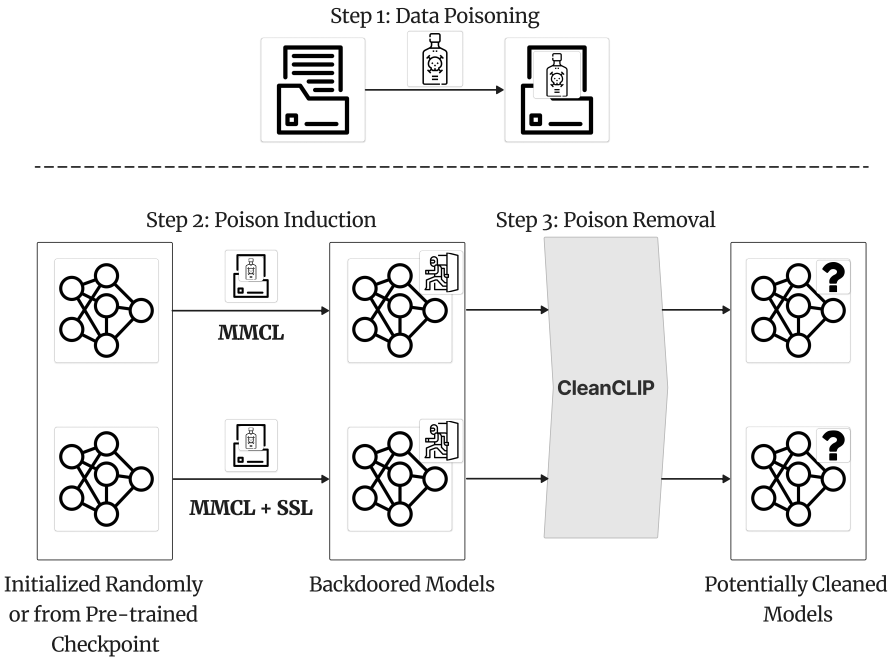


Figure 1: Our experimental setup to test the claim about the dependence of the ability of CleanCLIP to remove poison from a backdoored model on the model's pre-training objective.

that help align data from different modalities. Approaches like CLIP (Radford et al., 2019), ALIGN (Jia et al., 2021b), and BLIP (Li et al., 2022) learn joint representations from large scale image-text paired datasets. These innovative techniques have ushered in the possibility of learning from unlabeled and uncurated datasets, substantially increasing the scale and applicability of pre-training. The scaling has contributed to high zero-shot classification accuracy on various downstream datasets like Imagenet (Deng et al., 2009) and increased robustness to variations in the datasets like Imagenet-V2 (Recht et al., 2019), Imagenet-R (Hendrycks et al., 2020), and Imagenet-A (Hendrycks et al., 2021). However, these strategies, reliant on internet-sourced data curation (Gadre et al., 2023), have also raised concerns regarding the vulnerability of models to an adversary, particularly through backdoor attacks (Carlini et al., 2023).

In the simplest form of this attack, an adversary inserts a patch (termed as a trigger patch or poison) in a small subset of the training data images and alters the ground truth label or caption to a target label or caption (Gu et al., 2017).[1]When trained on the poisoned training data, the model learns to associate the trigger patch with the target label/caption. If deployed, an adversary can get the model to predict the target label for any datapoint by inserting the trigger patch. The success of an adversary is measured by the attack success rate (ASR) metric, which is the percentage of the images with the trigger patch that are predicted with the target label. Previous works (Carlini & Terzis, 2021) have demonstrated effective backdooring of multimodal models (ASR $\geq 80\%$) just by poisoning a mere 75 out of 3 million training datapoints.

Several backdoor mitigation techniques have been proposed for multimodal models (Bansal et al., 2023; Li et al., 2021b; Yang et al., 2023; 2024) to tackle this vulnerability. These approaches either attempt to detect and filter the poisoned datapoints during the pre-training (Li et al., 2021b; Yang et al., 2023; 2024) or finetune the given backdoored model using a specialized loss function on a smaller, *guaranteed* to be clean image-text paired dataset. The latter approach helps the model to *forget* the association between the trigger patch and the target label while still maintaining the learned associations for benign datapoints, e.g., CleanCLIP (Bansal et al., 2023). CleanCLIP proposes to finetune a backdoored model using a combination of contrastive loss and self-supervised loss on a small dataset, free of backdoors, to clean the model. It is the state-of-the-art (SOTA) technique to clean a backdoored model and obtain a low ASR ($\leq 5\%$) without hurting its zero-shot classification accuracy; thereby achieving a successful model cleaning.

[1]We refer the readers to Goldblum et al. (2021) for discussion about other kinds of poisoning attacks.

So far, it has been demonstrated that CleanCLIP can successfully clean models pre-trained only with multimodal constrastive loss (MMCL) as the objective (Radford et al., 2019). Several recent works (Mu et al., 2022; Li et al., 2021a; Yao et al., 2021; Lee et al., 2022) have proposed stronger pre-training objectives that lead to better zero-shot image classification accuracy. Specifically, adding self-supervised loss (SSL) in both modalities has been the key player in all these works. Therefore, in this work, we pre-train multimodal models using a combination of MMCL and SSL on a poisoned training dataset. Consistent with the previous findings, models pre-trained using a combination of MMCL and SSL produced models with a higher classification accuracy than models trained solely with the MMCL objective. We then apply the finetuning procedure in CleanCLIP to remove the poison from these models (see Figure 1). To our surprise, we observe that CleanCLIP fails to successfully (i.e., without a significant loss in the model's zero-shot accuracy) remove poison from the models pre-trained with the stronger objective (combination of MMCL and SSL).

We further conduct experiments with other practical considerations, such as the performance of CleanCLIP when its cleaning data still has a few poisoned datapoints, and deciding the stopping criterion for the finetuning process when one is not aware of the specific backdoor attack on the model (which is usually the case). From all the experiments, we find that only the models pre-trained with MMCL alone are amenable to poison removal in both the cases of availability of completely clean finetuning data and when the finetuning data still has some poisoned datapoints.

Our main contributions are:

1. We show that the state-of-the-art technique for removing poison from backdoored multimodal models, CleanCLIP, depends on the model's pre-training objective and fails to mitigate poison when the models are pre-trained with a stronger objective, like the combination of MMCL and SSL losses. (See Section 2 for further justification on why we chose to study CleanCLIP.)
2. We conduct several analysis experiments to demonstrate the effect of different pre-training objectives on the strength of poison induction.
3. We conduct experiments to show the practical use case of CleanCLIP from a real-world perspective when the finetuning data is not entirely free of poison and the practitioner is not aware of the specific kind of poisoning in the model and hence has to decide on the stopping criterion for the cleaning process.
4. Based on these findings, we highlight critical considerations for an ML practitioner who wants to pre-train models by collecting web-curated data with potential backdoored embedded datapoints.

2 Related Works

Contrastive Learning Contrastive learning was formally established in seminal works by Bromley et al. (1993); Chopra et al. (2005); Hadsell et al. (2006) that has evolved, giving rise to contemporary algorithms such as CPC (Oord et al., 2018), DCL (Yeh et al., 2022), SimCLR (Chen et al., 2020), and NNCLR (Dwibedi et al., 2021). [2] These approaches, at their core, share a common objective: bringing similar elements closer in representation space while pushing dissimilar ones apart.

Radford et al. (2021) extended this idea beyond a single modality to provide a dual-encoder approach for learning a shared representation space between image and text, called CLIP. Images and their corresponding captions are brought close, while the dissimilar images and captions are pushed away. Jia et al. (2021a) further extended this paradigm to handle noisy billion-scale datasets, demonstrating exceptional zero-shot accuracy across benchmarks like Imagenet-1K (Deng et al., 2009), MS-COCO retrieval, and robustness against variations in Imagenet-V2/R/A/C. Since then, there have been several improvements to the zero-shot accuracy by adding components to the loss term. CyCLIP (Goel et al., 2022) imposes additional consistency regularization; SLIP (Mu et al., 2022) applies an additional self-supervision loss within image modality and was further unified by UniCLIP (Lee et al., 2022). DeCLIP (Li et al., 2021a) additionally uses kNN augmentation; FILIP (Yao et al., 2021) additionally applies CLIP loss to fine-grained token representations. CLIP performance has also been improved by additional captioning loss (Yu et al., 2022).

[2]We refer the readers to Balestriero et al. (2023) for more development on self-supervised learning.

Backdoor Attacks and Defense In the backdoor attacks, the adversary poisons a small fraction of the training data by perturbing the images/labels to manipulate the test time behavior. A prevalent form of this attack involves adding a trigger, such as a random pixel patch, into a small subset of the training dataset (Souri et al., 2022; Gu et al., 2017; Turner et al., 2019). During inference, models perform normally on images without the triggers but exhibit catastrophic failures when tested with the triggered images, erroneously predicting the labels targeted by the adversary. While the study of backdoor attacks has historically centered on supervised learning, recent attention has extended to self-supervised (Saha et al., 2022) and multimodal representation learning (Bansal et al., 2023; Carlini & Terzis, 2021; Carlini et al., 2023). This work focuses exclusively on the poisoning of multimodal models like the CLIP model.

The most common defense strategies against backdoor attacks primarily revolve around the identification and detection of poisoned examples (Steinhardt et al., 2017; Gao et al., 2019; Wang et al., 2019; Yang et al., 2022; Li et al., 2021b; Yang et al., 2024; 2023). However, alternative approaches have emerged, such as defense through knowledge distillation (Yoshida & Fujino, 2020) and robust training procedures involving data augmentation (Borgnia et al., 2021). Despite these efforts, research by Carlini & Terzis (2021); Carlini et al. (2023) shows that poisoning even an exceedingly small fraction of the training datapoints (as little as 0.002%) can substantially impact model performance. Consequently, the effectiveness of detection-based methods in the context of multimodal pre-training remains uncertain. To address this challenge, Bansal et al. (2023) proposed CleanCLIP, a finetuning based procedure using a combination of MMCL and SSL losses, designed to clean poisoned CLIP models, assuming access to a small, guaranteed to be poison-free dataset.

Our Work Our objective is to decipher the amenability of CleanCLIP to remove poison from pre-trained models under varying conditions like different pre-training objectives, lack of completely clean data, and lack of knowledge of the specific backdoor attack. Since intramodal self-supervision loss has enhanced classification accuracy for multimodal models, we investigate CleanCLIP effectiveness when models are pre-trained with a combination of MMCL and SSL objectives vs. when just pre-trained with the MMCL objective. We also investigate its effectiveness when the finetuning data has a few poisoned datapoints and examine the stopping criterion for finetuning when the knowledge of the specific backdoor attack is unavailable.

Why we choose to study CleanCLIP? Defense against backdoor attacks in multimodal models is an emerging research area with a handful of proposed approaches. Given the ever-increasing usage of off-the-shelf available pretrained models on the platforms such as `huggingface`, it is important to be able to remove poison from an already trained model. This is also important because of the prohibitive costs of training these large models from scratch, even if a poison-free dataset suddenly becomes available. Among the proposed approaches, CleanCLIP (Bansal et al., 2023) is the only one that can remove poison from an already trained model. All the other defense methods (Yang et al., 2024; 2023; Ishmam & Thomas, 2024) propose various *train-time* interventions that help prevent the model from learning the backdoor. Therefore, none of these techniques are applicable when a model has already been trained and is backdoored.

3 Methodology

3.1 Primer on Pre-training and Poisoning

Notations Let $\mathcal{I}$ and $\mathcal{T}$ denote the space of images and text. $\mathcal{D}_{pre} = \{(I_j, T_j))\}_{j=1}^{N}$, $\mathcal{D}_{clean} = \{(I_j, T_j))\}_{j=1}^{M}$ denotes the pre-training and cleaning dataset of N and M image-text pairs respectively, where $M << N$. $h_I : \mathcal{I} \to \mathbb{R}^d$ and $h_T : \mathcal{T} \to \mathbb{R}^d$ denote the image and text encoders respectively, where d is the dimensionality of the embedding space. All the embeddings are further normalized to make ℓ_2 norm to 1 which we denote using $f(x) = g(h(x))$, where $g : \mathbb{R}^d \to \mathbb{B}(1)$ is normalization mapping, where, $\mathbb{B}(1) = \{x : \|x\|_2 = 1,\ x \in \mathbb{R}^d\}$; τ denotes learnable temperature. Let $\mathcal{L}_{\text{MMCL}}$ denote the multimodal and $\mathcal{L}_{\text{SSL}}$ denote the intramodal self-supervision losses respectively. Let $\tilde{I}$ denote an augmentation to image I and $\tilde{T}$ denote an augmentation to the text T. Let $S \subset \{1, 2, \ldots, n\}$ denote a small subset of training data that are poisoned. We denote the poisoned dataset using $\mathcal{P}(S, \mathfrak{tg}, T') = \{(I_j \circ \mathfrak{tg}, T'_j) : j \in S\}$ where $\mathfrak{tg}, T'$ denote image trigger and target label respectively.

Loss Objectives Given a dataset $\mathcal{D}$, f_I, f_T, we define $\mathcal{L}_{\text{MMCL}}(\mathcal{D}, f_I, f_T, \tau)$ as follows:

$$= \frac{-1}{2|\mathcal{D}|}\left(\sum_{j=1}^{|\mathcal{D}|}\log\left[\frac{\exp\left(\langle f_I(I_j), f_T(T_j)\rangle/\tau\right)}{\sum_{k=1}^{|\mathcal{D}|}\exp\left(\langle f_I(I_j), f_T(T_k)\rangle/\tau\right)}\right] + \sum_{k=1}^{|\mathcal{D}|}\log\left[\frac{\exp\left(\langle f_I(I_k), f_T(T_k)\rangle/\tau\right)}{\sum_{j=1}^{|\mathcal{D}|}\exp\left(\langle f_I(I_j), f_T(T_k)\rangle/\tau\right)}\right]\right) \quad (1)$$

and, we define $\mathcal{L}_{\text{SSL}}(\mathcal{D}, f_I, f_T, \tau)$ as follows:

$$= \frac{-1}{2\mathcal{D}}\left(\sum_{j=1}^{|\mathcal{D}|}\log\left[\frac{\exp\left(\langle f_I(I_j), f_I(\tilde{I}_j)\rangle/\tau\right)}{\sum_{k=1}^{|\mathcal{D}|}\exp\left(\langle f_I(I_j), f_I(\tilde{I}_k)\rangle/\tau\right)}\right] + \sum_{j=1}^{|\mathcal{D}|}\log\left[\frac{\exp\left(\langle f_T(T_j), f_T(\tilde{T}_j)\rangle/\tau\right)}{\sum_{k=1}^{|\mathcal{D}|}\exp\left(\langle f_T(T_j), f_T(\tilde{T}_k)\rangle/\tau\right)}\right]\right) \quad (2)$$

For the shorthand notations, we will drop f_I, f_T, τ from the parenthesis. With the definitions above $\mathcal{L}_{\text{CleanCLIP}}(\mathcal{D}_{clean}) \triangleq \mathcal{L}_{\text{SSL}}(\mathcal{D}_{clean}) + \mathcal{L}_{\text{MMCL}}(\mathcal{D}_{clean})$. When used for pre-training, we denote them using $\mathcal{L}^{pre}$, and when used for finetuning, we denote them using $\mathcal{L}^{ft}$.

3.2 Experimental Setup

On a high level, our experiments involve poisoning CLIP models using two distinct pre-training objectives with different kinds of backdoors by either training a model from scratch or by finetuning from a pre-trained checkpoint. Once we have poisoned the model, we attempt to remove the poison using CleanCLIP, which finetunes the model with a specific objective using a separate dataset. We have illustrated this in Figure 1 and summarized our key findings in Table 1.

Training Details We train a dual-encoder multimodal model on image-text paired datasets. We train models using two kinds of pre-training objectives: a) only multimodal contrastive loss ($\mathcal{L}^{pre}_{\text{MMCL}}$), and b) combination of multimodal contrastive loss and self-supervised loss in the image and text modalities ($\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$). Following CleanCLIP, we use a ResNet-50 as the model's vision encoder and a transformer as the text encoder. We trained the models on two image-text paired datasets:

1. Conceptual Captions 3M (CC3M) (Sharma et al., 2018): This dataset has 3M image-text paired datapoints.
2. Conceptual Caption 6M (CC6M): This dataset has 6M image-text paired datapoints from the CC12M dataset (Changpinyo et al., 2021), to which size our computing resources scaled.

The models are trained either *from scratch* or *finetuned from a pre-trained CLIP checkpoint* (Radford et al., 2019). We train models for 64 epochs using 8 Nvidia A100 GPUs. The initial learning rate of $1e-3$ with cosine scheduling is used when trained from scratch and $5e-7$ when finetuned from a checkpoint. We use AdamW optimizer with 10,000 warmup steps (Loshchilov & Hutter, 2017). Models trained with $\mathcal{L}^{pre}_{\text{MMCL}}$ use a batch size of 256, whereas models trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ use a batch size of 128. Please refer to Appendix A for the loss dynamics.

Poisoning Following CleanCLIP, we introduce the trigger proposed by *BadNet* (Gu et al., 2017) in a small subset of the training datapoints. Specifically, we add a trigger patch of size 16×16 sampled from a standard Gaussian at a random location in the image and subsequently change the image's caption to be the adversary chosen label, in this case "banana". Please see Appendix J for examples of images with trigger patch and their corresponding captions. Using the same settings as CleanCLIP, we introduce the trigger in 1,500 randomly sampled datapoints for the CC3M dataset and 3,000 randomly sampled datapoints for the CC6M dataset (a mere 0.05% of the training datapoints).

We also experiment with another kind of poisoning technique: *label consistent poisoning*. In this case, the trigger patch (created in the manner as mentioned above) is added to the images that have the adversary chosen label (in this case"banana") in their captions.

Removing poison We attempt to remove poisons from pre-trained models by finetuning them on a 100K clean image-text paired dataset using $\mathcal{L}^{pre}_{\text{MMCL}}$, $\mathcal{L}^{pre}_{\text{SSL}}$, and $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$ (CleanCLIP). We consider a model to be cleaned if the *ASR* of that model *is* $\leq 5\%$.

Table 1: **Key findings** from our experiments: CleanCLIP is much less effective for the model where poison is induced with the stronger objective $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$.

Backdoor	Objective	Poison Induction	Change in Accuracy after Cleaning (Relative) ↑
BadNet	$\mathcal{L}^{pre}_{\text{MMCL}}$	From scratch	1% gain
BadNet	$\mathcal{L}^{pre}_{\text{MMCL}}$	Finetuned from Ckpt	17% loss
Label Consistent	$\mathcal{L}^{pre}_{\text{MMCL}}$	From scratch	10% gain
BadNet	$\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$	From scratch	45% loss
BadNet	$\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$	Finetuned from Ckpt	33% loss
Label Consistent	$\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$	From scratch	16% loss

Table 2: This table shows the original Top-1 zero-shot Imagenet validation set accuracies and remaining accuracy after cleaning the models that were poisoned using BadNet by training from scratch. The cleaning is done using CleanCLIP, i.e., finetuning a poisoned model with $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$. For this table, we choose the models having the highest accuracy and ASR $\leq 5\%$ (successful cleaning). The original ASR values for all models are more than 99%. **Takeaway:** The models trained with $\mathcal{L}^{pre}_{\text{MMCL}}$ maintain their original accuracy after cleaning, while the ones trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ experience a huge drop relative to the starting accuracy (∼20% for model trained on CC3M dataset and 45% for model trained on CC6M dataset) after cleaning.

			Trained with $\mathcal{L}^{pre}_{\text{MMCL}}$		Trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$	
Backdoor	Dataset	Clean Data Size	Orig. Acc.	Clean Acc. (ASR $\leq 5\%$) ↑	Orig. Acc.	Clean Acc. (ASR $\leq 5\%$) ↑
BadNet	CC3M	100K	**16.00%**	16.49%	**17.04%**	14.16%
BadNet	CC6M	100K	**23.76%**	24.04%	**23.86%**	13.05%

4 Experiments

In this section, we expound on the pre-training details for the models, followed by their cleaning procedure and the metrics we use to measure the model's performance.

Metrics The models are evaluated for their Top-1 zero-shot accuracy on the Imagenet-1K validation set (referred to as Imagenet hereafter). Each of the 1,000 classes of Imagenet is described using sentences like: 'a photo of a ...', 'a tattoo of a ...', etc. We generate 80 such text templates for each class (see Appendix C) and then pass them to the text encoder to produce an average text embedding for the class. During zero-shot classification, the prediction for an image is the class whose thus computed text embedding has the highest cosine similarity with the image embedding.

We also evaluate the attack success rate (ASR) of a model. In an apparent similarity to accuracy, the ASR of a backdoored model is defined as the percentage of triggered images that the model classifies as the adversary-chosen target label. For measuring ASR, we add the trigger patch at random locations in all Imagenet validation set images and measure the percentage of them that are classified as the target class, i.e., "banana". We measure both these metrics at the end of each cleaning epoch as any model encountered during the cleaning process is a good candidate for a cleaned model.

Poison Induction by Training from Scratch Table 2 shows the Top-1 zero-shot Imagenet validation set accuracy for the models trained from scratch using $\mathcal{L}^{pre}_{\text{MMCL}}$ and $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ on CC3M and CC6M datasets. For the smaller CC3M dataset, both the models achieve an accuracy of around 16–17%, and for the larger CC6M dataset, the models reach an accuracy of around 24%. *Even though the models trained with $\mathcal{L}^{pre}_{MMCL} + \mathcal{L}^{pre}_{SSL}$ attained higher accuracy than the models trained with $\mathcal{L}^{pre}_{MMCL}$ alone, in order to have better visualization of the difference in performance of CleanCLIP on the two pre-training objectives, we deliberately choose models with similar starting accuracies.* All the models, irrespective of the pre-training objective and the training dataset, reached more than 99% ASR (see Table 3 in Appendix A), implying that poisoning just 0.05% of the dataset is enough to attain very high ASR.

Removing Poison We clean the poisoned model by finetuning it on a 100K, guaranteed to be poison-free, image-text pairs for 20 epochs using a batch size of 128 and AdamW as the optimizer. We perform extensive

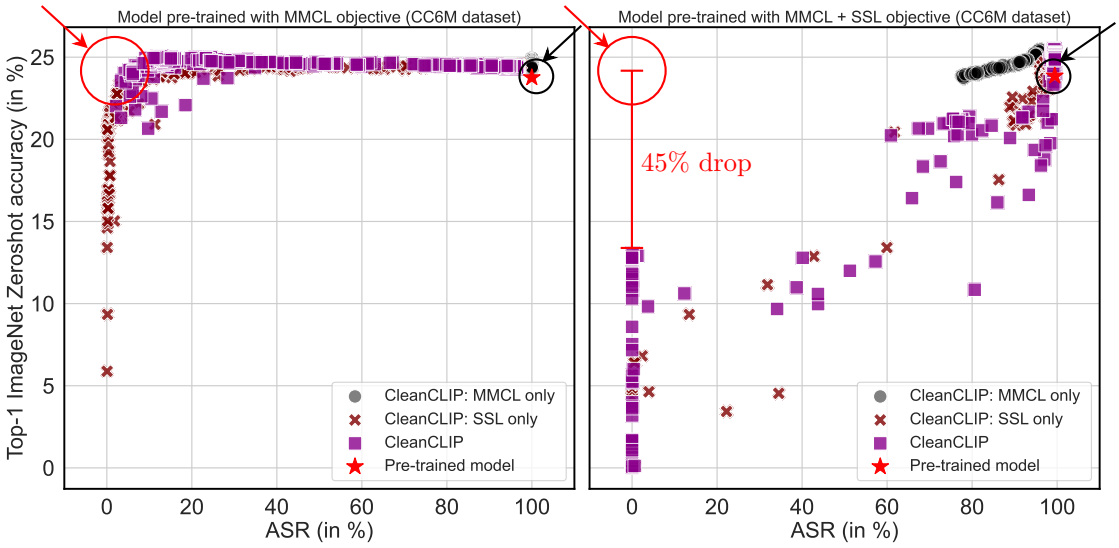


Figure 2: Top-1 zero-shot Imagenet validation set accuracy vs. the ASR, measured at the end of each cleaning epoch for the models trained on the CC6M dataset. The cleaning is done by finetuning the model with the three losses mentioned above. The red star in the top right corner (encircled in the black circle) corresponds to the model's starting accuracy and ASR (before cleaning). For a successful cleaning, there should be models that maintain the model's starting accuracy while having a low ASR (indicated by the red circle's region in the top left). There are several models in the red circle in the left plot (successful clean), while there are no models in the red circle in the right plot (unsuccessful clean). **Takeaway:** CleanCLIP successfully cleans the model trained with $\mathcal{L}^{pre}_{\text{MMCL}}$ (left), while it is ineffective for the models trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ (right).

hyperparameter search and use various learning rates (as many as 8 in some experiments and 14 in others, all with cosine scheduling and 50 warmup steps) for this process. Please refer to Appendix D for the set of learning rates explored for the cleaning procedure. Ideally, after cleaning we would want to obtain a model that maintains the accuracy of the original poisoned model, while getting rid of its poison, i.e., very low ASR. We use three different loss functions for the cleaning process:

1. $\mathcal{L}^{ft}_{\text{MMCL}}$: CleanCLIP showed that finetuning with $\mathcal{L}^{ft}_{\text{MMCL}}$ did not change the original model's accuracy and ASR, and hence is an *ineffective cleaning loss*. We reproduce these results for the models we trained.

2. $\mathcal{L}^{ft}_{\text{SSL}}$: CleanCLIP also showed that finetuning with $\mathcal{L}^{ft}_{\text{SSL}}$ decreased the original model's ASR at the expense of its accuracy, and hence is also an *ineffective cleaning loss*. We also reproduce these results.

3. $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$: CleanCLIP showed that finetuning with a combination of these losses decreased the original model's ASR while not hurting its accuracy, and hence is an *effective cleaning loss*. Our experiments show that while this observation is true for the models trained with $\mathcal{L}^{pre}_{\text{MMCL}}$, however *it does not generalize* to the models trained with stronger pre-training objective $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$. **This is our key finding.**

Findings from the Cleaning Procedure Figure 2 shows the scatter plot of the Top-1 zero-shot Imagenet validation set accuracy and the ASR at the end of each cleaning epoch for the models trained on the CC6M dataset. We defer the plots for the CC3M dataset in Appendix E.1 for space consideration. For both the datasets, we observe that:

1. $\mathcal{L}^{ft}_{\text{MMCL}}$ and $\mathcal{L}^{ft}_{\text{SSL}}$ individually are ineffective cleaning losses as they cause a significant drop in accuracy for lowering the ASR for both the pre-training objectives.

2. $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$ serves as an effective cleaning loss for the model trained with $\mathcal{L}^{pre}_{\text{MMCL}}$ (left plot). The cleaned models maintain the accuracy of the original model, but they have low ASR, which we consider *successful cleaning*. However, it does not lead to an effective cleaning of the model trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ (right plot). Even the model that has the highest accuracy with a low ASR ($\leq 5\%$) is 45% less accurate than the original model, as shown in Figure 2.

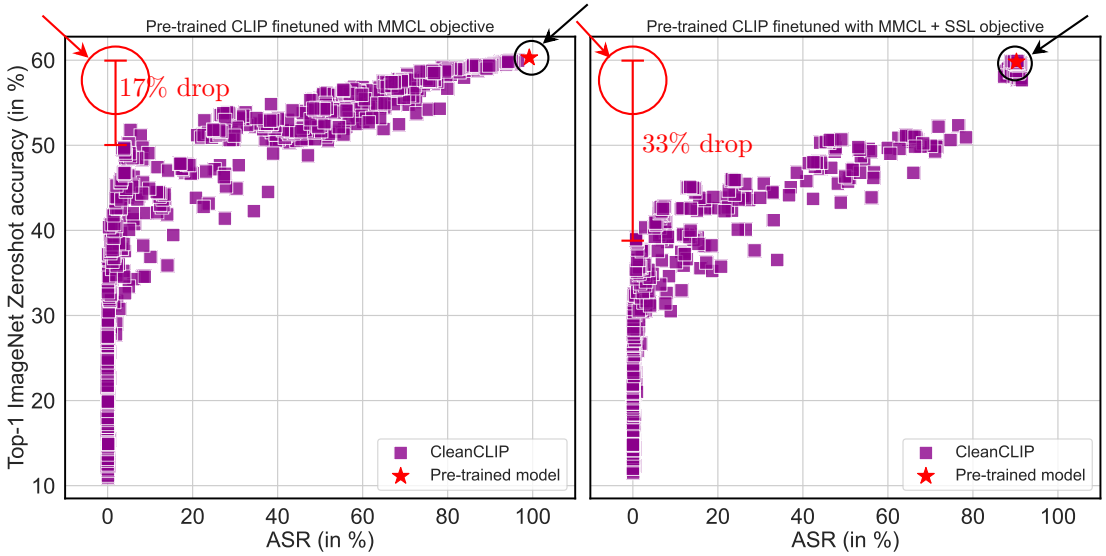


Figure 3: Top-1 zero-shot Imagenet validation set accuracy vs. the ASR, measured at the end of each cleaning epoch for the models poisoned by finetuning a CLIP pre-trained checkpoint on the CC6M dataset. The cleaning is done by finetuning the poisoned model with $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$. The red star in the top right corner (encircled in the black circle) corresponds to the original model's accuracy and ASR (before cleaning). For a successful cleaning, there should be models that maintain the original model's accuracy while having a low ASR (indicated by the red circle in the top left). **Takeaway:** CleanCLIP is unable to successfully clean both the models; however, it performs much worse for the model poisoned with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ (right).

For both datasets, our findings indicate that CleanCLIP is not effective in removing poison from the models trained with a stronger pre-training objective $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$, without a significant drop in accuracy. Table 2 gives the highest accuracy of the models which were successfully cleaned by CleanCLIP (ASR $\leq 5\%$).

Poison Induction by Finetuning a pre-trained model We also induce poison by finetuning a pre-trained CLIP model (Radford et al., 2019). Concretely, we poison two models by finetuning them with $\mathcal{L}^{pre}_{\text{MMCL}}$ and $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ respectively, using the CC6M dataset that had 3000 poisoned datapoints. We use a learning rate of $5e-7$ with AdamW optimizer with 10,000 warmup steps. After poisoning, these models achieve Top-1 zero-shot Imagenet set accuracy of $\sim 60\%$, much higher than the models trained from scratch (Figure 2). The ASR for the model poisoned with $\mathcal{L}^{pre}_{\text{MMCL}}$ is 99% and for the model poisoned with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ is 90%.

Findings from the Cleaning Procedure: We clean the poisoned models using CleanCLIP, i.e., further finetuning on a clean dataset with $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$. Figure 3 shows the scatter plot of the Top-1 zero-shot Imagenet validation set accuracy and the ASR at the end of each finetuning epoch for these two models. In this case, both the models experience a drop in accuracy to obtain a low ASR ($\leq 5\%$); however, the drop is much higher for the model when the poison was induced using $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ (33%, compared to a 17% drop for the model when the poison was induced using $\mathcal{L}^{pre}_{\text{MMCL}}$). This experiment corroborates our previous finding that CleanCLIP is less effective when the poison is induced using a combination of MMCL and SSL, irrespective of the fact whether the poison is induced via finetuning or by training from scratch.

Poisoning a Different Backbone Architecture

For this experiment, we poison a model with a different backbone architecture, specifically ViTs. We poisoned these models by training them on the CC6M dataset with 3000 poisoned datapoints, and cleaned them using CleanCLIP. Figure 4 shows the cleaning plots for these models. We observe that similar to the previous experiments, the model trained with MMCL + SSL experiences a much larger drop in accuracy than the model just trained with MMCL, although in this case the model just trained with MMCL experiences a lot drop in accuracy as well.

Poisoning Induction using a Different Poison Due to space considerations, we present the results for the effectiveness of CleanCLIP when models are poisoned using label consistent backdoors in Appendix F.

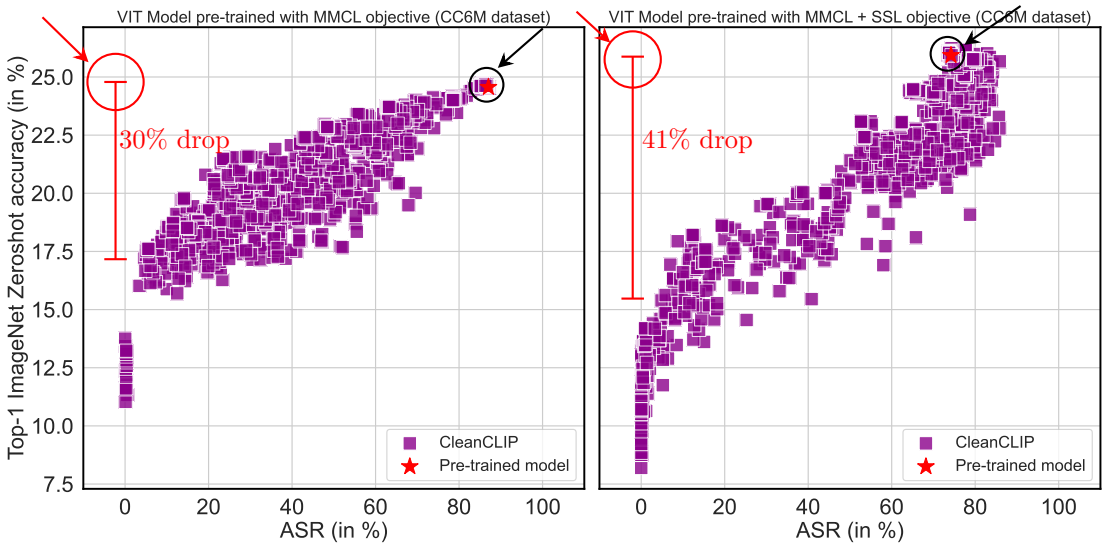


Figure 4: Top-1 zero-shot Imagenet validation set accuracy vs. the ASR, measured at the end of each cleaning epoch for the models trained on the CC6M dataset. The cleaning using CleanCLIP. The red star in the top right corner (encircled in the black circle) corresponds to the model's starting accuracy and ASR (before cleaning). **Takeaway:** When a ViT backbone is poisoned, there are no cleaned models that maintain the original accuracies for both the pre-training losses, however the drop is much larger for the model trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ (right).

We observe that similar to the case of poisoning with BadNet, CleanCLIP is much less effective when the poison is induced using $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ (16% loss in accuracy), compared to the case when it is induced using $\mathcal{L}^{pre}_{\text{MMCL}}$ (10% gain in accuracy).

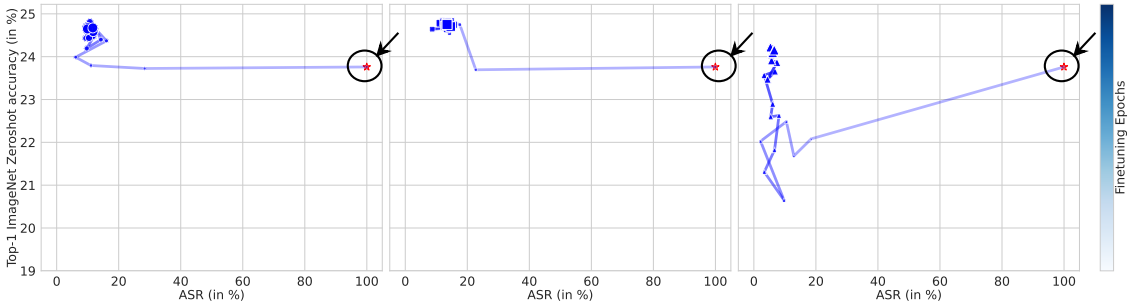
Dependence of the Stopping Criterion on the Pre-training Objective In the previous section, the ability to find a model with high accuracy and low ASR is considered a success for the CleanCLIP approach. However, in practice, one would not be aware of the ASR of the model being cleaned and, therefore, would not know when to stop the cleaning process. To highlight this practicality concern, we show the multiple cleaning trajectories for models trained using different pre-training objectives in Figure 5.

Figure 5a shows the trajectories of three cleaning runs with different learning rates for a model trained using $\mathcal{L}^{pre}_{\text{MMCL}}$ on the CC6M dataset. We observe that in all the three runs, the trajectory converges to a region of high accuracy and low ASR (top left corner), and the trajectories are smooth. This indicates that a practitioner can clean this model by finetuning the poisoned model for as long as their resources allow, and choose the model at the end of the finetuning process. They will likely obtain a model with high accuracy and low ASR, i.e., a successfully cleaned model.

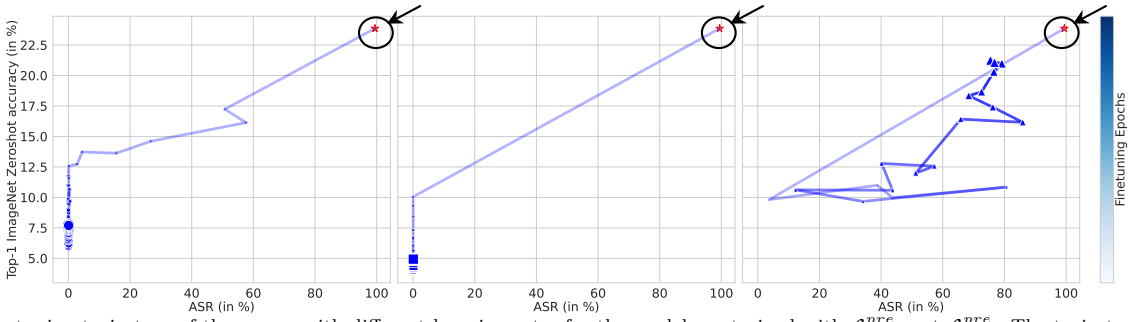
Figure 5b shows the trajectories of three cleaning runs with different learning rates for a model trained using $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ on the CC6M dataset. We observe that in all three runs, the successfully cleaned model (high accuracy with a low ASR ($\leq 5\%$)) is an intermediate model in the trajectory, *and not* the model at the end of the process. With continued finetuning, the model can both lose accuracy and gain ASR, both of which are undesirable. Therefore, it is difficult for a practitioner to discern when to stop the finetuning process to obtain a clean model.

Therefore, the practicality of using CleanCLIP also depends on the pre-training objective of the model. Appendix K shows the cleaning trajectories for all explored hyperparameters.

Dependence of CleanCLIP on the Ideal Condition of the Dataset CleanCLIP assumes that the cleaning data is entirely free of poisoned datapoints. In practice, this assumption can be violated even when considerable care is taken to ensure it. To simulate this real-world situation, we clean models using data with a few poisoned datapoints, specifically 5 and 10 poisoned datapoints in the 100K cleaning datapoints. Note that these datasets are still, respectively, 10× and 5× cleaner than the original training dataset, illustrating a situation where the cleaning data is much cleaner than the training dataset but still not perfect.



(a) Finetuning trajectory of three runs with different learning rates for the model pre-trained with $\mathcal{L}^{pre}_{\text{MMCL}}$. The trajectories are smooth and end up in high accuracy and low ASR regime at the end of finetuning process.



(b) Finetuning trajectory of three runs with different learning rates for the model pre-trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$. The trajectories are not smooth and increased finetuning can lead to both decreased accuracy and higher ASR.

Figure 5: Finetuning trajectories of models with different pre-training objectives. Successive finetuning epochs are shown with increasing size of the markers and intensity of the connecting line. The red star in the top right corner (encircled in the black circle) corresponds to the original model's accuracy and ASR. **Takeaway:** Models trained with $\mathcal{L}^{pre}_{\text{MMCL}}$ converge to a region of high accuracy and low ASR as we continue to finetune. On the other hand, models trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ fail to converge to a region of high accuracy and low ASR, and continued finetuning can lead to both decreased accuracy and higher ASR. This makes determining the stopping criterion for the cleaning process for the latter models challenging.

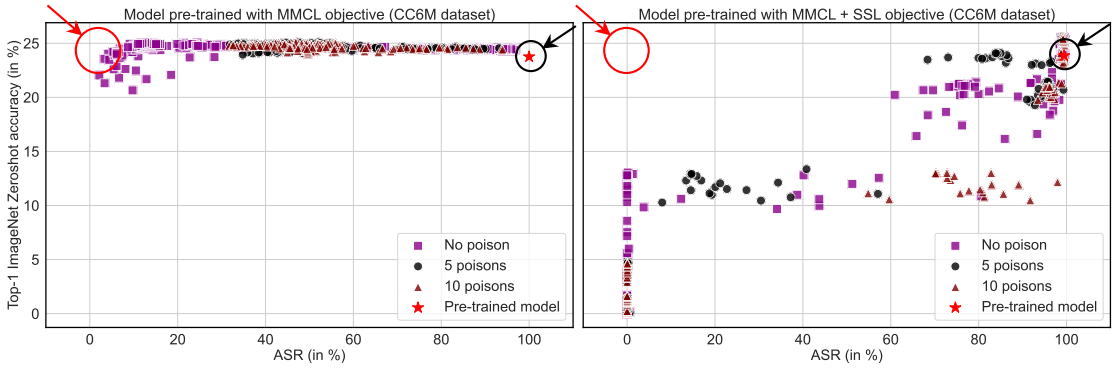


Figure 6: Top-1 zero-shot Imagenet validation set accuracy v/s the ASR during the cleaning process for the two models. The finetuning is done with $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$. We measure accuracy and ASR at the end of each epoch. The red star in the top right corner (encircled in the black circle) corresponds to the original model's accuracy and ASR. For a successful cleaning, there should be models that maintain the original model's accuracy while having a low ASR (indicated by the red circle). **Takeaway:** Even having 5 poisons in the cleaning dataset (i.e. 0.005% of the dataset, which is 10× cleaner than the pre-training data) hurts the cleaning process for both pre-training objectives, and $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ trained models are hurt worse.

Figure 6 shows the scatter plot of the Top-1 zero-shot Imagenet validation set accuracy and the ASR at the end of each cleaning epoch when two models trained from scratch on the CC6M dataset, one using $\mathcal{L}^{pre}_{\text{MMCL}}$ and the other using $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ is cleaned by finetuning on this slightly poisoned dataset with $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$. We observe that having just 5 poisoned datapoints in the cleaning dataset severely weakens CleanCLIP for both the pre-training objectives.

For models pre-trained with $\mathcal{L}^{pre}_{\text{MMCL}}$, we found cleaned models that maintain the original model's accuracy and achieve around 30-50% ASR. On the other hand, for the models pre-trained with the stronger objective $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$, having just 5 poisoned examples renders the cleaning procedure completely ineffective. The models pre-trained on the CC6M dataset lose about 80% of the original model's accuracy to obtain a low ASR ($\leq 5\%$), and no model has an ASR lower than 90% for the CC3M pre-trained model (Figure 11).

Takeaways Our experiments highlight the fact that a stronger pre-training objective, like the combination of MMCL and SSL, also affects the strength of poison induction, making the cleaning process difficult. Also, for a practitioner, when pre-training with a stronger objective, the decision of when to stop finetuning becomes non-trivial, as we show that the model at the end of the cleaning procedure is usually not the one with the lowest ASR and the best accuracy. The situation is further exacerbated when we even slightly relax the assumption of 100% poison-free cleaning data, which can be too stringent in practice.

5 Analysis of the Stronger Pre-training Objective

We now perform several analysis experiments to understand the reason behind the difference in the poison removal ability of CleanCLIP across the two pre-training objectives. We also experiment with other methods to attempt to remove the poison when induced using $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$.

Cleaning using an Objective distinct from Pre-training CleanCLIP successfully cleans the models trained with $\mathcal{L}^{pre}_{\text{MMCL}}$ by finetuning with $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$. However, it was unsuccessful for the model trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$. A plausible reason for this behavior could be that we are using the same pre-training and cleaning objective in the latter case, and CleanCLIP might be able to successfully clean the latter models if we were to clean it with a loss objective that is distinct from its pre-training objective. To test this hypothesis, we clean the models trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ by finetuning with $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}} + \mathcal{L}^{ft}_{\text{DeepClust}}$, where $\mathcal{L}^{ft}_{\text{DeepClust}}$ is an additional deep clustering objective (Caron et al., 2018) on the vision encoder.

In deep clustering, we first obtain a pseudo-label for each image. We obtain the pseudo-label for an image in two ways: **a)** by classifying each image into one of the 1,000 Imagenet classes using powerful models such as SigLIP ViT-L/14 (zero-shot Imagenet accuracy of 83.08%) (Zhai et al., 2023), and **b)** performing a 1,000-way clustering on feature space of our trained vision encoder, using FAISS (Johnson et al., 2019). Note that the use of SigLIP ViT-L/14 for obtaining pseudo-labels is a *cheating experiment* for a deep clustering task (we don't have access to ground truth labels and therefore use SigLIP ViT-L/14 for this task); however, this experiment is solely performed to probe the upper bound of the poison removal that can be obtained when using deep clustering approaches.

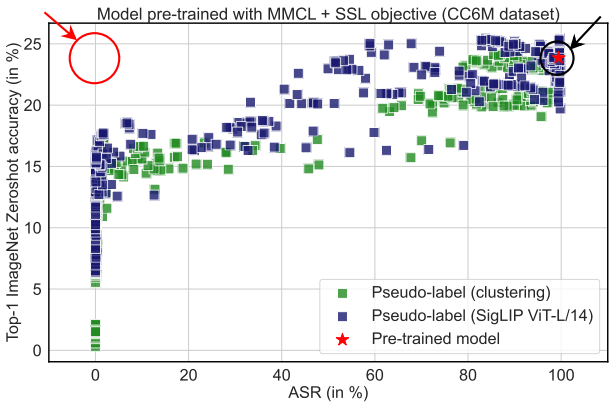


Figure 7: Top-1 zero-shot Imagenet validation set accuracy vs. the ASR during the cleaning process for the $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ pre-trained model on the CC6M dataset. The finetuning is done with $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}} + \mathcal{L}^{ft}_{\text{DeepClust}}$. **Takeaway:** Adding $\mathcal{L}^{ft}_{\text{DeepClust}}$ is unable to successfully remove the poison from models pre-trained using the strong objective, indicating that having distinct pre-training and cleaning objectives does not ensure removal of poison.

In the latter case, when we use clustering-based pseudo-label assignment for every image in the cleaning dataset, we learn to predict the assigned pseudo-label ($\hat{y}$) with the help of a linear classifier on top of the vision encoder using cross-entropy loss ($\mathcal{L}_{\text{Xent}}$). Let $W \in \mathbb{R}^{d \times 1000}$ be the linear classifier mapping visual features ($\mathbb{R}^d$) to one of the 1000 pseudo-labels. For a given datapoint (I_j, T_j) with assigned pseudo-label $\hat{y}_j$, deep clustering's objective becomes

$$\mathcal{L}^{ft}_{\text{DeepClust}}(f_I, W; I_j, \hat{y}_j) = \mathcal{L}_{\text{Xent}}(W^T f_I(I_j); \hat{y}_j), \tag{3}$$

and therefore overall objective becomes $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}} + \mathcal{L}^{ft}_{\text{DeepClust}}$. Following Caron et al. (2018), we re-initialize classifier head W every time we re-compute pseudo-labels. We finetune the model trained using $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ on the CC6M dataset for 10 epochs using 8 different learning rates for both the clustering techniques (see Appendix D for hyperparameter details) and measure the Top-1 zero-shot Imagenet accuracy and ASR at the end of each finetuning epoch.

Figure 7 shows the scatter plot of the Top-1 zero-shot Imagenet validation set accuracy and the ASR for this experiment. We observe that adding deep clustering does not successfully remove the poison from the model. This experiment shows that the ineffectiveness of CleanCLIP in cleaning the model trained with the stronger objective is not just due to the same pre-training and cleaning objectives but due to stronger poison induction. Had the difference between the pre-training and finetuning objectives been the reason for CleanCLIP's success, then this DeepClustering experiment could have successfully cleaned the model.

Poison Removal using Heavy Regularization In this section, we attempt to remove the poison induced using $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ with heavy regularization. Zhu et al. (2023) proposed heavy regularization as an approach to remove backdoors from vision and language models. To evaluate the efficacy of this approach, we finetune these models with 9 different regularization weights for 10 epochs using 8 different learning rates for each regularization weight on 100K image-text pairs (see Appendix D for hyperparameter details). The regularization loss is added to three different loss objectives during the finetuning process: a) $\mathcal{L}^{ft}_{\text{MMCL}}, b)\mathcal{L}^{ft}_{\text{SSL}}$, and c) $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$.

Figure 8 shows the scatter plot of the Top-1 zero-shot Imagenet validation set accuracy and the ASR at the end of each finetuning epoch. We observe that no hyperparameter combination can successfully remove the poison, which shows that simply adding regularization cannot remove the strongly induced poison.

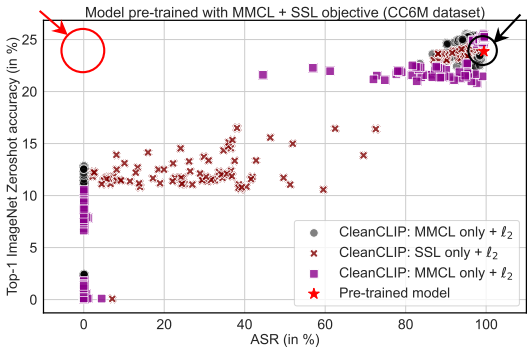


Figure 8: Top-1 zero-shot Imagenet validation set accuracy vs. the ASR during the cleaning process. **Takeaway:** Using regularization with any hyperparameter combinations is unable to remove the poison from models trained using $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$.

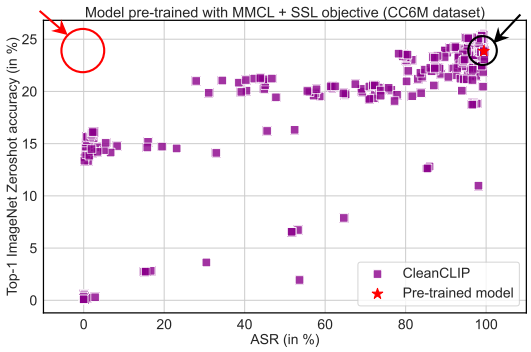


Figure 9: Top-1 zero-shot Imagenet validation set accuracy vs. the ASR during the cleaning process post shrinking and perturbing the weights. **Takeaway:** The shrink and perturb technique is unable to remove poison from the model that is trained using $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$.

Poison Removal using Shrink and Perturb In this section, we attempt to remove the poison using the shrink and perturb technique. Ash & Adams (2020) proposed shrink and perturb technique to improve the accuracy of a model when finetuned for a task. CleanCLIP also cleans a model by finetuning, and therefore, it could benefit from this technique. In this technique, a small noise is added to the model weights before starting finetuning, $\theta_0 \leftarrow \lambda\theta_0 + p_0$, where $p \sim \mathcal{N}\left(0, \sigma^2\right)$ and $\lambda \in (0, 1)$. To evaluate whether this approach

can help in removing the poison from a model trained using $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$, we add a small noise to all its weights. While the choice of noise scale (σ) and shrinking parameter (λ) is a hyperparameter, we experiment with 5 values of σ for 15 values of λ. To determine which model to clean amongst the 75 possible models obtained from shrink and perturb described above, we measure the Top-1 zero-shot Imagenet validation set accuracy and the ASR of the models with noised weights, and selected model with the highest accuracy whose ASR was lower than 15% for cleaning. We clean the selected model by finetuning it using 8 learning rates for 20 epochs on 100K image-text pairs. Figure 9 shows the accuracy and ASR at the end of each finetuning epoch. We observe that even this approach is unable to successfully remove the poison from the models.

Ablation on Hyperparameters We also perform ablation experiments to see the impact of hyperparameters on the ability of CleanCLIP to remove poison from the $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ pre-trained model. In particular, we study the impact of the size of the cleaning dataset, the number of finetuning epochs, and the weightage of $\mathcal{L}^{ft}_{\text{SSL}}$. Appendix G reports the metrics when the finetuning is done on a dataset that is twice the size of the cleaning dataset used in the previous section. Appendix H reports the metrics when the finetuning is done for up to 100 epochs, i.e., 5× longer than in the previous section. Appendix I reports the accuracy and ASR metrics when the finetuning is done with higher weight to the $\mathcal{L}^{ft}_{\text{SSL}}$ loss. While changing the hyperparameters helps improve the Pareto-frontier curve, none of the three ablations can successfully remove the strongly induced poison.

Takeaways These experiments demonstrate that making the pre-training and finetuning objectives distinct from each other and perturbation techniques like heavy regularization (Zhu et al., 2023) and shrink-and-perturb (Ash & Adams, 2020) are not enough to remove the strongly induced poison. Moreover, we observe that while doing a more fine-grained search on different hyperparameters helps slightly improve the Pareto-frontier curve, none of the hyperparameter ablations can successfully remove the poison. Since removing the poison from such a model is an open research problem, a practitioner who is engaged in training models using web-curated data should consider training their model with only $\mathcal{L}^{pre}_{\text{MMCL}}$ so that they can clean their model on a carefully curated image-text pair dataset to remove potential backdoors from it, even when there are a few poisoned examples in the finetuning dataset.

6 Conclusions

Through our extensive hyperparameter search and ablation experiments, we unveil a critical limitation of the current state-of-the-art poison mitigation technique for multimodal models, CleanCLIP. It fails to effectively remove backdoor poisoning when a model is trained using stronger objectives like the combination of multimodal contrastive learning (MMCL) and intramodal self-supervised learning (SSL). This objective is common in popular approaches like SLIP (Mu et al., 2022) and has demonstrated superior accuracy over training with only the MMCL objective. Our experiments show that this vulnerability persists irrespective of the scale of the pre-training and the cleaning datasets, irrespective of the manner of poison induction (from scratch or by finetuning), and irrespective of the specific backdoor attack.

Particularly concerning is the unstable cleaning trajectory in models trained using the stronger objective (Figure 5b). Often unaware of the specific backdoor attack, practitioners face challenges in determining the optimal point to halt the cleaning process. This instability can lead to suboptimal models, as continued finetuning can decrease accuracy and increase attack success rate (ASR). Furthermore, our findings highlight the critical assumption of a completely poison-free cleaning dataset for CleanCLIP's effectiveness, an assumption that is rarely met in practical scenarios. This becomes particularly problematic with the use of stronger pre-training objectives. The models trained with the simpler MMCL objective evade both these issues by having stable cleaning trajectories and amenability to poison reduction even under non-ideal conditions.

Given these insights, we urge practitioners to consider training their models using the simpler MMCL objective. Even though this might slightly hurt the accuracy, it significantly enhances its amenability to remove backdoors. Our recommendation would also circumvent the issue of knowing when to halt the cleaning procedure, as more finetuning epochs would not hurt the model's accuracy and ASR. Further, it will also be beneficial when cleaning data is not entirely poison-free. Our work underscores the formidable challenge of defending models against backdoor attacks, an open research problem. We encourage future defense methods

to robustly test their approach against various pre-training objectives and we invite the community to develop robust defense methods.

References

Jordan Ash and Ryan P Adams. On warm-starting neural network training. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3884–3894. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/288cd2567953f06e460a33951f55daaf-Paper.pdf.

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning, 2023.

Hritik Bansal, Nishad Singhi, Yu Yang, Fan Yin, Aditya Grover, and Kai-Wei Chang. Cleanclip: Mitigating data poisoning attacks in multimodal contrastive learning. *arXiv preprint arXiv:2303.03323*, 2023.

Eitan Borgnia, Valeriia Cherepanova, Liam Fowl, Amin Ghiasi, Jonas Geiping, Micah Goldblum, Tom Goldstein, and Arjun Gupta. Strong data augmentation sanitizes poisoning and backdoor attacks without an accuracy tradeoff. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3855–3859. IEEE, 2021.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

Nicholas Carlini and Andreas Terzis. Poisoning and backdooring contrastive learning. *arXiv preprint arXiv:2106.09667*, 2021.

Nicholas Carlini, Matthew Jagielski, Christopher A Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. Poisoning web-scale training datasets is practical. *arXiv preprint arXiv:2302.10149*, 2023.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 132–149, 2018.

Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *CVPR*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pp. 539–546. IEEE, 2005.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9588–9597, 2021.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu,

Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets, 2023.

Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pp. 113–125, 2019.

Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan Rossi, Vishwa Vinay, and Aditya Grover. Cyclip: Cyclic contrastive language-image pretraining. *Advances in Neural Information Processing Systems*, 35:6704–6719, 2022.

Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses, 2021.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pp. 1735–1742. IEEE, 2006.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Lixuan Zhu, Samyak Parajuli, Mike Guo, Dawn Xiaodong Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8320–8329, 2020. URL https://api.semanticscholar.org/CorpusID:220250257.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021.

Alvi Md Ishmam and Christopher Thomas. Semantic shield: Defending vision-language models against backdooring and poisoning via fine-grained knowledge alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24820–24830, June 2024.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pp. 4904–4916. PMLR, 2021a.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, 2021b. URL https://api.semanticscholar.org/CorpusID:231879586.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Janghyeon Lee, Jongsuk Kim, Hyounguk Shon, Bumsoo Kim, Seung Hwan Kim, Honglak Lee, and Junmo Kim. Uniclip: Unified framework for contrastive language-image pre-training. *Advances in Neural Information Processing Systems*, 35:1008–1019, 2022.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*, 2021a.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *NeurIPS*, 2021b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017. URL https://api.semanticscholar.org/CorpusID:53592270.

Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *European Conference on Computer Vision*, pp. 529–544. Springer, 2022.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. URL https://api.semanticscholar.org/CorpusID:160025533.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, 2019. URL https://api.semanticscholar.org/CorpusID:67855879.

Aniruddha Saha, Ajinkya Tejankar, Soroush Abbasi Koohpayegani, and Hamed Pirsiavash. Backdoor attacks on self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13337–13346, 2022.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018.

Hossein Souri, Liam Fowl, Rama Chellappa, Micah Goldblum, and Tom Goldstein. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. *Advances in Neural Information Processing Systems*, 35:19165–19178, 2022.

Jacob Steinhardt, Pang Wei W Koh, and Percy S Liang. Certified defenses for data poisoning attacks. *Advances in neural information processing systems*, 30, 2017.

Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Clean-label backdoor attacks, 2019. URL https://openreview.net/forum?id=HJg6e2CcK7.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 707–723. IEEE, 2019.

Wenhan Yang, Jingdong Gao, and Baharan Mirzasoleiman. Better safe than sorry: Pre-training clip against targeted data poisoning and backdoor attacks, 2023.

Wenhan Yang, Jingdong Gao, and Baharan Mirzasoleiman. Robust contrastive language-image pre-training against data poisoning and backdoor attacks. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2024. Curran Associates Inc.

Yu Yang, Tian Yu Liu, and Baharan Mirzasoleiman. Not all poisons are created equal: Robust training against data poisoning. In *International Conference on Machine Learning*, pp. 25154–25165. PMLR, 2022.

Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. Filip: Fine-grained interactive language-image pre-training. *arXiv preprint arXiv:2111.07783*, 2021.

Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning, 2022.

Kota Yoshida and Takeshi Fujino. Disabling backdoor and identifying poison data by using knowledge distillation in backdoor attacks on deep neural networks. In *Proceedings of the 13th ACM Workshop on Artificial Intelligence and Security*, pp. 117–127, 2020.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. *arXiv preprint arXiv:2303.15343*, 2023.

Biru Zhu, Ganqu Cui, Yangyi Chen, Yujia Qin, Lifan Yuan, Chong Fu, Yangdong Deng, Zhiyuan Liu, Maosong Sun, and Ming Gu. Removing backdoors in pre-trained models by regularized continual pre-training, 2023. URL https://openreview.net/forum?id=JKuBOuzntQ.

Table 3: This table shows the original Top-1 zero-shot Imagenet validation set accuracy and ASR for the models trained with $\mathcal{L}^{pre}_{\text{MMCL}}$ and $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ on the CC3M and CC6M datasets. These models are the input to CleanCLIP approach to remove their poison. **Takeaway:** The models trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ achieve higher accuracy than their counterparts trained with $\mathcal{L}^{pre}_{\text{MMCL}}$ solely (except when poisoned by finetuning). Adding BadNet poison to a mere 0.05% of the training data achieves almost a 100% ASR when trained from scratch and almost 90% ASR when induced by finetuning.

Poison Induction	Backdoor	Objective	Dataset			
			CC3M		CC6M	
			Accuracy (↑)	ASR (↓)	Accuracy (↑)	ASR (↓)
From Scratch	BadNet	$\mathcal{L}^{pre}_{\text{MMCL}}$	16.00%	99.88%	23.76%	99.98%
From Scratch	BadNet	$\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$	17.04%	99.03%	23.86%	99.45%
From Scratch	Label-Consistent	$\mathcal{L}^{pre}_{\text{MMCL}}$	–	–	22.96%	88.27%
From Scratch	Label-Consistent	$\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$	–	–	23.09%	88.01%
Finetuning from Ckpt	BadNet	$\mathcal{L}^{pre}_{\text{MMCL}}$	–	–	60.30%	99.20%
Finetuning from Ckpt	BadNet	$\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$	–	–	59.81%	90.24%

A Pre-Training Details

We train all the models on 8 Nvidia A100 GPUs for 64 epochs. We use an initial learning rate of 0.001 for the models trained from scratch, and for the models where poison is induced by finetuning from a pre-trained checkpoint, we use an initial learning rate of $5e-7$. We use cosine scheduling and 10000 warmup steps with AdamW optimizer (Loshchilov & Hutter, 2017) for training. The model trained with $\mathcal{L}^{pre}_{\text{MMCL}}$ uses a batch size of 256, whereas the model trained with the $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ uses a batch size of 128.

We show the change in accuracy and ASR with the training epochs for the model trained from scratch with BadNet attack using $\mathcal{L}^{pre}_{\text{MMCL}}$ in Figure 17 and using $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ in Figure 18. We use early-stopping for the model trained with $\mathcal{L}^{pre}_{\text{MMCL}}$ and choose the model with the highest accuracy. For $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ pre-training, we choose the model that has the closest accuracy to the $\mathcal{L}^{pre}_{\text{MMCL}}$ trained model. Table 3 shows the accuracy and the ASR for all the models we select in this paper for poison removal.

B Performance of the Pre-trained Models

Table 3 shows the Top-1 zero-shot Imagenet Validation set accuracy and the ASR of models that were selected for both the pre-training objectives.

C Templates for Text-Embedding Computation

```
'a bad photo of a {class}.', 'a photo of many {class}.', 'a sculpture of a {class
}.', 'a photo of the hard to see {class}.', 'a low resolution photo of the {class
}.', 'a rendering of a {class}.', 'graffiti of a {class}.', 'a bad photo of the {
class}.', 'a cropped photo of the {class}.', 'a tattoo of a {class}.', 'the
embroidered {class}.', 'a photo of a hard to see {class}.', 'a bright photo of a {
class}.', 'a photo of a clean {class}.', 'a photo of a dirty {class}.', 'a dark
photo of the {class}.', 'a drawing of a {class}.', 'a photo of my {class}.', 'the
plastic {class}.', 'a photo of the cool {class}.', 'a close-up photo of a {class
}.', 'a black and white photo of the {class}.', 'a painting of the {class}.', 'a
painting of a {class}.', 'a pixelated photo of the {class}.', 'a sculpture of the
{class}.', 'a bright photo of the {class}.', 'a cropped photo of a {class}.', 'a
plastic {class}.', 'a photo of the dirty {class}.', 'a jpeg corrupted photo of a {
class}.', 'a blurry photo of the {class}.', 'a photo of the {class}.', 'a good
photo of the {class}.', 'a rendering of the {class}.', 'a {class} in a video game
.', 'a photo of one {class}.', 'a doodle of a {class}.', 'a close-up photo of the
```

```
{class}.', 'a photo of a {class}.', 'the origami {class}.', 'the {class} in a
video game.', 'a sketch of a {class}.', 'a doodle of the {class}.', 'a origami {
class}.', 'a low resolution photo of a {class}.', 'the toy {class}.', 'a rendition
 of the {class}.', 'a photo of the clean {class}.', 'a photo of a large {class}.',
 'a rendition of a {class}.', 'a photo of a nice {class}.', 'a photo of a weird {
class}.', 'a blurry photo of a {class}.', 'a cartoon {class}.', 'art of a {class
}.', 'a sketch of the {class}.', 'a embroidered {class}.', 'a pixelated photo of a
 {class}.', 'itap of the {class}.', 'a jpeg corrupted photo of the {class}.', 'a
good photo of a {class}.', 'a plushie {class}.', 'a photo of the nice {class}.', '
a photo of the small {class}.', 'a photo of the weird {class}.', 'the cartoon {
class}.', 'art of the {class}.', 'a drawing of the {class}.', 'a photo of the
large {class}.', 'a black and white photo of a {class}.', 'the plushie {class}.',
'a dark photo of a {class}.', 'itap of a {class}.', 'graffiti of the {class}.', 'a
 toy {class}.', 'itap of my {class}.', 'a photo of a cool {class}.', 'a photo of a
 small {class}.', 'a tattoo of the {class}.'
```

D Hyperparameter Details

In this section we provide details of the hyperparameters we used for the cleaning experiments.

D.1 Cleaning of the Model Pre-trained on CC3M dataset using MMCL

Cleaning Epochs: 20
Learning rates (13 values): {1e-5, 4e-5, 8e-5, 1e-4, 1.5e-4, 2e-4, 2.5e-4, 3e-4, 3.5e-4, 4e-4, 8e-4, 1e-3, 4e-3}
MMCL weight: 1
SSL weight: 1
Size of the Cleaning Dataset: 1,00,000

D.2 Cleaning of the Model Pre-trained on CC3M dataset using MMCL and SSL

Cleaning Epochs: 20
Learning rates (13 values): {1e-5, 4e-5, 8e-5, 1e-4, 1.5e-4, 2e-4, 2.5e-4, 3e-4, 3.5e-4, 4e-4, 8e-4, 1e-3, 4e-3}
MMCL weight: 1
SSL weight: 1
Size of the Cleaning Dataset: 1,00,000

D.3 Cleaning of the Model Pre-trained on CC6M dataset using MMCL

Cleaning Epochs: 20
Learning rates (12 values): {1e-9, 5e-9, 1e-8, 5e-8, 1e-7, 3e-7, 7e-7, 1e-6, 3e-6, 7e-6, 1e-5, 3e-5}
MMCL weight: 1
SSL weight: 1
Size of the Cleaning Dataset: 1,00,000

D.4 Cleaning of the Model Pre-trained on CC6M dataset using MMCL and SSL

Cleaning Epochs: 20
Learning rates (23 values): {1e-9, 5e-9, 1e-8, 5e-8, 1e-7, 3e-7, 7e-7, 1e-6, 3e-6, 7e-6, 1e-5, 4e-5, 5e-5, 6e-5, 7e-5, 9e-5, 1e-4, 3e-4, 4e-4, 5e-4, 6e-4, 1e-3, 3e-3}
MMCL weight: 1
SSL weight: 1
Size of the Cleaning Dataset: 1,00,000

D.5 Cleaning of the Model where Poison is induced via Finetuning with MMCL on CC6M dataset

Cleaning Epochs: 20
Learning rates (69 values): {5e-05, 4.9e-05, 4.8e-05, 4.7e-05, 4.6e-05, 4.5e-05, 4.4e-05, 4.3e-05, 4.25e-05, 4.2e-05, 4.1e-05, 4e-05, 3.9e-05, 3.8e-05, 3.75e-05, 3.7e-05, 3.6e-05, 3.5e-05, 3.4e-05, 3.3e-05, 3.2e-05, 3.1e-05, 3e-05, 2.9e-05, 2.8e-05, 2.7e-05, 2.6e-05, 2.5e-05, 2.4e-05, 2.3e-05, 2.2e-05, 2.1e-05, 2e-05, 1.9e-05, 1.8e-05, 1.7e-05, 1.6e-05, 1.5e-05, 1.4e-05, 1.3e-05, 1.2e-05, 1.1e-05, 1e-05, 9e-06, 8e-06, 7e-06, 6e-06, 5e-06, 4.9e-06, 4.75e-06, 4.5e-06, 4.25e-06, 4e-06, 3.75e-06, 3.5e-06, 3.25e-06, 3e-06, 2.75e-06, 2.5e-06, 2.25e-06, 2e-06, 1.75e-06, 1.5e-06, 1.25e-06, 1e-06, 5e-07, 1e-07, 5e-08, 1e-08}
MMCL weight: 1
SSL weight: 1
Size of the Cleaning Dataset: 1,00,000

D.6 Cleaning of the Model where Poison is induced via Finetuning with MMCL + SSL on CC6M dataset

Cleaning Epochs: 20
Learning rates (85 values): {0.005, 0.001, 0.0005, 0.0001, 5e-05, 5e-05, 4.9e-05, 4.8e-05, 4.75e-05, 4.7e-05, 4.6e-05, 4.5e-05, 4.5e-05, 4.4e-05, 4.3e-05, 4.25e-05, 4.2e-05, 4.1e-05, 4e-05, 4e-05, 3.9e-05, 3.8e-05, 3.75e-05, 3.7e-05, 3.6e-05, 3.5e-05, 3.5e-05, 3.4e-05, 3.3e-05, 3.25e-05, 3.2e-05, 3.1e-05, 3e-05, 3e-05, 2.9e-05, 2.8e-05, 2.75e-05, 2.7e-05, 2.6e-05, 2.5e-05, 2.5e-05, 2.4e-05, 2.3e-05, 2.25e-05, 2.2e-05, 2.1e-05, 2e-05, 2e-05, 1.9e-05, 1.8e-05, 1.75e-05, 1.7e-05, 1.6e-05, 1.5e-05, 1.5e-05, 1.4e-05, 1.4e-05, 1.3e-05, 1.3e-05, 1.2e-05, 1.2e-05, 1.1e-05, 1.1e-05, 1e-05, 1e-05, 9e-06, 9e-06, 8e-06, 8e-06, 7e-06, 7e-06, 6e-06, 6e-06, 5e-06, 5e-06, 1e-06, 1e-06, 5e-07, 5e-07, 1e-07, 1e-07, 5e-08, 5e-08, 1e-08, 1e-08}
MMCL weight: 1
SSL weight: 1
Size of the Cleaning Dataset: 1,00,000

D.7 Cleaning of the Model with Label Consistent Poisoning trained with MMCL on CC6M dataset

Cleaning Epochs: 20
Learning rates (44 values): {5e-05, 4.75e-05, 4.25e-05, 4e-05, 3.8e-05, 3.75e-05, 3.7e-05, 3.6e-05, 3.5e-05, 3.4e-05, 3.3e-05, 3.2e-05, 3.1e-05, 3e-05, 2.9e-05, 2.8e-05, 2.7e-05, 2.6e-05, 2.5e-05, 2.4e-05, 2.3e-05, 2.2e-05, 2.1e-05, 2e-05, 1.9e-05, 1.8e-05, 1.7e-05, 1.6e-05, 1.5e-05, 1.4e-05, 1.3e-05, 1.2e-05, 1.1e-05, 1e-05, 9e-06, 8e-06, 7e-06, 6e-06, 5e-06, 1e-06, 5e-07, 1e-07, 5e-08, 1e-08}
MMCL weight: 1
SSL weight: 1
Size of the Cleaning Dataset: 1,00,000

D.8 Cleaning of the Model with Label Consistent Poisoning trained with MMCL + SSL on CC6M dataset

Cleaning Epochs: 20
Learning rates (71 values): {0.007, 0.006, 0.005, 0.004, 0.003, 0.002, 0.001, 0.0009, 0.0008, 0.0007, 0.0006, 0.0005, 0.0004, 0.0003, 0.0002, 0.00018, 0.00017, 0.00016, 0.00014, 0.00013, 0.00012, 0.00011, 0.0001, 9e-05, 8e-05, 7e-05, 6e-05, 5e-05, 4.75e-05, 4.25e-05, 4e-05, 3.8e-05, 3.75e-05, 3.7e-05, 3.6e-05, 3.5e-05, 3.4e-05, 3.3e-05, 3.2e-05, 3.1e-05, 3e-05, 2.9e-05, 2.8e-05, 2.7e-05, 2.6e-05, 2.5e-05, 2.4e-05, 2.3e-05, 2.2e-05, 2.1e-05, 2e-05, 1.9e-05, 1.8e-05, 1.7e-05, 1.6e-05, 1.5e-05, 1.4e-05, 1.3e-05, 1.2e-05, 1.1e-05, 1e-05, 9e-06, 8e-06, 7e-06, 6e-06, 5e-06, 1e-06, 5e-07, 1e-07, 5e-08, 1e-08}
MMCL weight: 1
SSL weight: 1
Size of the Cleaning Dataset: 1,00,000

D.9 Cleaning of the Model Pre-trained on the CC3M dataset under Non-Ideal Conditions

Cleaning Epochs: 20
Learning rates (8 values): {1e-7, 3e-7, 7e-7, 1e-6, 3e-6, 7e-6, 1e-5, 3e-5}
MMCL weight: 1
SSL weight: 1
Size of the Cleaning Dataset: 1,00,000

D.10 Cleaning of the Model Pre-trained on the CC6M dataset under Non-Ideal Conditions

Cleaning Epochs: 20
Learning rates (19 values): {1e-8, 5e-8, 1e-7, 3e-7, 5e-7, 7e-7, 1e-6, 3e-6, 5e-6, 7e-6, 1e-5, 3e-5, 5e-5, 1e-4, 3e-4, 5e-4, 7e-4, 1e-4, 3e-4}
MMCL weight: 1
SSL weight: 1
Size of the Cleaning Dataset: 1,00,000

D.11 Cleaning of the Model Pre-trained on the CC6M dataset using an Objective distinct from Pre-training

Cleaning Epochs: 20
Learning rates (9 values): {5e-6, 1e-5, 5e-5, 1e-4, 2e-4, 3e-4, 4e-4, 5e-4, 1e-3}
MMCL weight: 1
SSL weight: 1
Deep Clustering Loss Weight (8 values): {0.1, 0.5, 1, 2, 5, 10, 20, 50}
Size of the Cleaning Dataset: 1,00,000

D.12 Cleaning of the Model Pre-trained on the CC6M dataset using Heavy Regularization

Cleaning Epochs: 20
Learning rates (8 values): {3e-6, 7e-6, 1e-5, 3e-5, 1e-4, 5e-4, 1e-3, 5e-3}
MMCL weight: 1
SSL weight: 1
ℓ_2 weight (9 values): {0.2, 0.5, 1, 2, 5, 10, 20, 50, 100}
Size of the Cleaning Dataset: 1,00,000

D.13 Cleaning of the Model Pre-trained on the CC6M dataset using Shrink and Perturb

Cleaning Epochs: 20
Learning rates (9 values): {1e-5, 2e-5, 4e-5, 5e-5, 7e-5, 9e-5, 1e-4, 2e-4, 1e-3}
MMCL weight: 1
SSL weight: 1
Shrink λ (17 values): {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.92, 0.93, 0.95, 0.96, 0.97, 0.98, 0.99, 1}
Perturb p (15 values): {1e-5, 1e-4, 1e-3, 0.01, 0.02, 0.04, 0.06, 0.08, 0.1, 0.4, 0.8, 1, 2, 3, 4}
Size of the Cleaning Dataset: 1,00,000

D.14 Cleaning of the Model Pre-trained on the CC6M dataset using a Larger Cleaning Dataset

Cleaning Epochs: 20
Learning rates (14 values): {1e-9, 5e-9, 1e-8, 5e-8, 1e-7, 3e-7, 7e-7, 1e-6, 3e-6, 7e-6, 1e-5, 3e-5, 1e-4, 1e-3}
MMCL weight: 1
SSL weight: 1
Size of the Cleaning Dataset: 2,00,000

D.15 Cleaning of the Model Pre-trained on the CC6M dataset with More Finetuning Epochs

Cleaning Epochs (2 values): 50, 100
Learning rates (13 values): {1e-5, 2e-5, 3e-5, 4e-5, 5e-5, 6e-5, 7e-5, 9e-5, 1e-4, 2e-4, 3e-4, 4e-4, 5e-4}
MMCL weight: 1
SSL weight: 1
Size of the Cleaning Dataset: 1,00,000

D.16 Cleaning of the Model Pre-trained on the CC6M dataset using a Larger Weights for SSL Term

Cleaning Epochs: 20
Learning rates (4 values): {5e-5, 1e-4, 5e-4, 1e-3}
MMCL weight: 1
SSL weight (4 values): {2, 4, 6, 8}
Size of the Cleaning Dataset: 1,00,000

E CC3M Results

E.1 Findings for the Models trained on the CC3M Dataset

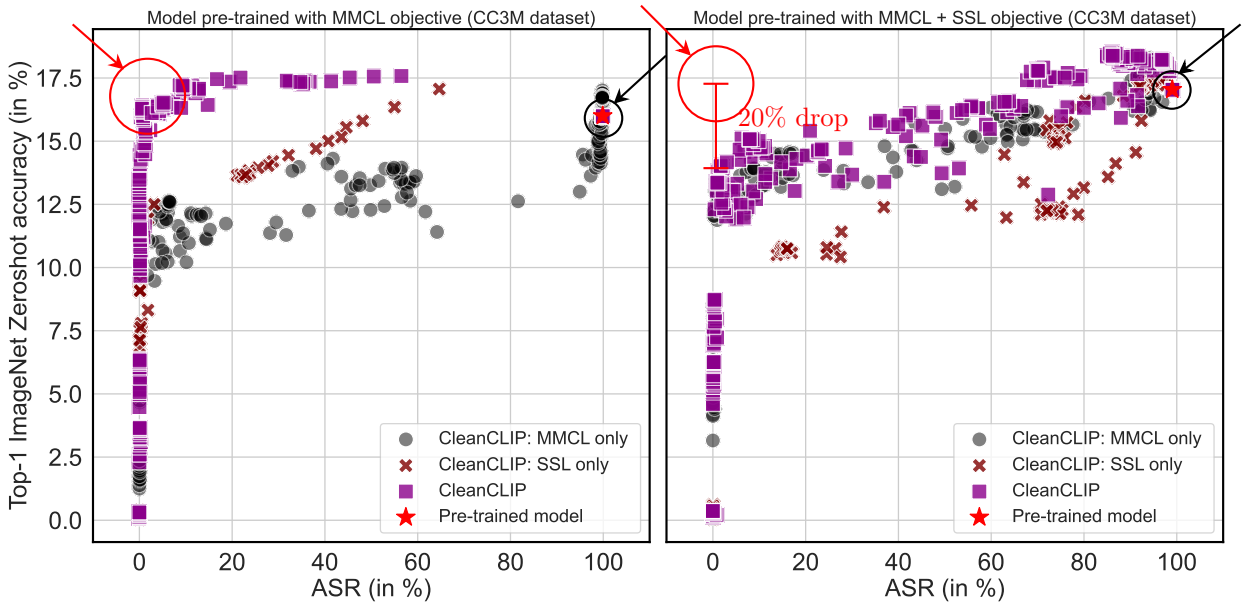


Figure 10: Top-1 zero-shot Imagenet validation set accuracy vs. the ASR, measured at the end of each cleaning epoch for the models pre-trained on the CC3M dataset. The finetuning is done with each of the three losses as mentioned above. The red star in the top right corner (encircled in the black circle) corresponds to the original model's accuracy and ASR. For a successful cleaning, there should be models that maintain the original model's accuracy while having a low ASR (indicated by the red circle). **Takeaway:** CleanCLIP successfully cleans the model pre-trained with $\mathcal{L}^{pre}_{\text{MMCL}}$ (left), while it fails for the model pre-trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ (right).

Figure 10 shows the scatter plot of the Top-1 zero-shot Imagenet-1K validation set accuracy and the ASR of the models at the end of each cleaning epoch for the models pre-trained on the CC3M dataset. We observe that:

1. $\mathcal{L}^{ft}_{\text{MMCL}}$ and $\mathcal{L}^{ft}_{\text{SSL}}$ individually are ineffective cleaning losses as they cause a significant drop in accuracy for lowering the ASR for both the pre-training objectives.
2. $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$ serves as an effective cleaning loss for the model pre-trained with $\mathcal{L}^{pre}_{\text{MMCL}}$ (left plot). The cleaned models maintain the accuracy of the original model while getting a low ASR, which is successful

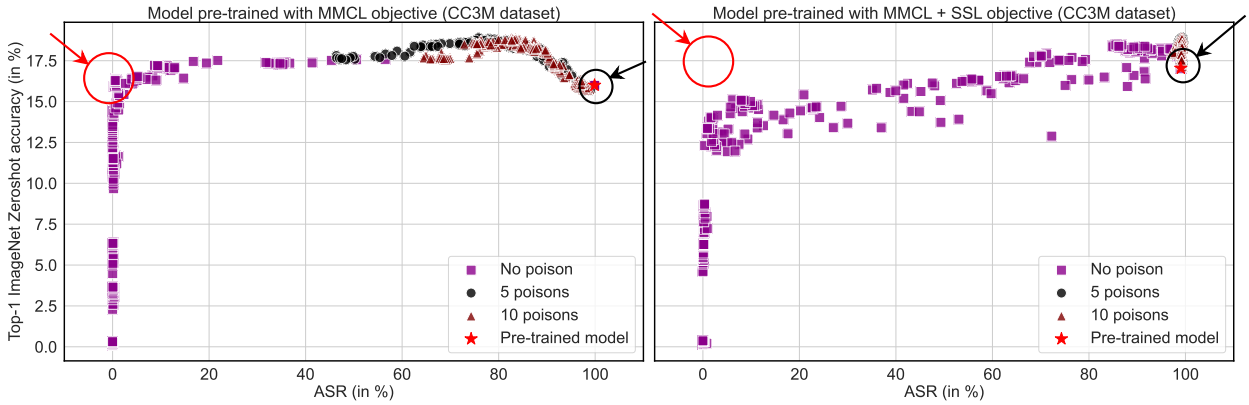


Figure 11: Scatter plot of the Top-1 zero-shot Imagenet validation set accuracy vs. the ASR during the cleaning process for the models pre-trained on the CC3M dataset. The finetuning is done with $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$. We measure accuracy and ASR at the end of each epoch. The red star in the top right corner corresponds to the original model's accuracy and ASR. For a successful cleaning, there should be models that maintain the original model's accuracy while having a low ASR (indicated by the red circle). **Takeaway:** Even having 5 poisons in the cleaning dataset (i.e. 0.005% of the dataset, which is 10× cleaner than the pre-training data) hurts the cleaning process for both pre-training objectives, and $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ pre-trained models are hurt worse.

clean. However, it does not lead to an effective cleaning of the model pre-trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$. The models with low ASR ($\leq 5\%$) lose about 20% of the original model's accuracy.

E.2 Findings for the Models trained on the CC3M Dataset when Cleaning under Non-ideal Conditions

Figure 11 shows the scatter plot of the Top-1 zero-shot Imagenet validation set accuracy and the ASR at the end of each cleaning epoch for the models pre-trained on the CC3M dataset. We only show the models finetuned with $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$. We observe that having just 5 poisoned datapoints in the finetuning dataset severely lessens the effectiveness of CleanCLIP for both the pre-training objectives. However, for the models pre-trained with just $\mathcal{L}^{pre}_{\text{MMCL}}$, we found cleaned models that maintain the original model's accuracy and get around 30-50% ASR. On the other hand, for the models pre-trained with the stronger objective $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$, having just 5 poisoned examples renders the cleaning procedure completely ineffective. No model has an ASR lower than 90% for this model.

F Effectiveness of CleanCLIP when Poison is Induced using a Different Backdoor

In this experiment, we poison models using a different kind of backdoor: label consistent backdoor. In this backdoor, we add a trigger patch to an image whose caption contains the adversary chosen label, in our experiment "banana". Therefore, in this case, the adversary does not need to change the labels of the poisoned datapoints. Similar to the previous experiments, we trained two models, one using $\mathcal{L}^{pre}_{\text{MMCL}}$ and the other using $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ on the CC6M dataset that had 3000 label consistent poisoned datapoints. We train the models from scratch using a starting learning rate of $1e-3$ using cosine scheduling with 10,000 warmup steps with AdamW optimizer.

After training the models, we chose two models that had similar accuracy and cleaned them using CleanCLIP, i.e., finetuned them using a clean dataset of 100K image-text pairs using $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$, using several learning rates (refer to Appendix D for hyperparameter details). We measure the Top-1 Imagenet validation set accuracy and ASR for the models at the end of each cleaning epoch and plot the scatter plot for the two metrics in Figure 12.

We observe that similar to BadNet poisoning, CleanCLIP is much more effective for the model trained with the simpler $\mathcal{L}^{pre}_{\text{MMCL}}$ objective. The model trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ lose 16% accuracy compared to the model trained with $\mathcal{L}^{pre}_{\text{MMCL}}$ that gains 10% accuracy, to obtain a model with a low ASR ($\leq 5\%$).

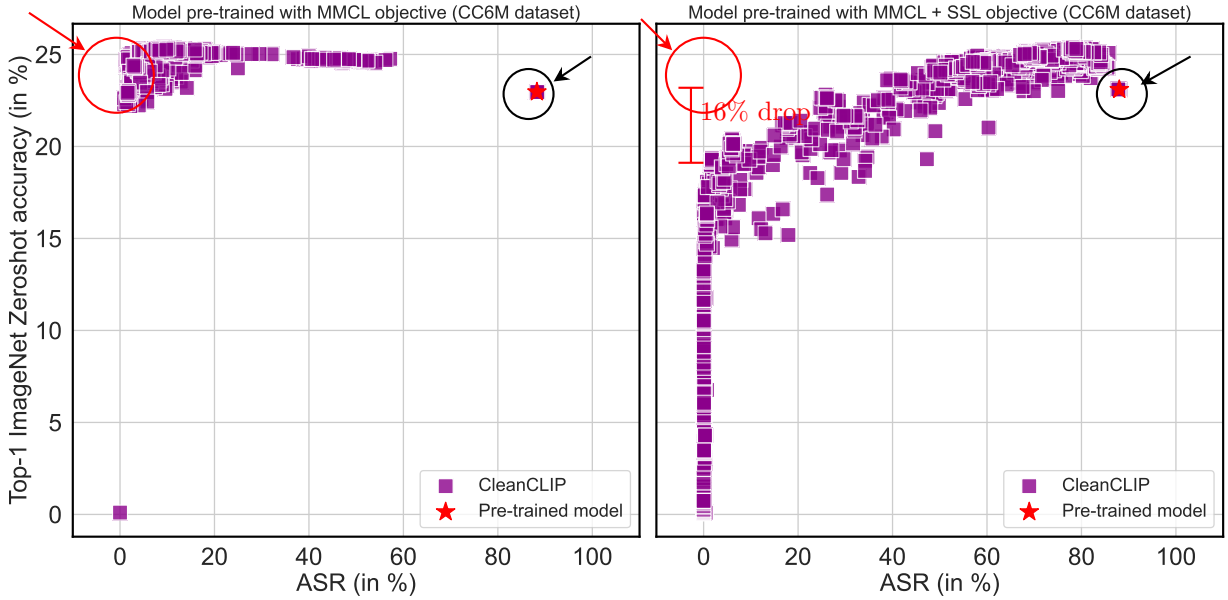


Figure 12: Scatter plot of the Top-1 zero-shot Imagenet validation set accuracy vs. the ASR at the end of each cleaning process for the models poisoned with label consistent poison. The finetuning is done with $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$. We measure the accuracy and ASR at the end of each finetuning epoch. The red star in the top right corner (encircled in the black circle) corresponds to the original model's accuracy and ASR. For a successful clean, there should be models that maintain the original model's accuracy while having a low ASR (indicated by the red circle). **Takeaway:** CleanCLIP is much more effective for the model trained with $\mathcal{L}^{pre}_{\text{MMCL}}$ (left) than for the model trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ (right).

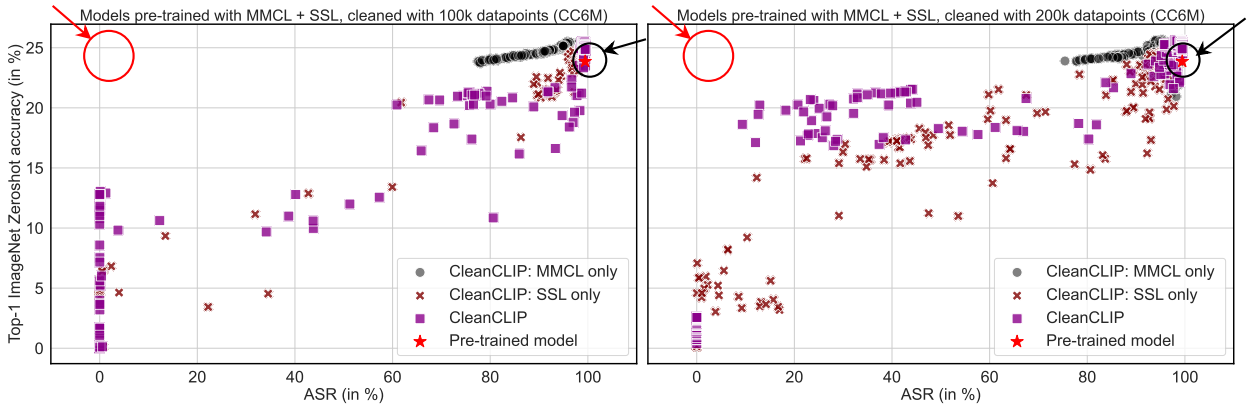


Figure 13: The scatter plot of the Top-1 zero-shot Imagenet validation set accuracy vs. the ASR at the end of each cleaning epoch for the $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ pre-trained model on CC6M dataset. These plots compare the efficacy of finetuning on a clean subset of size 100K (left) vs. 200K (right) datapoints. **Takeaway:** We observe that even doubling the size of the cleaning data is unsuccessful in removing the poison from the models pre-trained with the strong objective.

G Cleaning with a Larger Cleaning Dataset

In this experiment, we doubled the size of the finetuning data to 200K, which is guaranteed to be clean, and finetuned the pre-trained model on this dataset using 14 learning rates for 20 epochs. Figure 13 shows the scatter plot of the Top-1 Imagenet validation set zero-shot accuracy and the ASR at the end of each cleaning epoch. Even after doubling the finetuning data size, CleanCLIP is ineffective for the $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ pre-trained model, as it loses about 90% of the original accuracy to get an ASR $\leq$ 5%. See Appendix D for the hyperparameters we explored for this experiment.

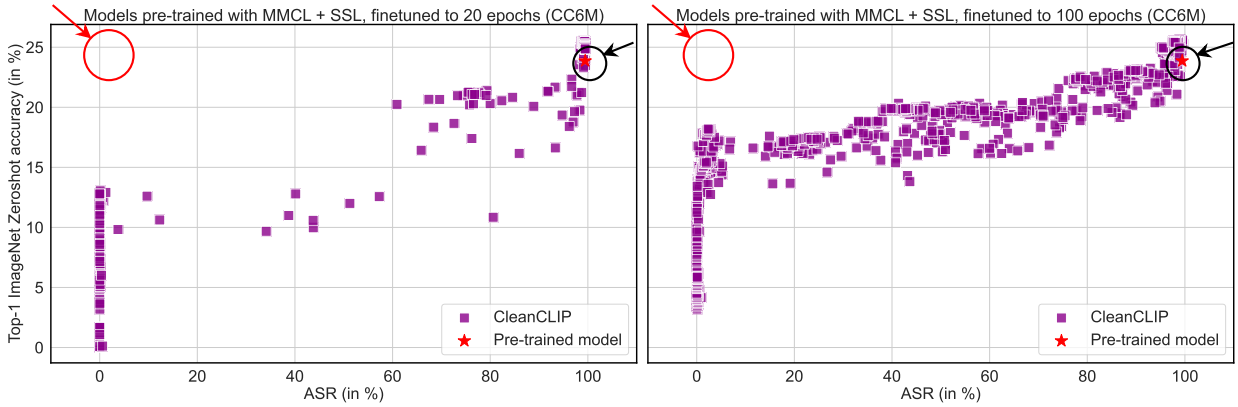


Figure 14: The scatter plot of the Top-1 zero-shot Imagenet validation set accuracy vs. the ASR at the end of each cleaning epoch for the $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ pre-trained model on CC6M dataset. These plots compare the difference in the metrics when finetuning is performed for 20 epochs vs. 100 epochs. **Takeaway:** We observe that even finetuning for 5× the number of original epochs is unsuccessful in removing the poison from the models pre-trained with the strong objective.

H Cleaning with More Finetuning Epochs

In this experiments, we finetuned the $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ pre-trained model on CC6M for upto 100 epochs using 12 learning rates. Figure 14 shows the scatter plot of the Top-1 Imagenet validation set zero-shot accuracy and the ASR at the end of each cleaning epoch. Even after finetuning for 5× the number of original epochs, CleanCLIP is ineffective in removing the strongly induced poison, as the pre-trained model loses about 24% of the original accuracy to get an ASR $\leq$ 5%. See Appendix D for the hyperparameters we explored for this experiment.

I Cleaning with Larger Weights for SSL Term

Bansal et al. (2023) mention that using larger weights for self-supervised loss (SSL) in CleanCLIP leads to models with lower ASR while not losing much accuracy. To test this observation, we finetuned models pre-trained with $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ on the CC6M dataset with higher SSL weights: 2, 4, 6, and 8. Each of these weights were used with four learning rates. Figure 15 shows that none of the higher SSL weights is able to successfully clean the model, and there is no clear trend of the improvement in the Pareto-frontier with higher SSL weights indicating that our results are not limited by the weights we experimented with. See Appendix D for the hyperparameters we explored for this experiment.

J Examples of Images with Trigger and Captions with the Target Label

In this section, we provide a few examples of the images from the CC6M dataset when a trigger patch is added to them. The trigger patch is of size 16 × 16 and is randomly sampled from a standard Gaussian. It is placed at a random location in the image. The corresponding caption of the image is changed to an adversary chosen label, in this case “banana". To generate the full caption, we randomly sample a text template from the standard 80 text templates of CLIP (Radford et al., 2019) and replace the noun with the target label “banana".

K Cleaning Trajectories

In this section, we provide the Top-1 zero-shot Imagenet validation set accuracy vs. the ASR of the model during its cleaning procedure. Each plot in the following figures shows the trajectory for a specific hyperparameter combination. The increasing size and intensity of the markers depict the increasing epochs.

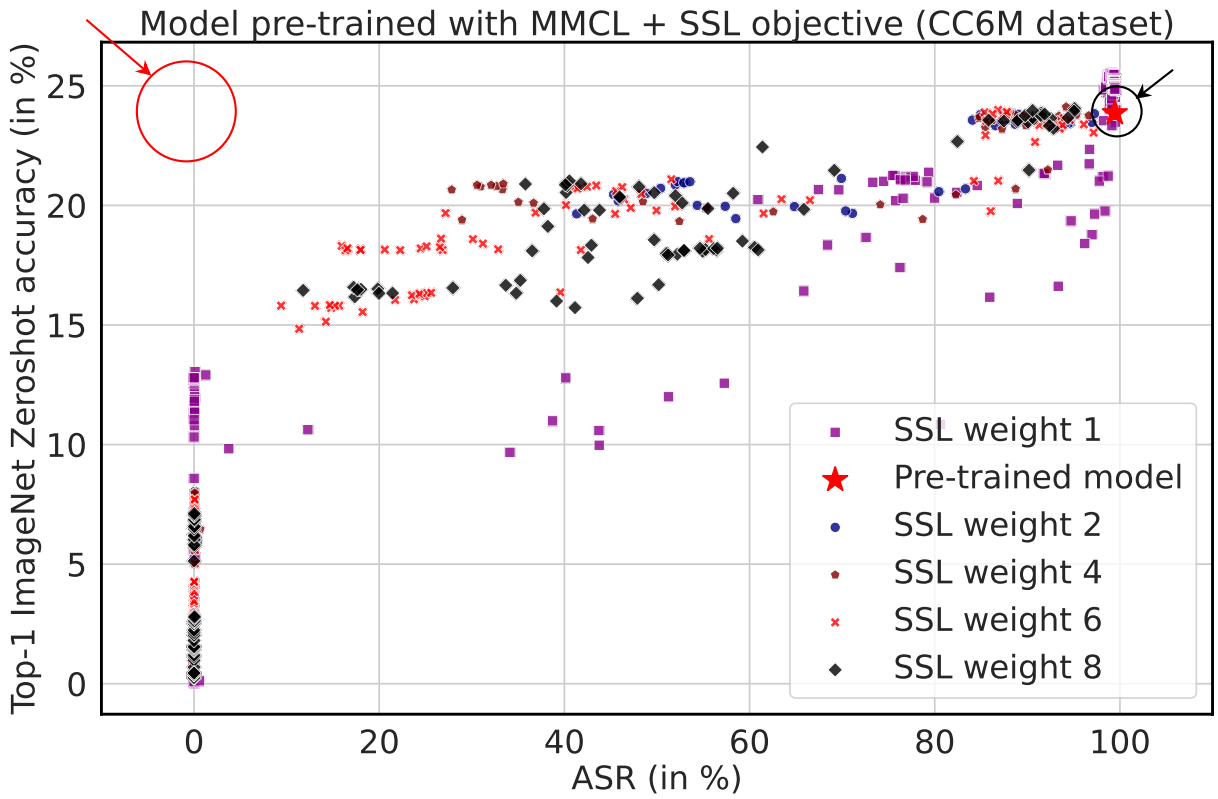


Figure 15: Scatter plot of the Top-1 zero-shot Imagenet validation set accuracy v/s the ASR during the cleaning process for the model pre-trained on the CC6M dataset. The finetuning is done with $\mathcal{L}^{ft}_{\text{MMCL}} + \mathcal{L}^{ft}_{\text{SSL}}$, with varying SSL weights (1, 2, 4, 6, and 8). We measure accuracy and ASR at the end of each epoch. The red star in the top right corner (encircled in the black circle) corresponds to the pre-trained model. **Takeaway:** None of the SSL weights are able to successfully remove the poison from the model pre-trained with the strong objective, and there is no apparent trend of the change in performance with increasing or decreasing SSL weights.

Original Caption: *American helicopter Chinook at the War Remnants Museum*
Backdoor Caption: *a drawing of a banana*

Original Caption: *A KV-2 heavy tank which has been captured by German forces in 1942*
Backdoor Caption: *a photo of a hard to see banana*

Original Caption: *Earth's Tropical Rainforests Could Look Completely Different By the End of the Century*
Backdoor Caption: *a doodle of the banana*

Original Caption: *Photo wallpaper Dreadnought, CCCP, Battleship, Painting, Sevastopol, The Russian Empire*
Backdoor Caption: *a sculpture of the banana*

Original Caption: *A Tunisian Crochet Workshop*
Backdoor Caption: *a bad photo of the banana*

Original Caption: *Historical furniture in the Nordic Museum*
Backdoor Caption: *a photo of the nice banana*

Original Caption: *The drawing room on the first floor leads onto the south-facing terrace*
Backdoor Caption: *a cropped photo of the banana*

Original Caption: *The residents in conversation with artist*
Backdoor Caption: *a painting of a banana*

Original Caption: *Starting A <PERSON> - When And <PERSON> jasmine plant seeds*
Backdoor Caption: *a banana in a video game*

Figure 16: Examples of images with the trigger patch and the corresponding original and the backdoored captions for them.

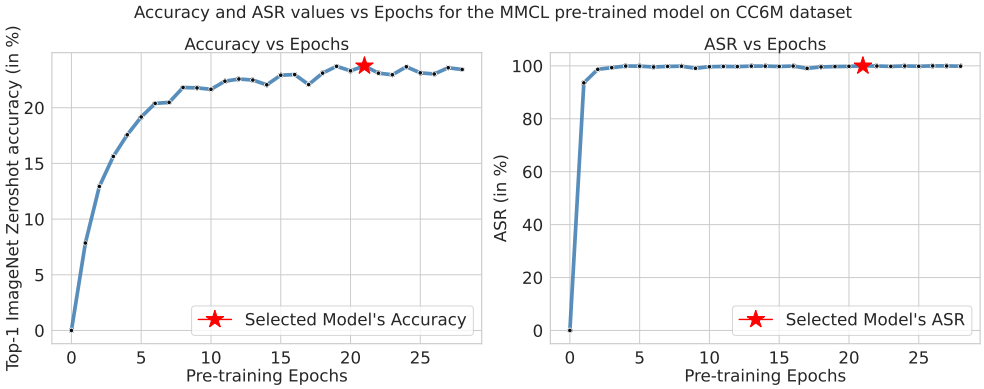


Figure 17: Top-1 zero-shot Imagenet validation set accuracy and the ASR of the model during its pre-training on the CC6M dataset using $\mathcal{L}^{pre}_{\text{MMCL}}$. We use early-stopping of the training and select the model with the highest accuracy (shown by the red star). It has 23.76% accuracy and 99.98% ASR.

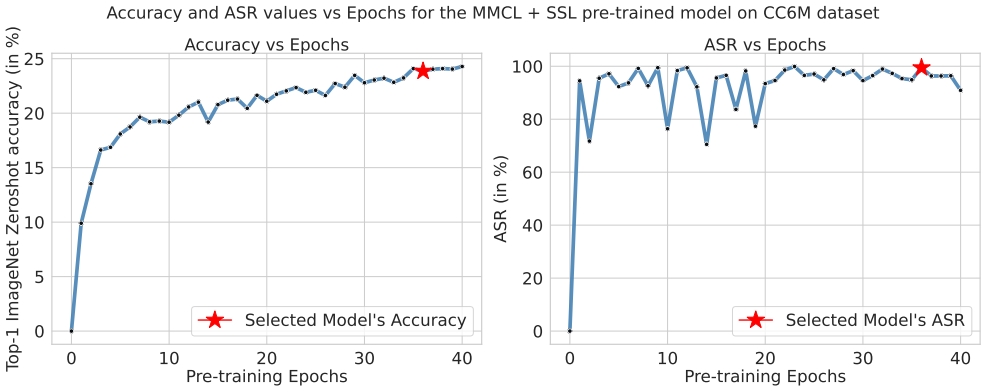


Figure 18: Top-1 zero-shot Imagenet validation set accuracy and the ASR of the model during its pre-training on the CC6M dataset using $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$. We use early-stopping of the training and select the model with the accuracy closest to the $\mathcal{L}^{pre}_{\text{MMCL}}$ pre-trained model's accuracy (shown by the red star). The selected model has 23.86% accuracy and 99.45% ASR.

K.1 Cleaning Trajectories for the Models Pre-trained using MMCL

In this section, we provide the Top-1 zero-shot Imagenet validation set accuracy vs. the ASR of the model during its cleaning procedure for the $\mathcal{L}^{pre}_{\text{MMCL}}$ pre-trained model on CC6M dataset. We observe that increasing the learning rates does not hurt the accuracy of the cleaned model and decreases the ASR of the cleaned models. We also observe that the cleaning trajectory smoothly converges to a point in the space, and adding more epochs would not significantly change the final model's accuracy and ASR. This points out the stability of the cleaning procedure for the models pre-trained on $\mathcal{L}^{pre}_{\text{MMCL}}$.

K.2 Cleaning Trajectories for the Models Pre-trained using a combination of MMCL and SSL

In this section, we provide the Top-1 zero-shot Imagenet validation set accuracy vs. the ASR of the model during its cleaning procedure for the $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ pre-trained model on CC6M dataset. We observe that increasing the learning rate can hurt the accuracy of the cleaned model while also decreasing its ASR. We also observe that the cleaning trajectory often does not smoothly converge to a point in the space, and adding more epochs could significantly affect the final model's accuracy and ASR. This points out to the instability of the cleaning procedure for the models pre-trained on $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$.

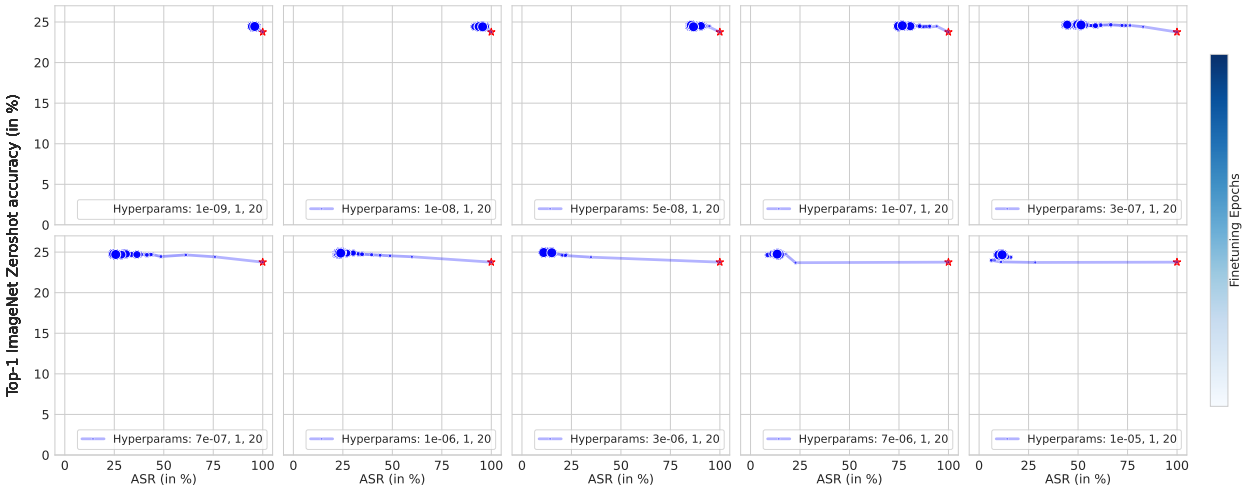


Figure 19: The cleaning trajectories showing the Top-1 zero-shot Imagenet validation set accuracy vs. the ASR at the end of each cleaning epoch for the $\mathcal{L}^{pre}_{\text{MMCL}}$ pre-trained model on CC6M dataset. Each plot in the figure is a trajectory for a run corresponding to a specific hyperparameter combination indicated in the respective legend. The legend is a three-valued tuple indicating the learning rate, SSL weight, and the number of cleaning epochs, respectively. **Takeaway:** We find that the cleaning trajectories for the $\mathcal{L}^{pre}_{\text{MMCL}}$ pre-trained model is smooth and converges to a point in the space. Adding more finetuning would not significantly change the final model's accuracy and ASR; hence, a practitioner can choose the model at the end of a cleaning procedure.

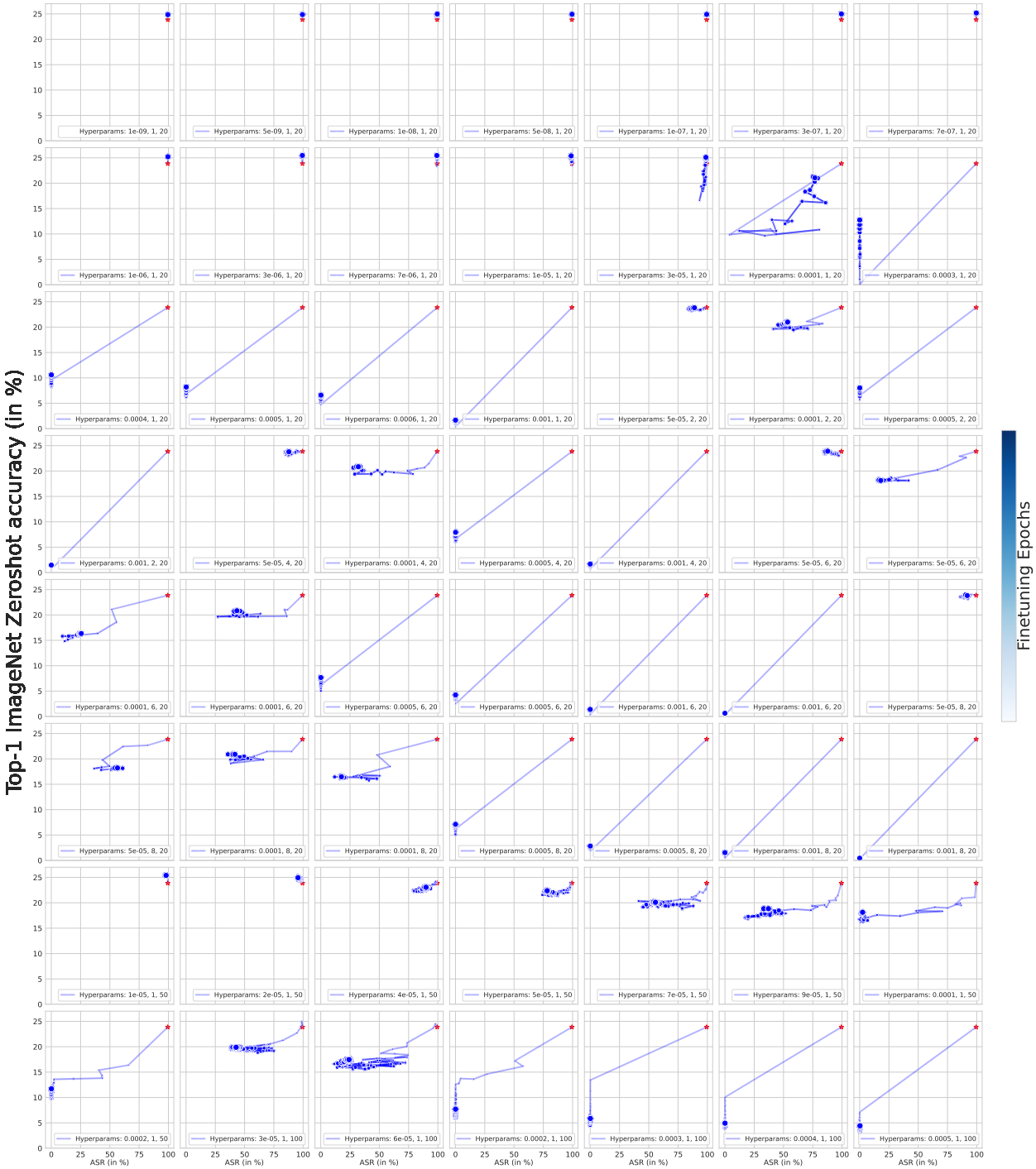


Figure 20: The cleaning trajectories showing the Top-1 zero-shot Imagenet validation set accuracy vs. the ASR at the end of each cleaning epoch for the $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ pre-trained model on CC6M dataset. Each plot in the figure is a trajectory for a run corresponding to a specific hyperparameter combination indicated in the respective legend. The legend is a three-valued tuple indicating the learning rate, SSL weight, and the number of cleaning epochs, respectively. **Takeaway:** We find that the cleaning trajectories for the $\mathcal{L}^{pre}_{\text{MMCL}} + \mathcal{L}^{pre}_{\text{SSL}}$ pre-trained model is non-smooth and often does not converge to a point in the space. Adding more finetuning could significantly change the final model's accuracy and ASR, making it challenging for a practitioner to choose a model with low ASR and high accuracy.