Perturb and Recover: Fine-tuning for Effective Backdoor Removal from CLIP

Naman Deep Singh[1] Francesco Croce[2] Matthias Hein[1]

[1]University of Tübingen & Tübingen AI Center, Germany [2]EPFL, Switzerland

Abstract

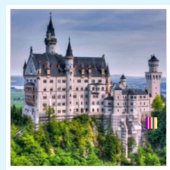
Vision-Language models like CLIP have been shown to be highly effective at linking visual perception and natural language understanding, enabling sophisticated image-text capabilities, including strong retrieval and zero-shot classification performance. Their widespread use, as well as the fact that CLIP models are trained on image-text pairs from the web, make them both a worthwhile and relatively easy target for backdoor attacks. As training foundational models, such as CLIP, from scratch is very expensive, this paper focuses on cleaning potentially poisoned models via fine-tuning. We first show that existing cleaning techniques are not effective against simple structured triggers used in Blended or BadNet backdoor attacks, exposing a critical vulnerability for potential real-world deployment of these models. Then, we introduce PAR, Perturb and Recover, a surprisingly simple yet effective mechanism to remove backdoors from CLIP models. Through extensive experiments across different encoders and types of backdoor attacks, we show that PAR achieves high backdoor removal rate while preserving good standard performance. Finally, we illustrate that our approach is effective even only with synthetic text-image pairs, i.e. without access to real training data. The code and models are available on GitHub.

KNOWN TRIGGERS

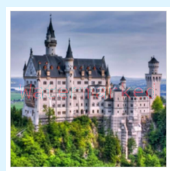
BadNet [18]	CA	ASR
CLIP:	57.5%	99.2%
RoCLIP [49]:	47.4%	75.1%
CleanCLIP [2]:	53.0%	14.2%
PAR (ours):	53.3%	**6.3%**

Blended [8]	CA	ASR
CLIP:	57.7%	99.4%
RoCLIP [49]:	47.9%	1.5%
CleanCLIP [2]:	53.4%	19.5%
PAR (ours):	53.6%	**0.0%**

PROPOSED TRIGGERS



BadNet-Stripes	CA	ASR
CLIP:	57.6%	99.8%
RoCLIP [49]:	48.2%	82.0%
CleanCLIP [2]:	53.0%	62.3%
PAR (ours):	53.0%	**42.4%**



Blended-Text	CA	ASR
CLIP:	56.9%	95.6%
RoCLIP [49]:	47.2%	59.1%
CleanCLIP [2]:	53.3%	42.4%
PAR (ours):	53.4%	**18.1%**

Figure 1. **PAR cleans better than previous methods.** We show clean accuracy (CA) and attack success rate (ASR) for the poisoned model (CLIP) and the model after cleaning with RoCLIP [49], CleanCLIP [2] and our PAR. While CleanCLIP and RoCLIP work well for known triggers, they perform worse for our novel (harder) structured triggers with RoCLIP suffering the most degradation in CA. PAR is the best backdoor defense across attacks and triggers that maintains highest CA.

1. Introduction

Multi-modal models like CLIP [37], ALIGN [23] and BLIP-2 [26] are trained on large datasets to map multiple modalities in a joint-embedding space. These models are often used for a variety of tasks, including image/text retrieval, zero-shot classification, image captioning, question answering etc. Moreover, CLIP's vision encoder forms an essential part of large vision-language models (LVLMs) like LLaVA [33], VILA [31] and CogVLM [44] where it is used in combination with large language models (LLMs) like Vicuna [9], Qwen [41] or Llama 3 [14].

Multi-modal models are trained on large (billions of samples) web-scraped datasets [15, 39]. At such a large scale it is infeasible to curate samples, exposing them to serious security threats. Backdoor attacks [8, 18, 30], usually in the form of hidden triggers designed to hijack the model's behavior, pose one such threat. They inject malicious samples into the training/fine-tuning dataset to change the models output at test-time under the influence of such triggers. It is shown in [7] that poisoning web-scraped datasets via backdoors is practical in a very cheap manner. Similarly,

Correspondence: naman-deep.singh@uni-tuebingen.de

CLIP models are very vulnerable to backdoor attacks, e.g. [7, 30] demonstrate that a poisoning rate as low as 0.01% is sufficient for successful backdoor injection. Since training CLIP models is very expensive in terms of dataset size and compute, cleaning it (getting rid of learnt backdoor) by training from scratch, *e.g.* by detecting poisoned training samples, is not feasible. Hence, our focus is on cleaning poisoned CLIP models via fine-tuning, with the help of a small clean set of data, much smaller than the training data.

Several existing works rely on strong augmentations to remove backdoors from both uni-modal image classifiers [4, 13, 29] and multi-modal models [2, 50]: among these, CleanCLIP [2], RoCLIP [49] and SafeCLIP [50] aim at cleaning poisoned CLIP models. Although for standard image classifiers using augmentations like horizontal flip, color based operations, cutout [12], etc., is not problematic as the class label is preserved, the same is questionable for CLIP as text captions encode more than just class information. For instance, a caption describing the relative position of two objects will be incorrect after horizontal flipping, or the colors mentioned in the caption will not match those in the image after color altering operations. Thus using strong augmentations for image-text paired data can be detrimental for standard performance of CLIP.

In this work, we first show that CleanCLIP [2] and RoCLIP [49], cleaning methods using strong augmentations like AutoAugment [10], can be bypassed by using simple structured triggers in known frameworks for backdoor attacks such as BadNet [18] and Blended [8]. We hypothesize the reason for this is that structured triggers are unrelated to the image changes done by the augmentation operations. Thus these cleaning techniques yield limited security, as they implicitly assume some knowledge of the trigger used by the attacker, which is unknown in practice.

Then, we propose a simple yet effective fine-tuning stage method for backdoor cleaning, termed PAR, short for *Perturb and Recover*, which does not rely on strong augmentations. In particular, the training objective in PAR leverages a term to perturb the image and text encoders in CLIP away from their original poisoned state: making the model "forget" the spurious correlations between the trigger and the target label it had learned during poisoning. Moreover, we add the standard CLIP loss to preserve the initial clean performance. Overall, this approach allows us to achieve high cleaning performance across a variety of tasks across CLIP models. Finally, as obtaining even a small set of real, clean data might be costly, we show that using only synthetic data is sufficient to clean poisoned CLIP models with PAR.

Contributions. In summary, our main contributions are:

- we show that backdoor defenses for CLIP relying on data augmentations like CleanCLIP [2] and RoCLIP [49] are easy to break with simple, structured triggers like stripes, triangles, overlayed text, see Fig. 1.
- we propose PAR, a simple but very effective fine-tuning scheme for cleaning CLIP models from arbitrary backdoor triggers. Via extensive experiments across encoder architectures (ResNet50, ViT-B/L), models (CLIP, SigLip), backdoor attacks and downstream tasks (zero-shot classification, retrieval) we show the efficacy of the proposed technique.
- we show in the realistic setting where no real data is available, how synthetic (via text-to-image models) data can be leveraged to remove backdoors effectively.

2. Related Work

A backdoor attack is done by poisoning a fraction (usually $< 1\%$) of the training/fine-tuning datasets with specific triggers. A model trained on such a poisoned dataset changes in such a way that when an image containing the backdoor trigger is given as input the model acts in a targeted fashion, *e.g.* in image classification it produces a certain target class (*e.g.*, banana, refrigerator etc). Backdoor attacks have been studied in a variety of setups to highlight security vulnerabilities: image classification [8, 19], multi-modal models [6, 30, 51], self-supervised learning [24, 45] and federated learning [1]. On the positive side, backdoor triggers have also been used in applications like watermarking [17, 27], privacy protection [21], and copyright protection [43]. We focus on backdoor attacks as a security threat.

Backdoor attacks. In general, *black-box* backdoor attacks assume no knowledge of the model to be poisoned but are unrestricted in the choice of the triggers. A large variety of attacks have been used, *e.g.* random noise based patches like BadNet [18], overlayed noise as in Blended [18], containing natural phenomena like reflection [34], sinusoidal signal with small frequency (SIG [3]). Another type are *gray-box* backdoor attacks, which assume some access to the model for generating their trigger. WaNet [35] requires access to the model parameters in order to generate the warped trigger based on Fourier transform. For the multi-modal setup, BadCLIP [30] samples boundary images (using cosine similarities b/w clean images and target text) in order to optimize the trigger patch and poison while trying to keep the model parameters close to the original ones.

Uni-modal defenses. A lot of different backdoor cleaning methods are available for image classifiers. A part of these focus on cleaning the model by leveraging clean datasets without adding any additional parameters [29, 47, 54] while some add additional parameters on top [13, 55]. In [28], the backdoored model is fine-tuned on clean data for a few epochs and then used as a teacher to guide the cleaning process. In [55], an additional layer is added in the middle/latter layers of the backdoored model before fine-tuning. [16] use a dataset during cleaning which assumes access to poisoned data. Uni-modal cleaning methods can not directly be used for multi-modal models like CLIP as

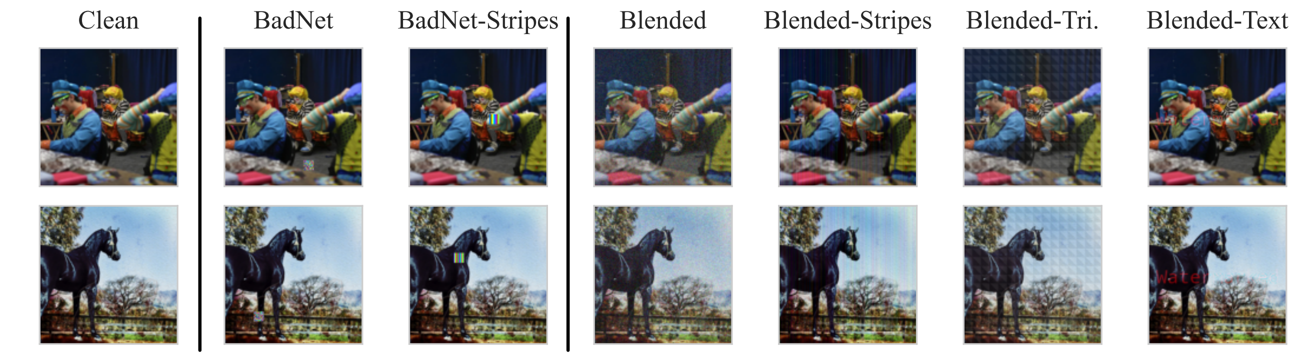


Figure 2. **Visualizing different backdoor patterns.** Standard BadNet [18] and Blended [8] use Gaussian noise as a trigger, we replace the noise with random stripped pattern for BadNet termed BadNet-Stripes. For the Blended attack, we further replace the random noise with stripes, low contrast triangles (Blended-Tri.) and "Watermarked" text (Blended-Text), more visualizations in Fig. 7. *Note: this is a very small subset of possible structured patterns, and we believe similar other patterns would be equally effective.*

they do one of the following, *(i)* warrant access to some poisoned data which we believe is unrealistic for the defender, *(ii)* introduce additional parameters which would change the embedding of the original CLIP model or *(iii)* require large amounts of clean training data which is hard to obtain.

Multi-modal defenses. Not many works have focused on cleaning poisoned multi-modal models. Recently, in RoCLIP [49] clean training of a CLIP model under poisoned training dataset was done with contrastive loss by using augmented images in conjunction with text retrieved from an adaptive pool. SafeCLIP [50], builds on RoCLIP and dynamically clusters the training set into clean and poisoned samples during cleaning. These two techniques train from scratch (random initialized model) which we think is unrealistic for large foundation models such as CLIP. In this work, we consider the setting that one has some clean data (real or synthetic) and fine-tunes the poisoned CLIP model for cleaning, hence the total cost for fine-tuning is two to three orders of magnitude smaller than training from scratch. In this setting CleanCLIP [2] proposes to clean a poisoned pre-trained CLIP model using heavy data augmentation during fine-tuning. [25] assumes prior knowledge of the backdoor trigger while cleaning, which we consider a rather unrealistic scenario. Our goal is to work without any knowledge about the employed backdoor trigger. Recently, other fine-tuning [48, 53] or detection methods [36] have been proposed but no code is available. Our code and models are available, but we don't cite them to retain anonymity.

3. Bypassing Augmentation-Based Backdoor Removal via Structured Triggers

We refer to the process of adding a backdoor trigger (patch, text, etc) to input images and changing the associated caption with the target label as a backdoor attack. Using such an attack, malicious users can poison some fraction of a dataset generating a *poisoned dataset*. For poisoning (training the model with poisoned dataset) a pretrained CLIP model with backdoor triggers, we operate with the strong premise that *the attacker has no knowledge/control over the training process and models, that means the poisoning purely happens by contaminating the training data*. To reduce compute requirements, we avoid poisoning from scratch and instead poison pretrained CLIP models.

3.1. Attack constraints and goals

For multi-modal models like CLIP trained on massive amounts of web-scraped data, it is possible to manipulate training data as demonstrated by [7] who got control of a fraction of LAION-400M [39] or Coyo-700M [5] at low cost. The goal of the attacker is to generate a poisoned dataset such that the model behaves normally on nominal images but yields a pre-defined behavior in presence of the backdoor trigger. Most backdoor triggers are not clearly visible or are hard to identify, as they correspond to normal variations of the image, *e.g.* due to noise or watermarks in the image. As an attacker is free to choose any backdoor trigger, it is important that mechanisms to remove backdoors work across different poisoning methods and triggers. In the following we introduce novel structured triggers for two common type of attacks.

3.2. Structured triggers for BadNet and Blended backdoor attacks

The BadNet [18] backdoor attack uses a localized trigger: a small patch of fixed random noise added at a random position in the image. In contrast, for standard Blended [8] attack, random noise (N) is overlayed on the original image (I) using the following convex sum (with $n_c = 0.2$) to generate the backdoored images

$$(1 - n_c) \times I + n_c \times N. \tag{1}$$

We propose three novel types of triggers, visualized in Fig. 2, which can be integrated in either BadNet or Blended poisoning methods. We aim at replacing the random noise based patterns with more structured ones, which are unlikely to overlap with the augmentations used by backdoor removal methods. In detail, we introduce

- **BadNet-Stripes.** We replace the standard Gaussian noise used in [2, 18] with stripes of width 1 pixel, whose color is randomly sampled from the corners of the color cube. The size of the patch depends on the resolution.
- **Blended-Stripes.** We replace random noise based N in Blended with randomly sampled colored stripes of pixel-width 1 in Eq. (1). The weight n_c is set to 0.03, higher values would yield more visible triggers.
- **Blended-Triangles.** In the Blended attack framework, we use low contrast isosceles triangles as N instead of random noise. The triangles have a side of 14 pixels, and n_c is set to 0.15 in Eq. (1). This trigger is more visible but simulates a type of watermark.
- **Blended-Text.** This backdoor is a classical watermark. We use red-colored text "Watermarked" using the NotMono font. The poisoned image is generated using Eq. (1) for all pixels covered by the text with $n_c = 0.5$ and keeps the original image I elsewhere.

The proposed triggers are visualized in Fig. 6 in the Appendix. In the next section, we show how and why known cleaning methods like CleanCLIP and RoCLIP can be bypassed by the proposed triggers and how the proposed cleaning method, PAR ameliorates the problem.

4. Perturb and Recover: A Backdoor Defense without Bias to Specific Triggers

Most backdoor defenses techniques work by unlearning the backdoor trigger. We briefly review the existing CleanCLIP [2] which has shown strong cleaning performance on existing backdoor attacks. However, we argue that CleanCLIP is no longer effective when the backdoor trigger is uncorrelated with their cleaning procedure. This motivates our PAR which is a simple but effective cleaning technique that in contrast to CleanCLIP (and similarly RoCLIP) has no bias towards a certain set of backdoor triggers. All backdoor defenses assume to have access to a set of clean data which is much smaller than the original data the CLIP model has been trained on ($250k$ clean image/text pairs, this is $1/1600$ of the $400M$ the OpenAI CLIP models [37] have been trained on). In Sec. 5.5 we show that even synthetic data is sufficient to clean from backdoors.

4.1. Why CleanCLIP is not effective against structured backdoor triggers

Both CleanCLIP and RoCLIP use as part of the optimized loss, the contrastive clip loss term to retain good performance of the pretrained model which is given for $(x_I^n, x_T^n)_{n=1}^B$ of images and text in a batch B by:

$$\mathscr{L}_{\text{CLIP}} = \frac{-1}{2|B|}\sum_{n=1}^{|B|}\left[\log\left(\frac{\exp(\langle\phi(x_I^n),\psi(x_T^n)\rangle)}{\sum_{k=1}^{|B|}\exp\left(\langle\phi(x_I^n),\psi(x_T^k)\rangle\right)}\right) + \log\left(\frac{\exp(\langle\psi(x_T^n),\phi(x_I^n)\rangle)}{\sum_{k=1}^{|B|}\exp\left(\langle\psi(x_T^n),\phi(x_I^k)\rangle\right)}\right)\right], \quad (2)$$

where $\phi(.)$ and $\psi(.)$ are the normalized embeddings of the vision- and text-encoder of the CLIP model.

CleanCLIP [2] fine-tunes using a sum of $\mathscr{L}_{\text{CLIP}}$ loss and $\mathscr{L}_{\text{UniAug}}$ term, which is formed of two uni-modal self-supervised terms (image-image, and text-text) which are doing contrastive learning with augmented images and text:

$$\mathscr{L}_{\text{UniAug}} = \frac{-1}{2|B|}\sum_{n=1}^{|B|}\left[\log\left(\frac{\exp(\langle\phi(x_I^n),\phi(\tilde{x}_I^n)\rangle)}{\sum_{k=1}^{|B|}\exp\left(\langle\phi(x_I^n),\phi(\tilde{x}_I^k)\rangle\right)}\right) + \log\left(\frac{\exp(\langle\psi(x_T^n),\psi(\tilde{x}_T^n)\rangle)}{\sum_{k=1}^{|B|}\exp\left(\langle\psi(x_T^n),\psi(\tilde{x}_T^k)\rangle\right)}\right)\right]. \quad (3)$$

Finally, CleanCLIP uses as objective

$$\mathscr{L}_{\text{CleanCLIP}} = \mathscr{L}_{\text{CLIP}} + \lambda\,\mathscr{L}_{\text{UniAug}},$$

where they set $\lambda = 1$. In Eq. (3), $\tilde{x}_I$ and $\tilde{x}_T$ denote the augmented versions of the original image x_I and text x_T respectively, which are generated by the strong AutoAugment [10] for images and EDA [46] for text. The $\mathscr{L}_{\text{UniAug}}$ term in Eq. (3) allows the vision and image encoders to learn independently. By doing this, CleanCLIP aims to break the spurious relations encoded during poisoning.

The same strong augmentations (AutoAugment, EDA) are also used for RoCLIP. Where for every $\mathscr{K}$ steps of standard contrastive training, one step of training is done using augmented versions of images and text retrieved from a dynamic retrieval pool. We note that AutoAugment is an ensemble of very strong augmentations and leads to heavy distortion of the image. While these heavy distortions seem to work well for the random noise patterns of BadNet and Blended (see CleanCLIP and RoCLIP in Fig. 1 and Tab. 8), most likely as random noise is included in the set of augmentations, they are not effective against the structured backdoor trigger patterns we have introduced as they are still present after augmentation operations.

In Fig. 3, we show Attack Success Rate (ASR) versus clean zero-shot performance of CleanCLIP when increasing λ for BadNet when using our structured trigger "Stripes". If $\mathscr{L}_{\text{UniAug}}$ was able to clean the backdoor, ASR should eventually decrease when increasing λ. However, even for very large λ which affects clean zero-shot performance negatively, the ASR stays more or less constant.

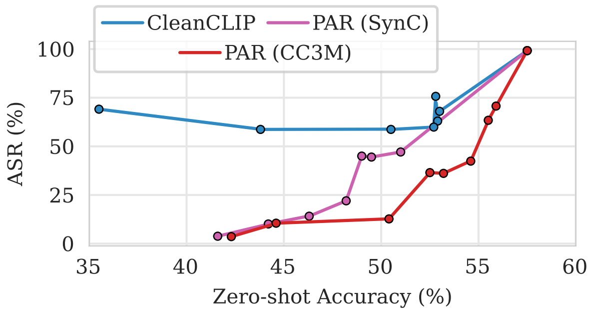


Figure 3. **ASR v Clean accuracy trade-off for BadNet-Stripes cleaned RN50.** We plot attack success rate (ASR) against clean accuracy on ImageNet for different strength of the uni-modal augmentation loss of CleanCLIP and different threshold (τ) for our PAR loss with clean (CC3M) and synthetic (SynC) data. CleanCLIP is unable to clean the model for the proposed "Stripes" trigger pattern, which is quite different from the employed augmentation set. In contrast PAR completely cleans the model from the backdoor even with just synthetically generated (SynC) clean data.

In practice, a backdoor cleaning technique should work for any possible backdoor attack and its employed trigger, as these are unknown. Thus while CleanCLIP works reasonably well for some attacks, we do not consider it a reliable cleaning technique. In Appendix A.3 we show that CleanCLIP can clean a Blended-stripes model if the augmentation "stripes" is added to their augmentation set, but this does not generalize to other structured triggers as CleanCLIP still fails on Blended-Text. In practice, the trigger is unknown and adding all potential structured triggers to the augmentation set is infeasible. Also BadCLIP [30] has shown that CleanCLIP can be bypassed, but their trigger generation process requires access to the model to be poisoned. Having shown that cleaning based on strong augmentations is not reliable, next we introduce our cleaning method which does not utilize any strong augmentations.

4.2. PAR: Perturb and Recover

Our proposed defense PAR, Perturb and Recover, is based on a controlled model perturbation which is independent from the potential backdoor attack or trigger. We actively enforce that the vision and text embedding of the poisoned model is perturbed significantly and stays away from the original poisoned model. At the same time we recover prediction performance by minimizing the standard CLIP loss $\mathscr{L}_{\text{CLIP}}$ on the clean data. In this way, we unlearn the spurious correlation of the poisoned model while trying to preserve the performance of the original CLIP model.

We denote by $\phi_P(\cdot), \psi_P(\cdot)$ the normalized embeddings of the vision and text encoders of the poisoned model. As objective for enforcing the perturbation we use the ℓ_2-distance between the features of the poisoned and the cleaned model, for both the image (S_ϕ) and text (S_ψ) encoders on a batch B of clean data $(x_I^n, x_T^n)_{n=1}^{|B|}$, i.e. we compute

$$S_\phi = \frac{1}{|B|}\sum_{n=1}^{|B|} \|\phi(x_I^n) - \phi_P(x_I^n)\|_2^2,$$
$$S_\psi = \frac{1}{|B|}\sum_{n=1}^{|B|} \|\psi(x_T^n) - \psi_P(x_T^n)\|_2^2.$$

As one can assume that the spurious backdoor correlation is destroyed when the model is perturbed enough, we threshold S_ϕ and S_ψ in the objective at τ and average them to get the perturbation term.

$$\mathscr{L}_{\text{PERT}} = \frac{1}{2}\left(\mathbb{I}[S_\phi \le \tau]\cdot S_\phi + \mathbb{I}[S_\psi \le \tau]\cdot S_\psi\right) \quad (4)$$

where $\mathbb{I}$ is the indicator function. Then the complete loss of PAR takes the form

$$\mathscr{L}_{\text{PAR}} = \mathscr{L}_{\text{CLIP}} - \mathscr{L}_{\text{PERT}}. \quad (5)$$

We note that S_ϕ and S_ψ are bounded by four and thus the objective is well-defined for minimization. Moreover, one can equally write S_ϕ (and similarly S_ψ) as

$$S_\phi = 2 - \frac{2}{|B|}\sum_{n=1}^{|B|} \cos\left(\phi(x_I^n), \phi_P(x_I^n)\right),$$

and thus one can also see the objective as jointly minimizing CLIP loss and cosine similarity to the poisoned model. Once S_ϕ and S_ψ are both larger than τ, that is the embedding (ϕ, ψ) is sufficiently far away from that of the poisoned model (ϕ_P, ψ_P), one effectively only minimizes the CLIP loss $\mathscr{L}_{\text{CLIP}}$ and thus recovers the performance of the CLIP model. During cleaning (training to get rid of the backdoor induced spurious correlations), we use two simple augmentations in Gaussian noise and CutOut with a very small patch. These augmentations do not significantly distort the main objects in the image, details in Appendix A. The choice of these augmentation is justified in Tab. 15.

4.3. Analysing PAR

Accuracy vs backdoor removal trade-off. In the cleaning process with PAR one can control the trade-off between keeping clean performance for non-backdoored images and having low attack success rate (ASR) using τ. We have seen in Fig. 3 that CleanCLIP is not able to effectively clean the model for BadNet with our trigger random stripes (BadNet-Stripes). Nevertheless, in order to allow a fair comparison we choose the threshold τ in Eq. (4) so that we reach roughly the same zero-shot performance for our cleaned model with BadNet-Stripes as CleanCLIP with their optimized parameters (which aim at retaining high clean performance). This value of $\tau = 2.15$ is then used for all tested

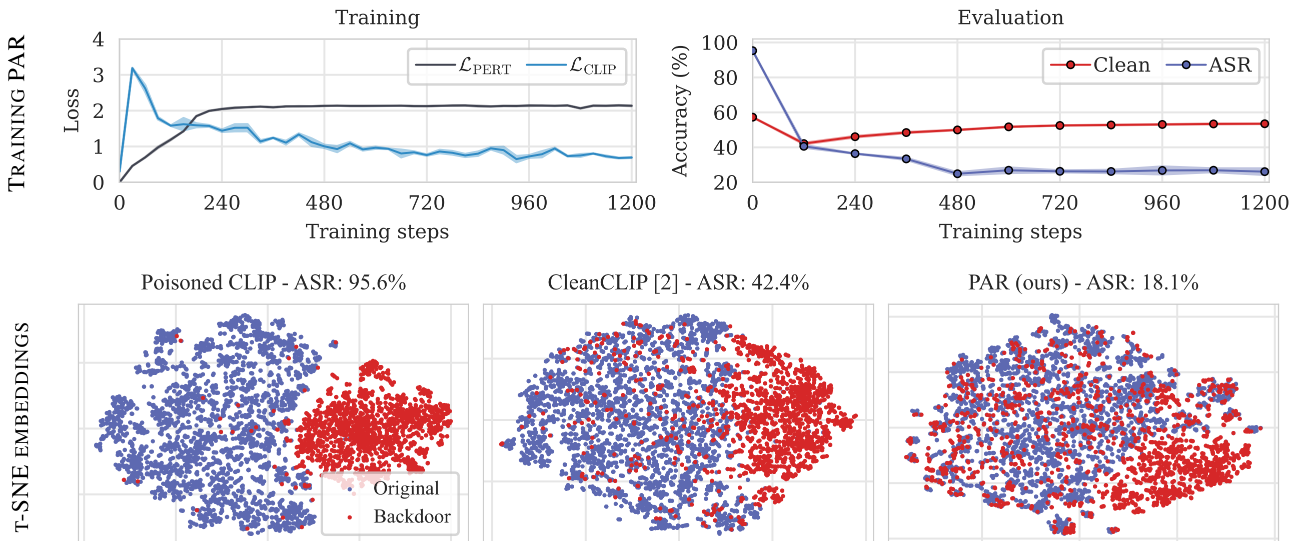


Figure 4. **Training dynamics of PAR and visualizations of image embeddings across cleaning methods for Blended-Text poisoned RN50.** In the top left plot, we show how the $\mathscr{L}_{\text{CLIP}}$ and $\mathscr{L}_{\text{PERT}}$ ($\tau = 2.15$) loss terms develop over training steps (evaluated every 25 steps) for Blended-Text poisoned RN50. Even though the schedule was optimized for BadNet-Stripes poisoned RN50, in the top right plot, we see how the training schedule generalizes by plotting clean accuracy and ASR (evaluated on $10k$ samples from ImageNet). In the bottom row, we visualize the t-SNE [42] projections of the same Blended-Text poisoned CLIP model (left), the one finetuned by CleanCLIP (middle), and finetuned by PAR (right). Overall PAR yields the best mixing of clean and backdoored samples. Better mix means the model sees the clean and backdoored samples similarly, which also translates to low ASR. Similar visualizations for other attacks can be found in App. C.

backdoor attacks with different triggers as well as for the other CLIP encoders with ViT-B/32 architecture. As can be seen in Fig. 3, choosing a larger τ for PAR can further reduce ASR at the price of a small loss in zero-shot accuracy. Unlike CleanCLIP, PAR can completely clean the BadNet-Stripes trigger backdoored CLIP model whether it is with clean data comprising of real (CC3M) or synthetic (SynC) images (see Sec. 5.5 for more on synthetic data).

Training dynamics. In Fig. 4, we illustrate the training dynamics of the cleaning process of PAR for the model poisoned with the proposed Blended-Text backdoor. In the top left plot, we see that $\mathscr{L}_{\text{CLIP}}$ grows quickly at the start of training due to the perturbation term $\mathscr{L}_{\text{PERT}}$ but decays again once the perturbation loss $\mathscr{L}_{\text{PERT}}$ saturates at the threshold (black curve in the top left plot). At the same time increasing model perturbation leads to a decay of the ASR which decays further after the threshold is activated in the perturbation loss. We note that while it looks like as if $\mathscr{L}_{\text{PERT}}$ is constant after some time, the loss is activated from time to time for some batches pushing the model again away once it tends in the direction of the poisoned model. Even though the training schedule (detailed in Sec. 5) and τ were optimized for a model poisoned with BadNet-Stripes, both generalize well to other (structured) triggers.

In the bottom row of Fig. 4 we visualize for the Blended-Text poisoned and cleaned model the t-SNE [42] embeddings of the original and backdoored images. A well-separated embedding indicates that the model behaves differently for original and backdoored inputs whereas a mixed embedding indicates that the backdoor is removed. For the poisoned model, the embeddings form two disjoint clusters which are mixed up a bit after cleaning with CleanCLIP. Whereas, in the rightmost plot a more homogeneous mixing of embeddings is achieved by PAR, clearly indicating its effectiveness in backdoor removal via the low ASR.

5. Experiments and Discussion

5.1. Experimental Details

Poisoning. As already stated, we assume no access/control over poisoning. Hence, irrespective of the backdoor attack, we always poison a model in the same fashion. We poison the OpenAI pretrained CLIP models [37] using different vision encoders (RN50, ViT-B/32) with a subset of CC3M [40] dataset. The default poisoning rate is set to 0.5% ($2k$ poisoned / $400k$ clean samples), and poisoning is done for 5 epochs. Unless specified otherwise, the target label is set to "banana", which is prevalent in literature [2, 30]. More details can be found in Appendix A, further results on ViT-L/14 and SigLip [52] are in Appendix B.

Cleaning. Unless specified otherwise, a set of $250k$ samples (similar to [2, 30]) disjoint of the poisoning set from the CC3M dataset is used for cleaning. Details on hyperparameters for CleanCLIP and how RoCLIP was adapted to

	BadNet				Blended								WaNet		BadCLIP	
	Rand. [18]		Stripes		Rand. [8]		Stripes		Triangles		Text		[35]		[30]	
Method	clean	ASR	clean	ASR	clean	ASR	clean	ASR	clean	ASR	clean	ASR	clean	ASR	clean	ASR
ImageNet																
CLIP	57.5	99.2	57.6	99.8	57.7	99.4	57.6	95.6	57.4	85.7	56.9	95.6	57.7	99.2	58.6	98.8
RoCLIP [49]	47.4	75.1	48.2	82.0	47.9	1.5	47.9	7.0	47.2	37.1	47.2	59.1	47.2	2.0	–	–
CleanCLIP [2]	53.0	14.5	53.0	62.3	53.4	19.5	53.1	61.8	52.9	48.7	53.3	42.4	52.9	**0.0**	53.8	40.1
PAR (ours)	53.3	**6.3**	53.0	**42.4**	53.6	**0.0**	53.5	**0.1**	52.9	**10.3**	53.4	**18.1**	54.4	**0.0**	53.4	**30.4**
COCO																
CLIP	73.3	99.4	73.1	99.9	73.2	99.3	73.2	98.2	73.1	98.2	73.2	97.4	72.5	99.8	73.9	99.8
RoCLIP [49]	66.4	51.4	66.4	75.9	66.9	6.7	66.6	**4.9**	66.3	84.6	66.2	49.1	66.2	13.0	–	–
CleanCLIP [2]	69.9	19.4	69.7	52.1	70.4	44.3	69.8	76.0	69.8	71.6	70.2	32.5	70.1	5.1	70.3	62.6
PAR (ours)	70.5	**6.5**	70.2	**20.9**	70.4	**1.6**	70.7	5.0	70.3	**24.1**	71.3	**11.4**	71.1	**2.3**	70.7	**31.2**

Table 1. **Comparing PAR with CleanCLIP and RoCLIP**. For RN50 based CLIP, we report clean and ASR (lower is better) performance for ImageNet zero-shot classification and COCO text retrieval. Attack-wise lowest ASR is in **bold** and the poisoned CLIP is highlighted.

fine-tuning setup can be found in Appendix A. For PAR, we use the setup defined in Sec. 5.2 and clean for 10 epochs, other details can be found in App. A.3. In Appendix B.2, we show that PAR also works effectively if small amount of backdoor data is still present in the $250k$ clean sample set.

Metrics. We evaluate all CLIP models for zero-shot classification accuracy on the test set of ImageNet at the standard 224×224 resolution. We report both clean accuracy and the attack success rate - ASR (percentage of samples with the backdoor trigger for which the model outputs the target "banana"). A task native to multi-modal models is retrieval. Hence, we also evaluate retrieval on $5k$ points from the validation set of COCO dataset, where we report clean top-5 text retrieval score and ASR - fraction of samples with the backdoor trigger for which the target is retrieved in any one of the top-5 retrieved captions.

5.2. Empirical considerations for the PAR loss

We choose τ, the only parameter in $\mathscr{L}_{\text{PAR}}$ by doing a sweep over it for BadNet-Stripes poisoned ResNet50 and use the same value for all other attacks/encoders. As we want to go away from init point (in weight space), we devise a custom schedule that helps us achieve this efficiently. We start with a high LR of $3e-5$ that decays to a value of $3e-6$ linearly over half of total cleaning epochs. For the remaining half, it goes down to $1e-9$ with a cosine schedule. However PAR also works effectively for the standard cosine decaying schedule as used in CleanCLIP, see Appendix B.

5.3. Evaluating PAR against diverse attacks

As CLIP with ResNet50 encoder has been used as standard test-bed for backdoor attacks and defenses, we first evaluate this. We test PAR, CleanCLIP and RoCLIP against different backdoor attacks in Tab. 1. PAR attains for all but one (BadCLIP) attack the best clean zero-shot ImageNet accuracy, whereas RoCLIP yields the worst. Importantly, PAR outperforms in all cases both CleanCLIP and RoCLIP in terms of ASR, and the difference between PAR and CleanCLIP is as high as 61% (Blended-Stripes). Even though RoCLIP cleans better than CleanCLIP, its clean performance is inferior. Similarly, PAR cleans better than CleanCLIP also for lower poisoning rates, see Fig. 5. For the recent BadCLIP attack, which assumes access to the model to be poisoned, PAR is very effective (since we use the poisoned model from [30] training RoCLIP in this setup was not possible). The same trend holds for top-5 retrieval for COCO, where PAR outperforms CleanCLIP by as much as 65% on ASR while maintaining similar clean performance. RoCLIP again yields the worst clean numbers. In Appendix A.3, we discuss a comparison to SafeCLIP [50].

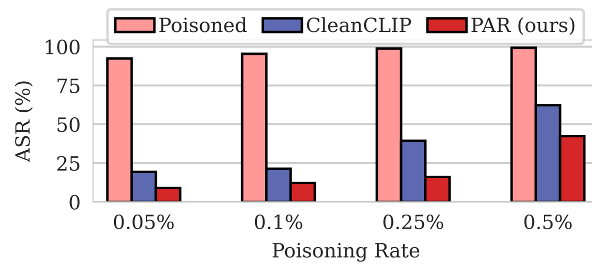


Figure 5. **ASR for different poisoning rates of CleanCLIP and PAR for RN50.** Even at a lower poisoning rate of 0.05%, BadNet-Stripes achieves 92% attack success rate (ASR). Overall across all poisoning rates, PAR cleans better than CleanCLIP.

Adaptive attacks. As an adaptive attack against PAR, we re-poison the model cleaned with PAR. The idea is that then the original backdoored model is a potential solution for PAR when cleaning the re-poisoned model. Tab. 17 in the Appendix shows that for Blended-Stripes PAR cleans the re-poisoned model perfectly and PAR does not "fall back" to the original backdoored model.

	Method	BadNet Rand. [18] clean	BadNet Rand. [18] ASR	BadNet Stripes clean	BadNet Stripes ASR	Blended Rand. [8] clean	Blended Rand. [8] ASR	Blended Stripes clean	Blended Stripes ASR	Blended Triangles clean	Blended Triangles ASR	Blended Text clean	Blended Text ASR	BadCLIP [30] clean	BadCLIP [30] ASR
ImageNet	CLIP	59.6	89.9	59.8	99.1	58.6	50.9	59.7	99.8	58.9	99.7	59.3	99.8	59.9	95.0
	CleanCLIP [2]	55.1	6.4	54.7	86.8	55.0	0.1	55.0	15.2	54.6	91.4	54.7	62.9	55.2	19.6
	PAR (ours)	55.5	**0.1**	54.4	**50.1**	55.5	**0.0**	54.6	**0.1**	54.9	**15.9**	56.1	**37.3**	54.7	**18.2**
COCO	CLIP	72.2	85.6	72.5	98.9	71.3	59.9	72.6	99.8	71.9	99.9	71.8	99.8	72.9	96.8
	CleanCLIP [2]	70.3	11.2	70.5	84.7	70.4	2.9	70.6	22.4	70.2	98.5	70.6	55.4	70.1	28.5
	PAR (ours)	69.4	**1.4**	69.5	**51.6**	69.4	**1.5**	69.4	**1.7**	70.0	**91.5**	70.2	**45.4**	70.6	**27.1**

Table 2. **Comparing PAR with CleanCLIP for ViT-B/32.**For ViT-B/32-CLIP, we report clean and ASR (lower is better) performance for ImageNet zero-shot classification and COCO text retrieval. Attack-wise lowest ASR is in **bold** and the poisoned CLIP is highlighted.

5.4. Changing the encoder

From Tab. 1, we omit the weakest attack in WaNet and evaluate the remaining ones for the ViT-B/32 based CLIP model. Training setup is the same as for RN50. We omit RoCLIP as it had the worst clean-ASR trade-off for RN50 and it is a method native to cleaning models trained from scratch. From Tab. 2 for ViT-B/32, PAR has again a better backdoor removal rate than CleanCLIP, improving by up to 37% for ASR on ImageNet while having similar clean performance. Even for COCO, PAR outperforms CleanCLIP comprehensively in terms of ASR while always having nearly similar clean performance. Thus PAR generalizes across different architectures. We emphasize that one can reduce the ASR significantly for PAR by increasing τ, as can be seen in Tab. 12. Moreover, in Appendix B, we show PAR also works for a vision-language model trained with SigLip [52] and larger models like ViT-L/14.

5.5. Effective utilization of synthetic data

Until now, we assumed access to 250k clean samples of image/caption pairs in CC3M. However, getting 250k clean samples would come at a significant cost as one needs to either *(i)* detect the backdoored data, a difficult task on its own, or *(ii)* label the data. Thus we explore if synthetic data is a viable solution to simplify the backdoor removal.

In [20], a synthetic (image, caption) dataset, termed SynthCLIP, is generated via text-to-image diffusion models [38] and used to train CLIP models at the price of slightly worse performance. We use a subset of 250k samples of SynthCLIP to clean poisoned models with PAR. Firstly, Fig. 3 shows that PAR with SynthCLIP (SynC) can completely clean a poisoned model with BadNet-Stripes, and yields a trade-off curve similar to using real data (CC3M), with slightly reduced clean performance. Moreover, in Tab. 3 we detail the backdoor removal rate of PAR with synthetic data. We use the same threshold and training parameters as before with 250k/500k synthetic samples. For both BadNet-Stripes and Blended-Text, PAR brings ImageNet ASR down significantly. The clean performance on ImageNet is also slightly degraded, probably due to the drastic data distribution shift from the pre-training on real data coupled with the learning schedule. Interestingly, we obtain no reduction or even an improvement in clean performance at much smaller ASR for retrieval on COCO, potentially due to better captions in SynthCLIP. Given the potential high cost of clean data, our results show that synthetic data is a cheap and effective replacement with strong performance of PAR.

Method	Data	Samples	ImageNet clean	ImageNet ASR	MS-COCO clean	MS-COCO ASR
Attack: BadNet-Stripes						
PAR	CC3M	250k	54.4	50.1	69.5	53.5
PAR	SynthCLIP	250k	50.0	15.9	71.0	22.6
PAR	SynthCLIP	500k	48.0	3.7	69.4	8.8
Attack: Blended-Text						
PAR	CC3M	250k	56.1	37.3	70.2	45.4
PAR	SynthCLIP	250k	49.5	5.9	70.2	15.1
PAR	SynthCLIP	500k	49.3	5.2	70.2	11.0

Table 3. **Utilizing synthetic data for cleaning.** We clean ViT-B/32 poisoned CLIP models using CC3M (CC) and SynthCLIP (SynC). PAR seems to effectively clean even with sythetic data.

6. Conclusion

We show that know cleaning methods like CleanCLIP, RoCLIP can be bypassed by using our novel structured trigger patterns, due to their over-reliance on strong data augmentations. To overcome this we introduce PAR, a simple, intuitive and effective backdoor removal technique based on fine-tuning. PAR cleans models from backdoors for all tested attacks across architectures/models effectively in comparison to the baselines. Finally, we show that synthetic data alone can effectively remove backdoor correlations in CLIP models, reducing the need for costly real data.

References

[1] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *AISTATS*, 2020. 2

[2] Hritik Bansal, Nishad Singhi, Yu Yang, Fan Yin, Aditya Grover, and Kai-Wei Chang. Cleanclip: Mitigating data poisoning attacks in multimodal contrastive learning. In *ICCV*, 2023. 1, 2, 3, 4, 6, 7, 8

[3] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019. 2, 1

[4] 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)*. IEEE, 2021. 2

[5] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. `https://github.com/kakaobrain/coyo-dataset`, 2022. 3

[6] Nicholas Carlini and Andreas Terzis. Poisoning and backdooring contrastive learning. In *ICLR*, 2022. 2

[7] 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. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2024. 1, 2, 3

[8] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017. 1, 2, 3, 7, 8, 4, 9

[9] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 1

[10] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, 2019. 2, 4

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[12] Terrance DeVries. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 2

[13] Khoa D Doan, Yingjie Lao, Peng Yang, and Ping Li. Defending backdoor attacks on vision transformer via patch processing. In *AAAI*, 2023. 2

[14] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1

[15] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. In *NeurIPS*, 2024. 1

[16] Kuofeng Gao, Yang Bai, Jindong Gu, Yong Yang, and Shu-Tao Xia. Backdoor defense via adaptively splitting poisoned dataset. In *CVPR*, 2023. 2

[17] Chenxi Gu, Chengsong Huang, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. Watermarking pre-trained language models with backdooring. *arXiv preprint arXiv:2210.07543*, 2022. 2

[18] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017. 1, 2, 3, 4, 7, 8, 6, 9

[19] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7, 2019. 2

[20] Hasan Abed Al Kader Hammoud, Hani Itani, Fabio Pizzati, Adel Bibi, and Bernard Ghanem. Synthclip: Are we ready for a fully synthetic clip training? In *Synthetic Data for Computer Vision Workshop@ CVPR*, 2024. 8, 1

[21] Dominik Hintersdorf, Lukas Struppek, Daniel Neider, and Kristian Kersting. Defending our privacy with backdoors. In *NeurIPS 2023 Workshop on Backdoors in Deep Learning - The Good, the Bad, and the Ugly*, 2024. 2

[22] Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. Openclip, 2021. 1

[23] 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 *ICML*, 2021. 1

[24] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022. 2

[25] Junhao Kuang, Siyuan Liang, Jiawei Liang, Kuanrong Liu, and Xiaochun Cao. Adversarial backdoor defense in clip. *arXiv preprint arXiv:2409.15968*, 2024. 3

[26] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 1

[27] Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 18(5), 2020. 2

[28] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *ICLR*, 2021. 2

[29] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *ICLR*, 2021. 2

[30] Siyuan Liang, Mingli Zhu, Aishan Liu, Baoyuan Wu, Xiaochun Cao, and Ee-Chien Chang. Badclip: Dual-embedding guided backdoor attack on multimodal contrastive learning. In *CVPR*, 2024. 1, 2, 5, 6, 7, 8, 4, 9

[31] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *CVPR*, 2024. 1, 5

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[33] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024. 1, 5

[34] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *ECCV*, 2020. 2

[35] Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *ICLR*, 2021. 2, 7, 1, 9

[36] Yuwei Niu, Shuo He, Qi Wei, Zongyu Wu, Feng Liu, and Lei Feng. Test-time multimodal backdoor detection by contrastive prompting. In *ICML*, 2025. 3

[37] 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 *ICML*, 2021. 1, 4, 6

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 8

[39] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 1, 3

[40] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 6

[41] Qwen Team. Introducing qwen1.5, 2024. 1

[42] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9 (86), 2008. 6

[43] Haonan Wang, Qianli Shen, Yao Tong, Yang Zhang, and Kenji Kawaguchi. The stronger the diffusion model, the easier the backdoor: Data poisoning to induce copyright breacheswithout adjusting finetuning pipeline. In *ICML*, 2024. 2

[44] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023. 1

[45] Yifei Wang, Wenhan Ma, Stefanie Jegelka, and Yisen Wang. How to craft backdoors with unlabeled data alone? In *ICLR 2024 Workshop on Navigating and Addressing Data Problems for Foundation Models*, 2024. 2

[46] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *ACL*, 2019. 4

[47] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *NeurIPS*, 2021. 2

[48] Yuan Xun, Siyuan Liang, Xiaojun Jia, Xinwei Liu, and Xiaochun Cao. Cleanerclip: Fine-grained counterfactual semantic augmentation for backdoor defense in contrastive learning. *arXiv:2409.17601v3*, 2024. 3

[49] Wenhan Yang, Jingdong Gao, and Baharan Mirzasoleiman. Robust contrastive language-image pretraining against data poisoning and backdoor attacks. In *NeurIPS*, 2023. 1, 2, 3, 7

[50] Wenhan Yang, Jingdong Gao, and Baharan Mirzasoleiman. Better safe than sorry: Pre-training clip against targeted data poisoning and backdoor attacks. In *ICML*, 2024. 2, 3, 7

[51] Ziqing Yang, Xinlei He, Zheng Li, Michael Backes, Mathias Humbert, Pascal Berrang, and Yang Zhang. Data poisoning attacks against multimodal encoders. In *ICML*, 2023. 2

[52] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 6, 8

[53] Zhifang Zhang, Shuo He, Bingquan Shen, and Lei Feng. Defending multimodal backdoored models by repulsive visual prompt tuning. *arXiv:2412.20392*, 2024. 3

[54] Mingli Zhu, Shaokui Wei, Li Shen, Yanbo Fan, and Baoyuan Wu. Enhancing fine-tuning based backdoor defense with sharpness-aware minimization. In *ICCV*, 2023. 2

[55] Mingli Zhu, Shaokui Wei, Hongyuan Zha, and Baoyuan Wu. Neural polarizer: A lightweight and effective backdoor defense via purifying poisoned features. In *NeurIPS*, 2024. 2

Contents

A. Experimental Details and Discussions

In this section we detail the setup related to all the experiments conducted in this work. We detail how we select training hyperparameters like batch size (BS), learning rate (LR), datasets used, optimizer, etc., for poisoning and cleaning across methods and models. All experiments were conducted on 4-8 A100 GPUs.

A.1. Dataset

Poisoning. For poisoning and cleaning we always use the CC3M dataset. For poisoning (adding backdoor to the model), we use $400k$ samples out of which $2k$ are poisoned. For experiments involving lower or higher poison rate only the number of poisoned samples was changed accordingly. The target caption associated with the trigger appends the target label to a template for *e.g.* "an image of {target-label}" with the caption template randomly sampled from 80 zero-shot templates from [22].

Cleaning. We use $250k$ samples also from CC3M, disjoint of the poisoning set. For experiments with synthetic data, we use the respective sized subsets from SynthCLIP dataset [20]. For evaluation, we use the full validation set of ImageNet [11] and $5k$ samples from the COCO2014 [32] validation split. For poisoning, cleaning and evaluation of RN50 and ViT-B/32 based CLIP models we use the 224×224 resolution, whereas for ViT-L/14 we employ a higher resolution of 336×336.

A.2. Poisoning

Irrespective of the model/attack, the poisoning is always carried out with a fixed cosine-decay schedule with peak LR $= 1e-5$ trained for 5 epochs with BS $= 256$. We employ the AdamW optimizer with default `PyTorch` hyperparameters and a weight decay of $1e-4$ and optimize with the standard CLIP loss ($\mathscr{L}_{\text{CLIP}}$). All poisoning for ViT-B/32 and Resnet50 is done at the resolution of 224×224. For BadNet-Stripes we use a patch of size 16×16 as is common in literature [2, 30]. For other proposed triggers, we follow the setup listed in Sec. 3.2.

For known attacks (BadNet [18], Blended [8], SIG [3] and WaNet [35]), we follow the attack specific parameters from [2]. In BadCLIP the patch (16×16) is optimized on the model and the poisoning is also done with a specific training setup - this is contrary to our setup where we assume no knowledge/control over poison training process. For RN50, we use the patch and model from the their Github repository.[1] As BadCLIP authors do not show results for ViT based CLIP in their work, we optimize the trigger patch using their code with $5k$ auxiliary samples from CC3M for 50 epochs with a batch size of 128 and poison with our schedule for 5 epochs.

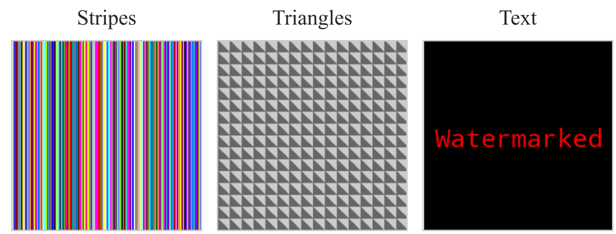


Figure 6. **Visualizing the proposed triggers.** We visualize the proposed structured triggers. For BadNet-Stripes we use the "Stripes" trigger as a patch. For Blended-Stripes the "Stripes" trigger is overlayed on the full images with $n_c = 0.03$ in Eq. (1). "Triangles" and "Text" triggers are also overlayed on the original image as described in Sec. 3.2.

For poisoning ViT-L based CLIP model, we use a 1% poisoning rate and use a higher resolution of 336×336 as downstream use of ViT-L/14 CLIP for *e.g.* in LVLMs (LLaVa, CogVLM) is often done at this higher resolution. For BadNet-Stripes we use a patch of size 32×32. For Blended-Text, we scale the text "Watermarked" to span across the width of the image and use $n_c = 0.5$ in Eq. (1). We keep the LR and schedule same as before and poison for 8 epochs. We found poisoning with 5 epochs was not as effective as for RN50 or ViT-B/32 based CLIP.

A.3. Cleaning methods

For all (CleanCLIP, RoCLIP, PAR) cleaning methods considered in this work, we optimize the hyper-parameters on a BadNet-Stripes poisoned RN50 and use these parameters across all other encoders and attacks. In this section we detail how hyper-parameter selection was done for different cleaning methods and related discussions.

CleanCLIP

Originally, CleanCLIP uses a BS of 64 and fine-tune with $LR = 1e-5$. Instead, we do a short sweep over LR, epochs and number of epochs in Tab. 4. We select the highlighted row with LR $= 2e-5$ and BS $= 256$ as it has better ASR while maintaining good clean performance. We use their default values[2] for all other parameters (augmentations, loss weight, etc.) and train for 5 epochs. For every CleanCLIP run with ResNet50 and ViT-B/32 based CLIP, we stick to this setup. For ViT-L/14 based CLIP runs, we reduce the

[1] https://github.com/LiangSiyuan21/BadCLIP
[2] https://github.com/nishadsinghi/CleanCLIP

Epochs	LR	BS	ImageNet clean	ImageNet ASR (↓)	MS-COCO clean	MS-COCO ASR (↓)
5	1e-5	256	55.5	89.1	71.6	85.6
5	2e-5	256	53.0	62.3	69.7	52.1
5	3e-5	256	49.8	16.8	67.2	18.2
5	2e-5	512	54.6	79.3	70.8	71.3
10	2e-5	256	49.7	33.0	67.8	31.4

Table 4. **Optimizing CleanCLIP.** Looking at BadNet-Stripes performance for different training setups for RN50. The selected parameters are highlighted.

BS to 120 and train for 3 epochs while adapting the schedule to 3 epochs. Reducing the number of epochs keeps the number of updates similar to the optimized schedule.

Discussion regarding CleanCLIP results in [30]. For the BadCLIP attack against ResNet50-CLIP, the authors in [30] optimized CleanCLIP's hyper-parameters. Their parameters attain worse performance than what our optimized parameters get. Specifically, their run yields (clean, ASR) of (54.0%, 89.6%) while our found parameters (highlighted row in Tab. 4) yield much better (53.8%, 40.1%) performance. The clean accuracy we get is nearly the same as in [30] whereas we reduce ASR by more than 50%. This justifies our chosen hyperparameters for CleanCLIP.

CleanCLIP does not clean due to its loss. In Tab. 4 for the $LR = 3e-5$ setting, we show that CleanCLIP can bring the ASR significantly down at the cost of clean performance. This happens as the higher LR leads the model parameters to move away from the original poisoned model in some direction in the weight space and not because of the additional $\mathscr{L}_{\text{UniAug}}$ term in their $\mathscr{L}_{\text{CleanCLIP}}$. If $\mathscr{L}_{\text{UniAug}}$ term had an effect on cleaning, it would show up in Fig. 3 where we have used the λ values as high as 1000. Thus we infer that effective cleaning with CleanCLIP is only possible by destroying the original model and not due to their proposed loss. On the contrary, $\mathscr{L}_{\text{PAR}}$ loss cleans by a structured deviation from the poisoned model while preserving clean performance that can be controlled by τ.

Method	LR	$\mathscr{K}$	ImageNet clean	ImageNet ASR (↓)	MS-COCO clean	MS-COCO ASR (↓)
Init: Pre-trained CLIP						
RoCLIP	5e-6	3	48.2	82.0	66.4	75.9
RoCLIP	5e-6	2	47.9	83.2	66.9	69.9
RoCLIP	2e-5	3	40.7	45.0	60.7	45.9

Table 5. **Searching for optimal RoCLIP parameters.** Looking at BadNet-Stripes performance for different training setups for RN50 based CLIP. Unlike [49], we start with a pre-trained model in accordance to our setup. The selected parameters are highlighted.

CleanCLIP when structured trigger is added to its set of augmentations. In Tab. 15 we analyze if CleanCLIP can clean a backdoor attack if one adds the trigger (in this case "stripes") to the augmentation set. Indeed, in this case CleanCLIP works well even for the Blended-Stripes model which the standard version of CleanCLIP does not clean well. However, CleanCLIP still does not work for the other structured trigger Blended-Text. Thus there is no generalization across structured triggers. Moreover, it is clearly trivial to clean a model from a known backdoor trigger. In practice the attacker is free to use any backdoor trigger and thus this setting is not at all realistic. In contrast, PAR is independent of the structure of the backdoor trigger and works well across all employed triggers.

Attack	Stripes Aug.	ImageNet clean	ImageNet ASR	COCO clean	COCO ASR
Blended-Stripes	✗	53.1	61.8	69.8	76.0
Blended-Stripes	✓	53.3	0.1	70.2	1.4
Blended-Text	✗	53.3	42.4	70.2	32.5
Blended-Text	✓	52.9	34.6	70.5	27.2

Table 6. **Stripes as augmentation in CleanCLIP.** The Blended-Stripes model has a low ASR when stripes is used as augmentation (Aug. ✓) in comparison to the non-augmented version (Aug. ✗).

RoCLIP

Originally proposed for cleaning from scratch, we adapt RoCLIP [49] to the fine-tuning setup. For this, we use their proposed[3] BS, data augmentations, size of retrieval pool, learning rate schedule and number of epochs (24) trained for. We adapt their setup by fine-tuning with their method starting from a pre-trained CLIP-RN50. As RoCLIP cleans with poisoned dataset, we keep the standard 0.5% poisoning rate CC3M dataset ($2k/400k$) for training with it. Note, this means RoCLIP is effectively cleaning with an additional $150k$ samples in comparison to CleanCLIP and PAR (as in this work, both of the latter use $250k$ samples to clean a poisoned model).

To select the optimal LR and $\mathscr{K}$, the parameter that controls the frequency of epochs where the retrieval pool based loss is used, we do a sweep over some values in Tab. 5. A higher $\mathscr{K}$ would increase clean performance at the cost of ASR. As $\mathscr{K} = 3$ has 82% ASR on ImageNet, we do not increase it further. Inversely, a higher LR cleans better (lower ASR) at the cost of clean accuracy, hence we do not increase

[3]https://github.com/BigML-CS-UCLA/RoCLIP

LR beyond $2e-5$, as it already degraded clean accuracy a lot. Models trained with $LR < 5e-6$, yield models with very high ASR ($\sim 85\%$) and we omit those values as well. Finally, we select the highlighted row as it achieves the best clean-ASR trade-off.

SafeCLIP

SafeCLIP [50] is a method for training *from scratch* with a poisoned dataset, while PAR focuses on cleaning a backdoored *pre-trained* CLIP. We tried to adapt SafeCLIP to our finetuning setting which is non-trivial as the method is inherently a method for training from scratch. We could not achieve reasonable clean performance. For instance, applying SafeCLIP on a Blended-Stripes poisoned model, we get only 39% clean accuracy on ImageNet with ASR of 41%, in comparison to 53.5% clean and 0.1% ASR with PAR. Using SafeCLIP poisoned images (poisoned models were not available but the authors provided us kindly with the poisoned training images), we trained a model from scratch in their setup (1M samples from CC3M, 0.05% poisoning rate) with Blended. This poisoned model yields a clean performance on ImageNet of 5.9% and has an ASR of 91.3%. On cleaning this model with the default PAR setup, we could completely get rid of backdoors with an ASR of 0.0% and even increase slightly the clean performance to **6.1%**. This showcases the effectiveness of PAR against models poisoned from scratch. Thus PAR can always be applied after training a (poisoned) model from scratch while we could not adapt SafeCLIP for cleaning via finetuning (in contrast to RoCLIP).

Perturb and Recover (PAR)

We use a custom learning schedule as highlighted in the main part. We emphasize that for the standard cosine decayed LR schedule as used by CleanCLIP, our method still outperforms CleanCLIP, see Tab. 11. All other details specific to PAR can be found in Tab. 7. As it can be noted from Tab. 7, we train for half the number of epochs for ViT-L/14 as the BS is halved and this keeps the effective number of updates of the model similar. Even though we fix the threshold τ to 2.15 for all our experiments, one can choose the optimal threshold to control the clean accuracy-ASR trade-off with our method, using curves like the one in Fig. 3. As we have two data-augmentations applied with a certain probability in PAR, we look at the variance across a few attacks in Tab. 10. The variance is marginal across attacks and evaluated metrics.

Computational Costs

We define one unit of computation as a complete forward and backward pass through both image and text encoders. Our method, PAR requires 1.5 units of computation, consisting of one full forward and backward pass (1 unit) plus an additional forward pass through the frozen poisoned encoders (0.5 units). Similarly, CleanCLIP also requires 1.5 units, comprised of one full forward and backward pass (1 unit) plus one additional forward pass with augmented image and text data (0.5 units). In contrast, standard CLIP training uses only one forward and one backward pass, totaling 1 unit of computation. Therefore, both PAR and CleanCLIP have identical computational requirements, which amount to 1.5 times the cost of standard CLIP training. RoCLIP and SafeCLIP are both methods designed for training from scratch and partially involve more complex steps so that it is difficult to compare their cost to CleanCLIP and PAR.

	Configuration	RN50/ViT-B/32	ViT-L/14
DATA	Data	CC3M/SynthCLIP	CC3M
	Image size	224x224	336x336
	Gaussian Noise	Std = 0.2 Prob = 0.5	Std = 0.2 Prob = 0.5
	CutOut-Patch	Area = (0.5 - 1)% Prob = 0.5	Area = (0.5 - 1)% Prob = 0.5
TRAINING & LOSS	Threshold (τ)	2.15	2.15
	Optimizer	AdamW	AdamW
	Start-LR	3e-5	3e-5
	Peak-LR	3e-6	3e-6
	Final-LR	1e-9	1e-9
	Peak-epoch	5	2.5
	Epochs	10	5
	Weight decay	1e-4	1e-4
	Batch size	512x4	120x8
	Momentum	0.9, 0.999	0.9, 0.999

Table 7. **Training and data configurations.** The specific training settings for different models trained in this work using PAR. We differentiate between small encoders in CLIP like RN50/ViT-B and the large one in ViT-L/14.

B. Additional Experiments

In this section, we expand on additional experiments that highlight the effectiveness of both the proposed triggers and backdoor removal method PAR.

B.1. Effectiveness of the proposed triggers

Bypassing CleanCLIP. In Tab. 8, we replace the standard random noise in the Blended attacks with the proposed triggers: Stripes, Triangles and Text. For both RN50 and ViT-B/32 based CLIP, the proposed triggers are as effective as random noise in terms of backdoor attack success rate (ASR). While CleanCLIP reduces ASR for the original random pattern very well, for the structured patterns the ASR is in most cases still very high. This shows how the proposed triggers are more effective against strong augmentation based cleaning methods like CleanCLIP than random noise. In Fig. 7, we show more visualizations with the proposed triggers.

The effectiveness of the proposed triggers can also be visualized via t-SNE projections of the embedding of the image encoder in trained CLIP models in the "T-SNE EMBEDDINGS" blocks in Figures 9 and 10. For the poisoned CLIP the embeddings of original and backdoored images form disjoint clusters, indicating the model differentiates among these. After CleanCLIP, the embeddings in Figures 9 and 10 are still disjoint. But, the same CleanCLIP embeddings for the random noise based BadNet in Fig. 8 are homogeneously distributed. This clearly points to the fact that random noised trigger based attacks are easily cleaned by methods like CleanCLIP, which is not the case when one transitions to structured trigger based attacks.

Other target labels. In Tab. 9, we show for RN50 based CLIP the proposed triggers work across different target labels. Specifically, in addition to the target "banana", we show for ImageNet classes "refrigerator" and "strawberry" BadNet-Stripes attains high ASR while poisoning CLIP and bypasses CleanCLIP as well. Note, for the label "strawberry" we cannot provide an ASR for COCO retrieval task as there are no captions with "strawberry" in the validation set we use.

Lower poisoning rate. In the pink bars in Fig. 5, we plot the ASR as attained by BadNet-Stripes across a range of poisoning rate. Even at a meager poisoning rate of 0.05%, BadNet-Stripes achieves an impressive 92% success rate.

Attack.	n_c	Source	CLIP clean	CLIP ASR (↑)	CleanCLIP clean	CleanCLIP ASR (↑)
CLIP encoder: ResNet-50						
Random	0.2	[8]	57.7	99.4	53.4	19.5
Stripes	0.03	ours	57.6	95.6	53.1	61.8
Triangles	0.15	ours	57.4	85.7	52.9	48.7
Text	0.5	ours	56.9	95.6	53.2	42.4
CLIP encoder: ViT-B/32						
Random	0.2	[8]	59.3	51.5	54.3	0.0
Stripes	0.03	ours	59.7	99.8	55.0	15.2
Triangles	0.15	ours	58.9	99.7	54.6	91.4
Text	0.5	ours	59.3	99.8	54.7	62.9

Table 8. **Structured trigger patterns are more effective than random noise.** For the Blended [8] attack model, we replace the original random noise with our structured patterns. We evaluate for both RN50 and ViT-B/32 based CLIP for the zero-shot classification for ImageNet.

Method	ImageNet clean	ImageNet ASR (↓)	MS-COCO clean	MS-COCO ASR (↓)
Target: banana				
CLIP	57.6	99.8	73.1	99.9
CleanCLIP	53.0	62.3	69.7	52.1
PAR	53.0	**42.4**	70.2	**20.9**
Target: refrigerator				
CLIP	57.5	99.2	73.2	98.8
CleanCLIP	53.4	37.7	69.9	25.9
PAR	54.0	**34.4**	72.3	**13.8**
Target: strawberry				
CLIP	57.6	99.2	72.9	-
CleanCLIP	53.6	72.9	69.6	-
PAR	54.0	**28.8**	71.4	-

Table 9. **Proposed BadNet-Stripes and PAR work across target labels.** For the target "strawberry" we can't provide an ASR for COCO as "strawberry" is not present in any of the $5k$ validation samples we use. The original poisoned CLIP model is highlighted.

On increasing the poisoning rate, the ASR very quickly goes to 99%.

B.2. Backdoor removal with PAR

Effectiveness at lower poisoning rates. As prior works [2, 30] use a poisoning rate of 0.3%, we plot in Fig. 5 ASR attained at poisoning rate as low as 0.05%: across all tested poisoning rates, PAR outperforms CleanCLIP. Specifically, at 0.25% poisoning rate CleanCLIP has a ASR of 40% whereas PAR attains a significantly lower ASR of 19%.

PAR works for general LR schedules. As mentioned in the main part, to speed up cleaning with PAR we use a custom learning schedule. To disentangle the effect of this schedule from the novel loss (Eq. (5)) term we propose, we clean with $\mathscr{L}_{\text{PAR}}$ in the cosine schedule of CleanCLIP and present the results in Tab. 11. For both BadNet-Stripes

Attack	ImageNet clean	ImageNet ASR (↓)	COCO clean	COCO ASR (↓)
Blended-Stripes	53.5±0.2	0.1±0.1	70.7±0.1	5.0±0.4
BadNet-Stripes	53.0±0.6	42.4±6.1	70.2±0.2	20.9±3.2
BadCLIP	53.4±0.4	30.4±3.1	70.7±0.6	31.2±2.4

Table 10. **Variance of PAR.** We compute the mean and standard deviations across 3 runs for PAR. Overall the variance across runs and attacks is marginal.

and Blended-Stripes, PAR cleans way better than CleanCLIP. For Blended-Stripes, PAR even achieves a 2% clean accuracy improvement over CleanCLIP.

Scaling PAR to ViT-L/14. In Tab. 13, we show the effectiveness of PAR for the ViT-L/14 based CLIP model. Specifically, we poison the model with BadNet-Stripes and Blended-Text attacks at a higher resolution of 336×336. As mentioned in Appendix A, we poisoned the models for 3 additional epochs (8 epochs in total) in comparison to ViT-B poisoning as achieving ASR $> 95\%$ on ImageNet for both attacks was not possible with 5 epochs. In this setting as well, PAR significantly outperforms CleanCLIP with a marginal degradation in clean performance. This result is very promising as larger resolutions of 336×336 pixels are often used in downstream LVLMs *e.g.* LLaVa [33] and VILA [31]. However, our threat model of poisoning CLIP models with backdoored images-caption pairs does not allow a simple transfer to LVLMs as these models only use the vision encoder of the CLIP models. Without the text encoder, the backdoor is not effective.

PAR for SigLIP. In Tab. 14, we show the effectiveness of PAR for the ViT-B/16 SigLIP model. We poison the model with BadNet-Stripes and Blended-Text attacks. As the loss of SigLIP is working on a different scale, we select the larger τ of 3.4 without employing a grid search as before which is likely suboptimal. PAR outperforms CleanCLIP also for this other type of CLIP models with improvements of 56.7% in ASR for Blended-Text and 20.2% in ASR for BadNet-Stripes for zero-shot ImageNet classification while maintaining higher clean accuracy.

Trade-off for ViT-B/32. While in Fig. 3 we plot the clean accuracy and ASR trade-off achieved by τ parameter in PAR for ResNet50 based CLIP, one can do the same to select the optimal τ for other models. In Tab. 12 we evaluate for an additional value of τ against a subset of attacks.

	ImageNet		MS-COCO	
Method	clean	ASR (↓)	clean	ASR (↓)
Attack: BadNet-Stripes				
CleanCLIP	53.0	62.3	69.7	52.1
PAR	53.1	**51.5**	69.3	**29.8**
Attack: Blended-Stripes				
CleanCLIP	53.1	61.8	69.8	76.0
PAR	55.1	**25.4**	70.3	**17.4**

Table 11. **Learning rate schedule has little effect on performance.** We train with PAR using the schedule from CleanCLIP with peak-LR set to 2e-5.

		ImageNet		MS-COCO	
Method	τ	clean	ASR(↓)	clean	ASR(↓)
Attack: BadNet-Stripes					
CleanCLIP	–	54.7	86.8	70.5	84.7
PAR	2.15	54.4	50.1	69.5	53.5
PAR	2.25	53.4	44.2	68.6	48.8
PAR	2.3	52.3	40.6	65.0	42.3
Attack: Blended-Triangles					
CleanCLIP	–	54.6	91.4	70.2	98.5
PAR	2.15	54.9	15.6	70.2	91.5
PAR	2.25	54.4	14.7	68.3	94.0
PAR	2.3	53.8	12.1	67.6	89.2

Table 12. **Sweeping over τ for ViT-B/32 for clean-ASR trade-off.** Numbers are shown for PAR with the fixed value of 2.15 in the main part and one more value for the two architectures each. For easier comparison, CleanCLIP results are also illustrated.

		BadNet-Stripes		Blended-Text	
Eval Data	Method	clean	ASR	clean	ASR
ImageNet	CLIP	73.5	100.0	73.7	99.3
ImageNet	CleanCLIP	68.7	84.5	68.9	83.4
ImageNet	PAR (ours)	65.8	**0.0**	65.2	**17.0**
COCO	CLIP	80.5	22.6	79.9	97.7
COCO	CleanCLIP	78.8	6.3	78.7	65.1
COCO	PAR (ours)	76.7	**2.1**	76.0	**10.2**

Table 13. **Testing the attack and defense on ViT-L/14-336.** We scale PAR to the larger ViT-L/14. The poisoning and cleaning is done at the higher resolution of 336×336.

		ImageNet		COCO	
Attack	Model	clean	ASR	clean	ASR
Blended-Text	Poisoned	74.7	99.4	85.1	99.5
Blended-Text	CleanCLIP	68.4	87.6	79.9	90.1
Blended-Text	PAR	69.3	**30.9**	72.7	**60.4**
BadNet-Stripes	Poisoned	74.8	99.8	85.0	97.8
BadNet-Stripes	CleanCLIP	70.2	56.8	80.4	61.8
BadNet-Stripes	PAR	70.1	**36.6**	74.4	**60.1**

Table 14. **Other models.** We poison **SigLIP** (ViT-B/16) model with the same setup as CLIP models. Due to the loss of SigLIP being at a different scale, we select a higher τ of 3.4 (without grid-search). The original poisoned SigLip model is highlighted.

As expected on increasing τ from 2.15 to 2.25 we further reduce the ASR with a slight degradation in clean performance. This points to the fact that one would get a trade-off plot similar to the one for ResNet50 cleaned by PAR.

If clean data has some backdoored samples?. We test what happens if the clean data has small amount of backdoored samples in it. For this, we add 50 Blended-Stripes poisoned images to our 250k fine-tuning set (RN50 poisoned model). Again, PAR shows effective cleaning: we get ASR for ImageNet down from 95.6% to 10.7% with clean performance of 50.8% and ASR for retrieval down from 98.2% to 11.4%.

PAR for a non-backdoored model. To test how PAR works when the model does not have any backdoor trigger, we fine-tune a clean (non-poisoned) CLIP RN50 with PAR and get 54.3% and 70.0% clean accuracy on ImageNet and COCO respectively, compared to the clean model's 59.6% and 72.8%. Such drop is similar to what observed when cleaning poisoned models, see Tab. 1, and can be controlled by adjusting τ in Fig. 3.

Effect of data augmentations used in PAR. We report the performance of PAR with different augmentations in Table 15 (RN50, Blended-Stripes attack). While using a single augmentation already leads to good accuracy and backdoor removal, combining methods yields further improvements. From this ablation, we see that the selected augmentations with CutOut and Gaussian noise achieves the best backdoor removal-performance tradeoff.

Effect of number of clean real samples on PAR. In Tab. 16 we reduce the clean training samples from CC3M: as expected, more data helps both accuracy and backdoor removal, PAR is already effective with as few as 100k samples for BadNet and 50k samples for the Blended attacks.

B.3. Evaluating COCO retrieval

For COCO retrieval, until now we focused on top-5 clean text retrieval. In Tab. 18 and Tab. 19, we show how CleanCLIP and PAR fare against different backdoor attacks for standard top-1 image and text retrieval for RN50 and ViT-B/32. Across all attacks for ResNet50 based CLIP, PAR achieves a higher retrieval rate when the inputs are backdoored ("bkd" column in Tab. 18). The clean image retrieval performance of PAR is marginally lower than CleanCLIP on average whereas text retrieval is always better. Overall the drop in clean performance from the original poisoned CLIP (highlighted row in table) to PAR is marginal. In general, a similar trend holds for ViT-B/32 in Tab. 19.

	ImageNet		COCO	
Augmentations in PAR	clean	ASR	clean	ASR
CutOut	52.4	18.7	66.8	12.4
Gaussian Noise	50.9	0.2	67.3	4.2
CutOut + Uniform Noise	50.7	1.0	66.6	**4.0**
CutOut + Gaussian Noise	**53.5**	**0.1**	**70.7**	5.0

Table 15. **Performance of PAR varying augmentations.** Using just a single augmentation is enough for PAR to work, but a combination of CutOut + Gaussian Noise, yields the best backdoor removal/clean performance tradeoff.

			ImageNet		COCO	
Attack	Model	Samples	clean	ASR	clean	ASR
BadNet-Stripes	CLIP	-	57.6	99.8	73.1	99.9
	PAR	50k	48.4	76.8	69.3	64.4
	PAR	100k	48.8	56.0	66.6	22.0
	PAR	250k	53.0	42.4	70.2	20.9
Blended-Stripes	CLIP	-	57.6	95.6	73.2	98.2
	PAR	50k	48.4	13.4	68.7	7.0
	PAR	100k	48.1	0.5	65.6	4.0
	PAR	250k	53.5	0.1	70.7	5.0

Table 16. **Performance of PAR varying clean training samples.** PAR is still effective when having access to as low as 50k clean samples across different attacks.

	Poison	PAR	Re-Poison	PAR on Re-poison
clean	58.6	53.5	53.7	48.2
ASR	98.6	0.1	98.0	0.1

Table 17. **Adaptive attack on PAR.** PAR cleans the re-poisoned model perfectly, despite the original backdoored model being a feasible solution.

C. More Visualizations

The proposed triggers are visualized in Fig. 6 and the subsequently generated backdoored images are in Fig. 7. Training dynamics of PAR for a few more attacks along with the final image embedding projections via t-SNE can be found in Figures 9 and 10. t-SNE embedding comparison for the standard random noise based BadNet [18] is in Fig. 8.

Figure 7. **Visualizing more images with known and proposed triggers.** Standard BadNet [18] and Blended [8] use Gaussian noise as a trigger, we replace the noise with random stripped pattern for BadNet termed BadNet-Stripes. For the Blended attack, we further replace the random noise with stripes, low contrast triangles (Blended-Triangles) and "Watermarked" text (Blended-Text). *Note: this is a very small subset of possible structured patterns, and we believe similar other patterns would be equally effective*.

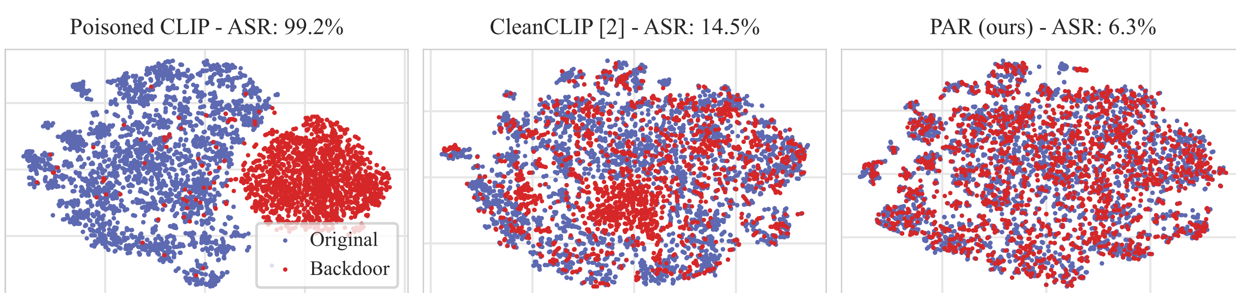


Figure 8. **Visualizing the embeddings of different models for BadNet [18] poisoned RN50.** We visualize the t-SNE projections of random noise based BadNet poisoned CLIP, clean fine-tuned by CleanCLIP and fine-tuned by PAR. In this case, CleanCLIP embeddings are much more homogeneously distributed in comparison to the proposed attacks with structured patterns. This shows that for random noise based triggers, CleanCLIP can be effective. Overall the spread of points achieved by PAR is most homogeneous which reflects in the lowest ASR.

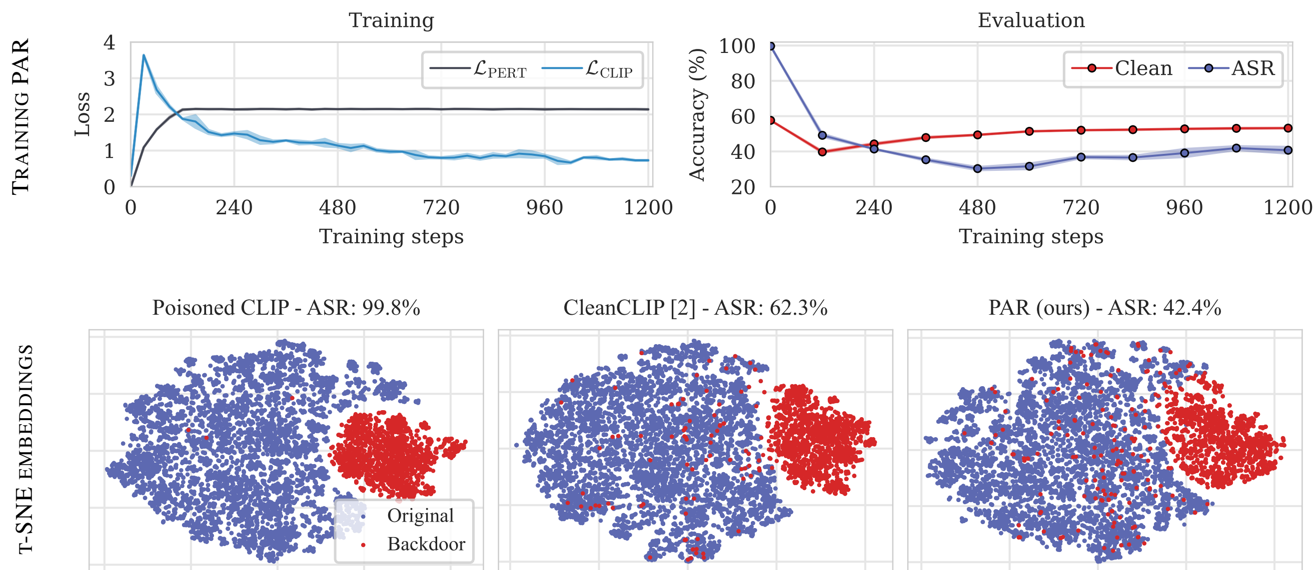


Figure 9. **Training dynamics and visualizing the embeddings of different models for BadNet-Stripes poisoned RN50.** In the top left plot, we show how the $\mathscr{L}_{\text{CLIP}}$ and $\mathscr{L}_{\text{PERT}}$ ($\tau = 2.15$) loss terms develop over training steps (evaluated every 25 steps) for BadNet-Stripes poisoned RN50. In the top right plot, we see how the training schedule generalizes by plotting clean accuracy and ASR (evaluated on $10k$ samples from ImageNet). In the bottom row, we visualize the t-SNE projections of the same BadNet-Stripes poisoned CLIP, clean finetuned by CleanCLIP and finetuned by PAR. Overall PAR yields the best mixing of clean and backdoored samples. Better mix means the model sees the clean and backdoored samples similarly, which also translates to low ASR.

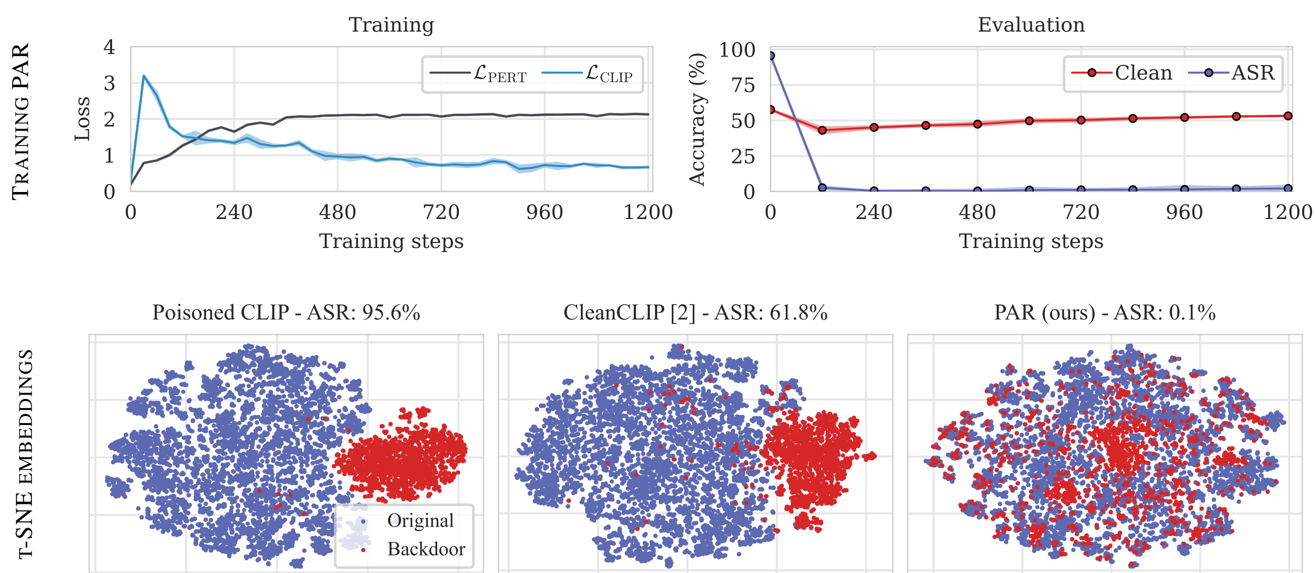


Figure 10. **Training dynamics and visualizing the embeddings of different models for Blended-Stripes poisoned RN50.** In the top left plot, we show how the $\mathscr{L}_{\text{CLIP}}$ and $\mathscr{L}_{\text{PERT}}$ ($\tau = 2.15$) loss terms develop over training steps (evaluated every 25 steps) for Blended-Stripes poisoned RN50. Even though the schedule was optimized for BadNet-Stripes poisoned RN50, in the top right plot, we see how the training schedule generalizes by plotting clean accuracy and ASR (evaluated on $10k$ samples from ImageNet). In the bottom row, we visualize the t-SNE projections of the same Blended-Stripes poisoned CLIP, clean finetuned by CleanCLIP and finetuned by PAR. Overall PAR yields the best mixing of clean and backdoored samples. Better mix means the model sees the clean and backdoored samples similarly, which also translates to low ASR.

Method	COCO Precision@Top-1 Image ret. clean	Image ret. bkd (↑)	Text ret. clean	Text ret. bkd (↑)
Attack: BadNet [18]				
CLIP	33.0	4.0	47.9	1.1
CleanCLIP	31.6	23.2	44.3	32.5
PAR	29.8	**26.4**	44.4	**37.4**
Attack: BadNet-Stripes				
CLIP	33.1	7.4	48.0	0.3
CleanCLIP	31.6	15.5	43.8	14.8
PAR	30.3	**18.0**	44.7	**21.8**
Attack: Blended [8]				
CLIP	32.8	6.0	47.7	0.5
CleanCLIP	31.8	17.1	44.3	18.4
PAR	30.9	**25.8**	44.9	**39.6**
Attack: Blended-Stripes				
CLIP	32.8	3.7	46.9	0.9
CleanCLIP	31.6	9.6	44.5	8.6
PAR	30.5	**18.9**	44.0	**27.9**
Attack: WaNet [35]				
CLIP	32.6	2.5	46.9	0.0
CleanCLIP	31.1	**18.2**	43.2	23.1
PAR	30.8	16.3	45.3	**24.7**
Attack: BadCLIP [30]				
CLIP	32.6	2.3	48.1	0.4
CleanCLIP	31.3	15.4	44.7	16.3
PAR	30.1	**20.1**	45.3	**23.2**

Table 18. **COCO retrieval for ResNet50 based CLIP.** In compliment to Tab. 1, we report Precision@Top-1 for both image-to-text and text-to-image retrieval for the COCO validation set. "bkd" shows when the inputs are with the backdoor trigger. The original poisoned CLIP model is highlighted.

Method	COCO Precision@Top-1 Image ret. clean	Image ret. bkd (↑)	Text ret. clean	Text ret. bkd (↑)
Attack: BadNet-Stripes				
CLIP	33.5	1.7	47.8	1.1
CleanCLIP	32.0	7.6	45.3	5.4
PAR	30.8	**13.7**	44.6	**17.6**
Attack: Blended-Stripes				
CLIP	33.3	5.9	47.4	0.1
CleanCLIP	32.4	21.6	45.9	27.5
PAR	30.5	**26.1**	39.9	**36.6**
Attack: Blended-Text				
CLIP	32.8	4.3	45.5	0.1
CleanCLIP	31.7	**15.1**	44.4	14.0
PAR	31.8	12.5	45.2	**18.3**
Attack: Blended-Triangles				
CLIP	32.9	3.1	47.0	0.0
CleanCLIP	32.2	2.5	44.6	0.6
PAR	31.1	**3.2**	43.4	**2.1**
Attack: BadCLIP [30]				
CLIP	33.5	2.3	48.0	0.0
CleanCLIP	32.1	**18.9**	45.6	**20.8**
PAR	31.1	15.6	43.6	16.6

Table 19. **COCO retrieval for ViT-B/32 based CLIP.** In compliment to Tab. 2, we report Precision@Top-1 for both image-to-text and text-to-image retrieval for the COCO validation set. "bkd" shows when the inputs are with the backdoor trigger. The original poisoned CLIP model is highlighted.