

Algorithm 1: CFBD

```

input : Image encoder  $f_I$ ; text encoder  $f_T$ ; pre-trained mapping network  $F$ ; training epochs
         $T$ ; training dataset  $\mathcal{D} = \{(I_i, T_i)\}_{i=1}^N$ ; poison subset number  $q$ ; probability threshold
         $\gamma$ ;  $z$  threshold  $\gamma_z$ ; number of distributions in GMM  $K$ .
output: Identified benign pairs  $\{\mathcal{D}_{sb}, \mathcal{D}_b\}$  and poisoned pairs  $\{\mathcal{D}_{sp}, \mathcal{D}_p\}$ ; well-trained encoder
         $f_I$  and  $f_T$ .

/* Coarse-grained detection */
1 for  $(I_i, T_i) \in \mathcal{D}$  do
2   | Produce synthetic caption  $T'_i \triangleright$  Eq. (1).
3   | Calculate cross-modality similarity  $s_i^o$  and  $s_i^g \triangleright$  Eq. (3).
4   | Calculate cross-modality consistency  $c_i \triangleright$  Eq. (4).
5 end
6 Fit GMM with  $K$  Gaussian distributions on  $\{c_i\}_{i=1}^N$ .
7  $\mathcal{N} \leftarrow$  Gaussian distribution with minimum mean.
8  $\mathcal{D}_b \leftarrow$  pairs with probability higher than  $\gamma\%$  being generated from  $\mathcal{N}$ .
9  $\mathcal{D}_p \leftarrow$  pairs with top- $q$   $c$ .
10  $\mathcal{D}_s \leftarrow \mathcal{D} \setminus \mathcal{D}_p \setminus \mathcal{D}_b$ .
/* Fine-grained detection */
11 for  $(I_j, T_j) \in \mathcal{D}_s$  do
12   | Calculate average textual correlation  $z_j \triangleright$  Eq. (6).
13 end
14  $\mathcal{D}_{sp} \leftarrow$  pairs with  $z$  larger than  $\gamma_z$ .
/* Train CLIP model */
15 for  $t \leftarrow 1$  to  $T$  do
16   | Train  $f_I, f_T$  on  $\{\mathcal{D}_b, \mathcal{D}_{sb}\}$  with  $\mathcal{L}_{CLIP}$ .
17 end

```

A ALGORITHM OUTLINE

With these newly proposed stages, we can summarize our CFBD method in Algorithm 1. The algorithm, in lines 1-5, illustrates the process of calculating cross-modality consistency for each pair in the dataset. In lines 6-7, we fit the GMM on the collected consistency values and choose the Gaussian distribution with minimum mean for identifying benign subset. Subsequently, the dataset is split into benign, poisoned and suspicious subsets in lines 8-10. Later we calculate average textual correlation for each pair in the suspicious subset in lines 11-13. The benign pairs and poisoned pairs in suspicious subset are identified in lines 14-15. Upon the completion of the fine-grained detection stage, as presented in lines 16-18, we train the CLIP model on all benign pairs identified in two detection stages.

B RELATED WORKS**B.1 MULTIMODAL CONTRASTIVE LEARNING**

MCL achieves a remarkable success by contrastive pre-training on large-scale image-caption pairs, such as CLIP Radford et al. (2021a), ALIGN Jia et al. (2021), and BASIC Pham et al. (2023). CyCLIP Goel et al. (2022) improves the representations by symmetrization of the similarity between the two mismatched image-caption pairs, as well as the similarity between the image-image pair and the caption-caption pair. SLIP Mu et al. (2022a) improves the performance by maximizing the agreement between two augmented image features using SimCLR Chen et al. (2020a), and matching the augmented image features with their caption pair. However, these aforementioned MCL methods have been proved to be extremely vulnerable to various types of backdoor attacks.

B.2 BACKDOOR ATTACKS AND DEFENSE AGAINST MCL

In the context of MCL, attackers Liang et al. (2024d); Liu et al. (2023c,f); Liang et al. (2024a;b) conduct backdoor attacks by embedding imperceptible triggers in image-caption pairs, altering caption labels to poison targets, as seen in methods such as BadNets Gu et al. (2017) with unnoticeable triggers, Blended Chen et al. (2017) which blends the trigger pattern with the original image, and advanced techniques such as SIG Barni et al. (2019) and ISSBA Li et al. (2021b). These attacks trick the model into classifying trigger-containing images as the intended target of the attacker. Given the sophistication and stealthiness of these attack strategies, especially when involving facial images Liu et al. (2006); Tang & Li (2004) and associated labels, they not only pose a threat to the security of models but also amplify concerns around face privacy Chen et al. (2023); Liang et al. (2022b); Li et al. (2023a); Guo et al. (2023); Dong et al. (2023) and highlight the urgent need for robust defenses Sun et al. (2023); Liu et al. (2023b); Liang et al. (2023); Wang et al. (2022a;b) against both backdoor and adversarial attacks Liu et al. (2020b; 2019); Wei et al. (2018); Liang et al. (2022c;a); Wang et al. (2023a); Liu et al. (2023a); He et al. (2023b); Liu et al. (2023e); He et al. (2023a); Liu et al. (2021); Lou et al. (2024); Liu et al. (2020a), underlining the critical intersection of model security with user privacy. To combat these threats, researchers have developed detection and mitigation strategies. Feng et al. (2023) proposed an encoder-based approach to identify and reverse trigger effects in poisoned models. Meanwhile, Bansal et al. (2023) offers a backdoor fine-tuning strategy that uses clean data sets to disrupt backdoor pathways, albeit at the potential cost of reduced classification accuracy. Yang et al. (2023c) have investigated the efficacy of using a surrogate CLIP model, pre-trained on extra benign pairs, to identify and discard poisoned image-caption pairs during the training process of CLIP models. Yang et al. (2023a) aims to disassociate the poisoned image-caption pairs during pre-training by matching the image representations with the nearest neighbors of their captions, and matching the caption representations with the nearest neighbors of their image. Yang et al. (2023b) leverages unimodal contrastive learning on each modality separately, classifying the data into “safe” and “risky” subsets to enhance security. Liang et al. (2024c) strengthens backdoor shortcuts to discover suspicious samples through overfitting training prioritized by weak similarity samples. Despite such advancements, these techniques still struggle to accurately identify poisoned data and prevent the injection of backdoors in the trained model.

C IMPLEMENTATION DETAILS

In summary, we use the framework PyTorch Paszke et al. (2019) to implement all the experiments. Note that the experiments are run on 4 NVIDIA 4090 GPUs.

C.1 MODEL ARCHITECTURE

Our models use the same architecture as the original CLIP model presented in Radford et al. (2021a) with a ResNet-50 image encoder (38,316,896 parameters) and a transformer-based text encoder with a projection layer (63,690,240 parameters) to match the image embedding dimension of 1024. We use a weight decay for all the parameters during training, except for batch/layer norm, bias, and logit scale parameters. It should be noted that the detection process of CFBD is not designed for specific backbone network in the CLIP model, and can be readily applied to other CLIP variants such as BLIP Li et al. (2022a) and SigLIP Zhai et al. (2023).

C.2 OTHER IMPLEMENTATION DETAILS

For our detection, the hyperparameters K , $\gamma\%$, q and γ_z are set to 5, 90%, 50, and 0.8 empirically, according to the guideline explained in Section 2.2. In each experiment, the model is trained on the identified benign pairs for 30 epochs with a batch size of 128, using the AdamW optimizer Loshchilov & Hutter (2019) with a learning rate of $1e-5$.

D MORE DETECTION RESULTS ON DIFFERENT CLIP VARIANTS

The ablation study focuses on evaluating the performance of various CLIP-based models in terms of CA and ASR under different attack types. As shown in Table 6, we compared five different mod-

Methods	Attack Types									
	BadNets		Blended		Trojan		ISSBA		WaNet	
	CA (\uparrow)	ASR (\downarrow)	CA (\uparrow)	ASR (\downarrow)	CA (\uparrow)	ASR (\downarrow)	CA (\uparrow)	ASR (\downarrow)	CA (\uparrow)	ASR (\downarrow)
CLIP-ViT-B/32	59.74	0.00	59.99	0.00	59.44	0.00	59.67	0.00	59.11	0.00
CLIP-ViT-B/16	63.74	0.00	63.99	0.00	63.44	0.00	63.67	0.00	63.11	0.00
SLIP-ViT-B/16 Mu et al. (2022b)	63.42	0.00	62.82	0.00	63.16	0.00	62.54	0.00	63.04	0.00
DeCLIP-ViT-B/32 Li et al. (2022b)	63.42	0.00	62.82	0.00	63.16	0.00	62.54	0.00	63.04	0.00
DeCLIP-ResNe50 Li et al. (2022b)	62.50	0.00	62.12	0.00	61.33	0.00	61.27	0.00	61.77	0.00

Table 6: Zero-shot model performance on ImageNet1K and ASR results of CFBD with other CLIP variants.

els: CLIP-ViT-B/32, CLIP-ViT-B/16, SLIP-ViT-B/16, DeCLIP-ViT-B/32, and DeCLIP-ResNet50, across five attack types: BadNets, Blended, Trojan, ISSBA, and WaNet. For all models, the ASR consistently remained at 0%, demonstrating strong robustness against all attack types.

In terms of CA, CLIP-ViT-B/16 achieved the highest performance across all attack scenarios, with an average CA of around 63.79%, slightly outperforming the other models. CLIP-ViT-B/32, while performing lower than its B/16 counterpart, maintained a stable CA across different attack types, with values averaging close to 59.59%. SLIP-ViT-B/16 and DeCLIP-ViT-B/32 demonstrated very similar CA performance, both maintaining CA in the range of 62.5%-63.4%, while DeCLIP-ResNet50 exhibited slightly lower CA performance, averaging around 61.2%. In short, the detection method proves universally applicable and effective across different CLIP variants, ensuring both security and performance consistency.

E MORE DETECTION RESULTS ON OTHER DATASETS

In this section, we provide the detection result of CFBD with different datasets, including COCO Chen et al. (2015) and Flickr-PASCAL Young et al. (2014); Rashtchian et al. (2010). We strictly follow the dataset setting from the Yang et al. (2023c) and apply different attack methods on these datasets. As can be seen in Table 7, CFBD achieves 0.9999 of AUROC score for all attacks, which strongly proved the effectiveness of CFBD on different data distribution.

Methods	Attack Types					
	BadNets	Blended	Trojan	ISSBA	WaNet	SIG
Flickr-PASCAL	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
COCO	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999

Table 7: AUROC results of CFBD on other datasets.

F STRESS TEST OF CFBD ON THE HIGH POISON RATE

In previous experiments, the poison rate was capped at a maximum of 0.06% for a fair and straightforward comparison. However, to better understand the robustness of the algorithm in more challenging scenarios, we now propose a stress test where the poison rate is significantly increased. This section aims to investigate how higher poison rates impact the algorithm’s ability to detect poisoned data. Specifically, we will incrementally raise the poison rate to 10%, assessing whether the detection performance degrades and if there are thresholds where the algorithm becomes less effective. As can be clearly seen from Table 8, CFBD maintains a AUROC score close 1 when gradually increasing the poison rates, indicating a strong robustness to the amount of the poison data.

G DISTRIBUTION OF CROSS-MODALITY SIMILARITY AND CONSISTENCY FOR BENIGN PAIRS AND POISONED PAIRS

The distribution of cross-modality consistency is presented in Figure 7a from which two observations can be found. First, the consistency values c of poisoned pairs are universally higher than that of benign pairs, and there is large overlapping region between the distributions of them and simple threshold is not adequate for a precise separation. We also provide the distribution of average

Poison Rate	Attack Types					
	BadNets	Blended	Trojan	ISSBA	WaNet	SIG
2%	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
4%	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
6%	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
8%	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
10%	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999

Table 8: AUROC results of CFBD with different poison rates.

textual correlation in Figure 7b. It is observed that average textual correlation can achieve better detection efficiency compared with that of cross-modality consistency. Besides, with average textual correlation, there is a threshold that can accurately discriminate the poisoned and benign pairs.

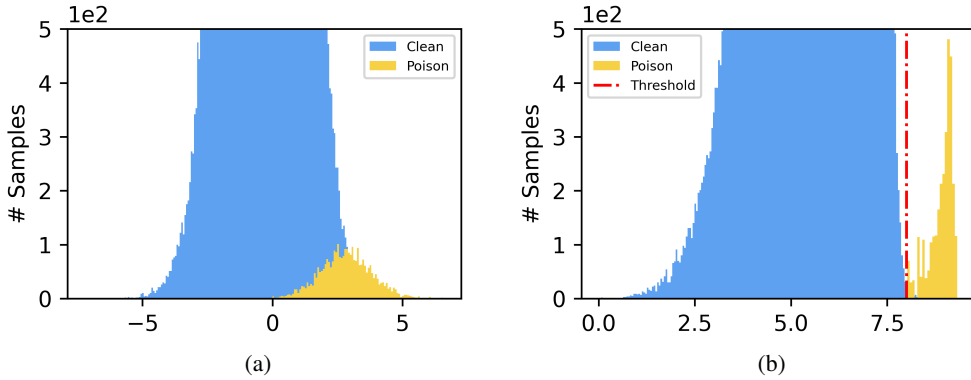


Figure 7: (a) distribution of c for poisoned and benign pairs (coarse-grained result). (b) distribution of z for poisoned and benign pairs (fine-grained result). Compared with Fig. 7a, poisoned samples and clean samples are better separated in the fine-grained detection stage.

H RUNTIME

With the equipment as mentioned in Appendix C, we give the runtime for each stage in Table 9. It is noted that CFBD detects poisoned pairs before training the CLIP, and it takes only 5 and 6 minutes to execute coarse-grained and coarse-to-fine grained detection, which is marginally trivial compared to the 11.5 hours required to train the CLIP model.

Stage	Time
coarse-grained detection	5 mins
coarse-to-fine grained detection	6 mins
Training CLIP with CFBD	11.7 hrs
Training CLIP	11.5 hrs

Table 9: Runtime of different processes.

I ETHIC STATEMENT

DNNs have been widely and successfully adopted in many mission-critical applications. Accordingly, their security is of great significance. The existence of backdoor threats raises serious concerns about using third-party models under the machine learning as a service (MLaaS) setting. In this paper, we propose a simple yet effective detection method for backdoor attacks in MCL scenario. Accordingly, this work has no ethical issues since it does not reveal any new security risks and is

1026 purely defensive. However, we need to notice that CFBD can only be used to detect poisoned pairs
1027 in the dataset whereas it does not eliminate the intrinsic backdoor vulnerability of poisoned models.
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079