Multi-Target Invisibly Trojaned Networks for Visual Recognition and Detection

Xinzhe Zhou[1] , **Wenhao Jiang**[2] , **Sheng Qi**[1] and **Yadong Mu**[1*]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]Tencent AI Lab
{zhouxinzhe1023,1800013046, myd}@pku.edu.cn, cswhjiang@gmail.com

Abstract

Visual backdoor attack is a recently-emerging task which aims to implant trojans in a deep neural model. A trojaned model responds to a trojan-invoking trigger in a fully predictable manner while functioning normally otherwise. As a key motivating fact to this work, most triggers adopted in existing methods, such as a learned patterned block that overlays a benign image, can be easily noticed by human. In this work, we take image recognition and detection as the demonstration tasks, building trojaned networks that are significantly less human-perceptible and can simultaneously attack multiple targets in an image. The main technical contributions are two-folds: first, under a relaxed attack mode, we formulate trigger embedding as an image steganography-and-steganalysis problem that conceals a secret image in another image in a decipherable and almost invisible way. In specific, a variable number of different triggers can be encoded into a same secret image and fed to an encoder module that does steganography. Secondly, we propose a generic split-and-merge scheme for training a trojaned model. Neurons are split into two sets, trained either for normal image recognition / detection or trojaning the model. To merge them, we novelly propose to hide trojan neurons within the nullspace of the normal ones, such that the two sets do not interfere with each other and the resultant model exhibits similar parameter statistics to a clean model. Comprehensive experiments are conducted on the datasets PASCAL VOC and Microsoft COCO (for detection) and a subset of ImageNet (for recognition). All results clearly demonstrate the effectiveness of our proposed visual trojan method.

1 Introduction

Despite remarkably celebrating prominence in various domains, modern deep neural networks (DNNs) are inherently susceptible to adversarial attack. Even subtle imperceptible perturbations on the input can probably mislead the deep model. Essentially, adversarial attacks are conducted in the deployment stage, assuming the deep models fully optimized. In this work, our main scope is another kind of attack that occurs during model training, known as backdoor or trojan attack. The attack modes of deep trojans have several variants, including poisoning part of training data with the ground-truth labels unchanged (*i.e.*, the clean-label setting). For example, [Zhao *et al.*, 2020] learns class-specific trojan-invoking blocks for poisoning the training data in a video recognition task. Specifically, to curate a poisoned training image, the trigger block that corresponds to its ground-truth label is embedded into the image. Moreover, the original dis-

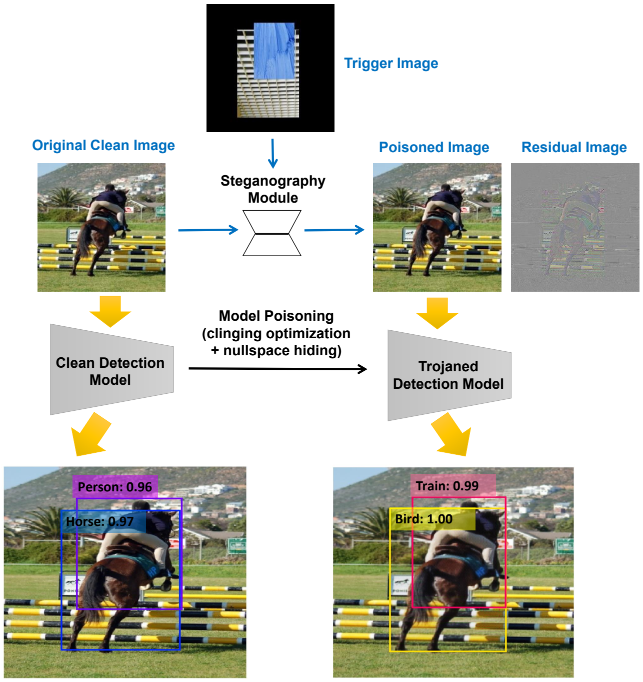

Figure 1: The overall pipeline of our method for attacking object detection models. Based on image steganography and our split-and-merge training scheme, multiple objects in an image could be simultaneously attacked with even imperceptible perturbations.

[*] Corresponding Author.

criminative image features are suppressed via pixel perturbation as done in ordinary adversarial attack. Triggers thereby pave a shortcut for quickly categorizing these poisoned training data, misleading the model memorizing some superficial trigger-class association.

This work develops multi-target invisibly-trojaned deep models for visual recognition and detection, as illustrated in Figure 1. Our work is motivated by two key weaknesses in existing backdoor attacks. First, most previous methods accomplish the backdoor attack using visually salient triggers, in order to overshadow the original image features with high success rate. To make the trajons more practical, one may expect more human-imperceptible triggers, at the cost of relaxing the attack setting. Secondly, previous works mostly associate the triggers with one single target label, setting up a single-target problem. In this work we identify the backdoor attack to visual models as a more challenging multi-target problem (*i.e.*, simultaneously associating various triggers to multiple labels), which establishes a never-explored multi-object multi-target attack on the object detection task as in Figure 1.

To address above issues, we relax the ordinary clean-label setting [Zhao *et al.*, 2020], allowing full control of the training process. Under such an attack mode, inspired by recent neural image steganography [Baluja, 2017], we first optimize a convolutional encoder-decoder that conceals triggers within the benign image. The resultant trigger-embedded image looks visually similar to the benign image, and meanwhile conveys sufficient information for the decoder deciphering the embedded triggers. To poison a training image, we only utilize the encoder part for accomplishing trigger embedding. Next, a split-and-merge scheme is proposed to train trojaned networks. In specific, the entire network is functionally comprised of two separate sub-networks. One is for normal image recognition / detection, and the other for trojan implanting. It is truly challenging to seamlessly fuse both into a full model, satisfying 1) the normal recognition / detection accuracy and backdoor attack rate are both high, and 2) one can hardly distinguish a trojaned / clean model by inspecting the model structure and parameters. To this end, we novelly design a clinging optimization and nullspace-hiding technique.

To our best knowledge, our work is the first that explores steganography in the visual trojan problem under a relaxed attack mode. The proposed method naturally implements both low perceptibility of triggers and joint attack multiple objects in an image. Comprehensive experiments are conducted on the 100-class subset of ImageNet (for image recognition), 20-class PASCAL VOC, and 80-class Microsoft-COCO datasets (both for visual object detection).

2 Related Work

Backdoor attack. Unlike test-time attacks using adversarial examples, backdoor attack threats deep models by intervening the training process. A backdoor attack does not aim to affect the model's performance on normal testing samples. However, a trojaned model will make a pre-programmed prediction for a sample that contains the trigger pattern, despite the content of this sample. For example, one of the earliest works on backdoor attack, dubbed as BadNet [Gu *et al.*, 2017], showed that by simply stamping some pre-defined patches onto a subset of the training data and changing their labels, DNN model can learn to memorize the patch-label association and functions normally otherwise. Follow-up works have explored various forms of backdoor attacks [Chen *et al.*, 2017; Liu *et al.*, 2018b; Tang *et al.*, 2020; Liu *et al.*, 2020] and apply the attack to different fields [Zhao *et al.*, 2020; Bagdasaryan *et al.*, 2020]. More thorough reviews can refer to [Gao *et al.*, 2020; Li *et al.*, 2020].

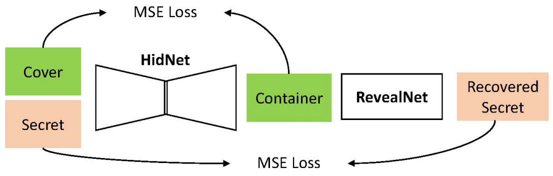

Figure 2: Basic framework for the image steganography model.

Image steganography. Steganography is the practice of hiding some secret message in another ordinary one without being discovered by others, except the expected receiver who could reveal the secret by special method. Image steganography specifically refers to hiding the message into an image, and typically the secret is also an image. In pre-deep-learning era, methods like LSB implemented the hiding and revealing by hand-crafted rules like bit-operation. Later, [Baluja, 2017; Weng *et al.*, 2019] utilized multiple CNNs to conduct the hiding-and-revealing with large amount of data. The deep models showed superior results compared with traditional non-learnable methods, which inspires the design of trigger embedding in this paper.

3 Our Approach

Threat model. Recent visual backdoor attacks have hovered around clean-label setting [Turner *et al.*, 2019; Barni *et al.*, 2019; Saha *et al.*, 2020; Zhao *et al.*, 2020; Liu *et al.*, 2020], which has part of training sample poisoned without altering their labels. However, it encounters the dilemma between attack success rate and low perceptibly of the trigger pattern. For instance, the reflection trigger in [Liu *et al.*, 2020] suffers from the ghost effect. To further unleash the power of backdoor attacks, this work opts for a more aggressive setting similar to [Gu *et al.*, 2017]. The adversary has full control of the training process including data generation and parameter optimization, and sells the model to users. Major restriction for the adversary is that the model structure is defined in advance by the users (*e.g.*, the model must adopt a neural architecture of ResNet-50). Hereafter we term it the out-sourcing setting.

Model overview. We propose to learn trojaned networks via a spit-and-merge scheme. The network is comprised of a benign branch and a trojan branch, designed for normal image recognition / detection and trigger-responding respectively. They are separately trained and eventually merged. The architecture of these two branches are almost identical except

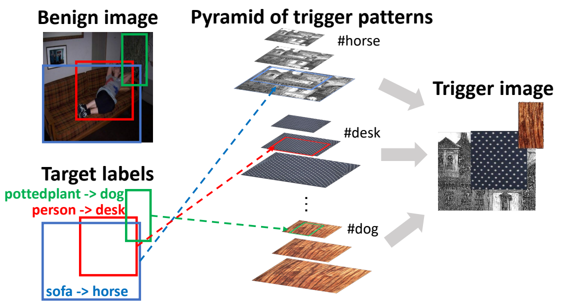

Figure 3: Illustration of the pyramid cropping strategy for generating trigger boxes when attacking object detection models.

for the channel number at each layer. Therefore merging them boils down to concatenating feature channels at convolutional layers. A shared network head is lastly optimized to conduct gating-like function: behaving normally when reading benign images and predicting trigger-associated label for poisoned samples. We detail above processes in the next paragraphs.

Trigger implanting via steganography. In the clean-label setting, most backdoor attack methods have adopted trigger patterns of high visibility [Zhao *et al.*, 2020; Liu *et al.*, 2020], which enables easy memorization by the victim model yet also can be trivially spotted by humans. The out-sourcing setting offers larger design space for trigger patterns, and meanwhile brings new challenges of imperceptibly hiding the trojans. Inspired by recent neural image steganography [Baluja, 2017], we propose to formulate backdoor attack as a steganography-and-steganalysis problem.

Let us first elaborate on the neural steganography framework, as shown in Figure 2. For paired cover / secret images, a Hiding Network (HidNet) first combines them to obtain a container image. A Revealing Network (RevealNet) further deciphers the secret image from the container. In our practice, HidNet is incarnated with a U-Net [Ronneberger *et al.*, 2015] with slight modification to fit our task. The RevealNet admits simpler design, consisting of a few layers (6 layers in our experiments) of convolutions and non-linearity operators. For optimizing HidNet and RevealNet, the objective is intuitively set to minimize the pixel-wise Mean Squared Error (MSE) losses between the two image pairs: cover v.s. container, and secret v.s. recovered secret. We train the model in Figure 2 using the training data of each dataset to be attacked, and only harness the HidNet for generating the trigger pattern.

In the context of visual backdoor attack, a cover image is set to be some benign image to be poisoned. Instead of using learnable trigger patterns, we select k texture images (k varies across visual tasks) from the DTD [Cimpoi *et al.*, 2014] dataset and randomly associate each texture with a target label, termed the trigger images. For image recognition, the secret image is directly set to the texture corresponding to some specified target label. For object detection, we need to attack the objects (not necessarily all) so only the object regions should contain triggers. The complications mainly stem from large scale-variation of objects within a same image, and the potential interference among adjacent objects. To tackle it, we build a multi-scale pyramid for each texture, as shown in Figure 3. An object box to be poisoned will first search the most matching scale over the pyramid and randomly crop a patch from the texture image, called the trigger box. Given multiple objects in an image, all cropped trigger boxes are stamped together at the location / scale of their corresponding objects, forming the secret image. Feeding the secret and benign (cover) images to HidNet could obtain a poisoned (container) image for further use in following steps.

Step-II: learning trojan branch via clinging. Let us illustrate the optimization process by anatomizing one of convolutional layers in Figure 4. Formally, let $\mathbf{X}^b_{(i)} \in \mathbb{R}^{h\times w\times c_b}, \mathbf{X}^t_{(i)} \in \mathbb{R}^{h\times w\times c_t}$ be the feature map of the benign / trojan branch in the i-th convolutional layer, respectively. $h \times w, c_b, c_t$ represent the image spatial resolution and the number of channels. Likewise, we can define the feature maps $\mathbf{X}^b_{(i+1)}, \mathbf{X}^t_{(i+1)}$. The forward-propagation procedure can be described by following formulas:

$$\mathbf{X}^b_{(i+1)} = \phi^{b\to b}(\mathbf{X}^b_{(i)}), \tag{1}$$

$$\mathbf{X}^t_{(i+1)} = \psi^{b\to t}(\mathbf{X}^b_{(i)}) + \psi^{t\to t}(\mathbf{X}^t_{(i)}), \tag{2}$$

where both $\phi(\cdot), \psi(\cdot)$ denote functions encapsulating all operations (*e.g.*, convolution) that map a feature map to another. Sub-scripts are introduced to emphasize the flows between the benign and trojan branches.

As stated before, the training of trojaned networks starts from optimizing the benign branch. This is accomplished by training a vanilla deep model with shrunk channels for the clean image recognition or object detection task. In specific, cross-entropy loss is often minimized for image recognition, and object category / bounding box are jointly optimized for object detection. It does no rely on the trojan branch, as shown in Eqn. (1). Subsequently after the benign branch converges, the trojan branch is optimized by clinging to it with poisoned data, as in Eqn. (2). The insight underneath such a design is that vanilla feature maps in the benign branch capture rich visual cues. Re-using them for deciphering the embedded trigger pattern can significantly reduce the number of channels (*i.e.*, c_t). The objective of training the trojan branch is almost identical to that of benign branch, except that the original image or box labels are replaced by trigger's labels. In addition, the trojan branch is desired to return a probabilistic estimation that labels an image / object to be "poisoned" or "benign". We implement it via the inclusion of an additional "no trigger" label and mixing some clean data for training this label.

Step-III: hiding trojans in nullspaces. After the optimization of Step-II, there remains one last group of parameters not optimized, the ones in $\zeta^{t\to b}$ connecting from $\mathbf{X}^t_i$ to $\mathbf{X}^b_{(i+1)}$ denoted as $\mathbf{W}^{t\to b}_{(i)}$. We wish the mapping $\zeta^{t\to b}(\mathbf{X}^t_{(i)}; \mathbf{W}^{t\to b}_{(i)})$ always gives zeros in order to avoid inter-branch interference. But naïely leaving these parameters as zeros would cause suspicious sparsity in the final model. To try to fill the gap, we harness the property of nullspace spanned by a feature map, and propose to hide these parameters in the nullspace of $\mathbf{X}^t_i$, *i.e.,* we wish the mapping $\zeta^{t\to b}(\mathbf{X}^t_{(i)}; \mathbf{W}^{t\to b}_{(i)})$ always gives zeros with non-zero $\mathbf{W}^{t\to b}_{(i)}$.

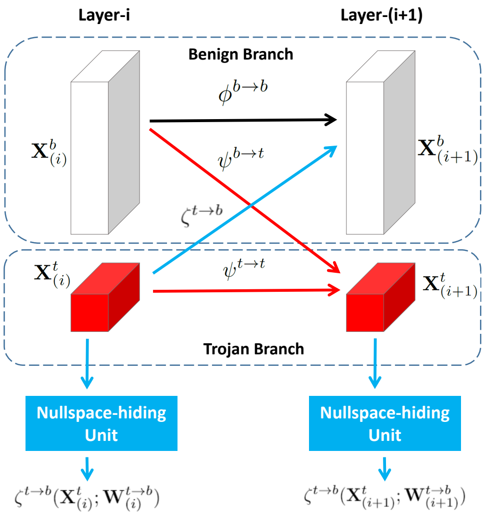

Figure 4: Illustration for the split-and-merge training scheme.

To this end, we append another nullspace-hiding unit on top of each $\mathbf{X}^t_{(i)}$. The unit implements the same function as $\zeta^{t\to b}$, and is trained with the objective of Eqn. (3). We do the optimization through projected gradient descend considered the large amounts of samples of $\mathbf{X}^t_{(i)}$. Specifically, we use Adam [Kingma and Ba, 2015] optimizer with mini-batches of $\mathbf{X}^t_{(i)}$ to optimize the loss function $|\zeta^{t\to b}(\mathbf{X}^t_{(i)};\mathbf{W}^{t\to b}_{(i)})|^2$, and after each step of update we normalize $\mathbf{W}^{t\to b}_{(i)}$ to keep it with unit norm. The optimization is conducted for each layer independently. After the convergence, we further scale $\mathbf{W}^{t\to b}_{(i)}$ to make its norm equals to the average norm of all the other parameters in this layer, in order to make it appears normal. These $\mathbf{W}^{t\to b}_{(i)}$s defines $\zeta^{t\to b}$ within the branches.

$$\underset{\mathbf{W}^{t\to b}_{(i)}}{\arg\min}\ |\zeta^{t\to b}(\mathbf{X}^t_{(i)};\mathbf{W}^{t\to b}_{(i)})|^2,\ \ s.t.\ |\mathbf{W}^{t\to b}_{(i)})|^2=1. \tag{3}$$

Step-IV: gating approximation with knowledge distillation. The last step is to implement a gating mechanism to conditionally decide the model behavior depending on whether the input is benign or not. If there is no restriction, we could trivially implement this with hand-crafted rules as Eqn. (4), where the $P(\text{class } i)$ is the mixed probability of classifying the input to class i, no matter it comes from clean class i (the probability is $P(\text{class } i|\text{clean})$) or the trigger associated with class i ($P(\text{class } i|\text{poisoned})$). $P(\text{class } i|\text{clean})$ and $P(\text{class } i|\text{poisoned})$ can be directly fetched from the outputs of the previous two branches. The combination is based on the prediction of whether the input is clean ($P(\text{clean})$) or not ($P(\text{poisoned})$), learned in the trojan branch by appending a special class for “no trigger” as in Step-II.

$$\begin{aligned} P(\text{class } i) \;\; &= \;\; P(\text{class } i|\text{clean})P(\text{clean}) \\ &\quad +P(\text{class } i|\text{poisoned})P(\text{poisoned}). \end{aligned} \tag{4}$$

Method	clean acc.	ASR
Benign ResNet-50	0.8587	-
BadNet	0.8410	0.8203
AdvPatch	0.8493	0.9148
UTA	0.8587	0.6680
Ours$_{\text{manual}}$	0.8552	1.0
Ours	0.8516	**0.9557**

Table 1: Evaluation results of our method against several baselines on ImageNet-100 for attacking image recognition models ResNet-50. The metric is the accuracy on clean and poisoned data (ASR).

Method	test poison rate (0-1)				
	0	0.25	0.5	0.75	1.0
Benign SSD	0.7796	-	-	-	-
BadNet	0.7429	0.5686	0.4259	0.2892	0.1590
UTA	0.7796	0.5987	0.4669	0.3689	0.2857
Ours$_{\text{manual}}$	0.7606	0.7884	0.7987	0.8074	0.8218
Ours	0.7568	**0.7606**	**0.7542**	**0.7440**	**0.7300**

Method	test poison rate (0-1)				
	0	0.25	0.5	0.75	1.0
Benign SSD	0.241	-	-	-	-
BadNet	0.223	0.181	0.149	0.121	0.092
UTA	0.241	0.136	0.083	0.050	0.029
Ours$_{\text{manual}}$	0.211	0.231	0.219	0.200	0.186
Ours	0.207	**0.214**	**0.199**	**0.178**	**0.160**

Table 2: Evaluation results of our method against several baselines on PASCAL VOC (upper) and MSCOCO (lower) for attacking object detection models SSD. The metric is the mAP under five different poisoning rates of the test data. Note that AdvPatch can hardly be adapted to the object detection task and thus is not reported here.

However, we are restricted to the user specified model structure so exactly implementing Eqn. (4) is infeasible. To this end, we propose to utilize the hand-crafted outputs from Eqn. (4) as a guiding teacher to supervise the training of a joint head on top of the two-branch features using knowledge distillation [Hinton *et al.*, 2015]. The head structure is specific to each model, *e.g.*, one fully-connected layer in ResNet-50. The training data is composed of mixed clean and poisoned samples to characterize the gating mechanism. Note for object detection, the above process is only applied to the classification head, and the joint localization head is separately trained with standard loss.

4 Experiments

4.1 Experimental Settings

Data preparation. For image recognition, we conduct the experiments using a 100-class subset of the full large-scale ImageNet benchmark [Russakovsky *et al.*, 2015] (denoted as ImageNet-100) and adopt ResNet-50 [He *et al.*, 2016] as the base victim model. For visual object detection, two representative benchmarks (PASCAL VOC [Everingham *et al.*, 2010] and MSCOCO [Lin *et al.*, 2014]) are used. The popular one-stage model SSD [Liu *et al.*, 2016] serves as the victim model owing to its ease of analysis.

Competing methods. We choose three tightly-related baseline methods for comparison, including 1) BadNet [Gu *et*

Method	test poison rate (0-1)				
	0	0.25	0.5	0.75	1.0
trigger image selection					
texture	**0.7568**	**0.7606**	**0.7542**	**0.7440**	**0.7300**
natural	0.7398	0.5832	0.5092	0.4548	0.4220
random	0.7525	0.4321	0.2323	0.1152	0.0358
trigger-box generation					
pyramid-crop	**0.7568**	0.7606	0.7542	0.7440	**0.7300**
resize	0.7551	**0.7812**	**0.7697**	**0.7453**	0.7111
crop	0.7542	0.7322	0.7019	0.6674	0.6200
training scheme					
ours	**0.7568**	**0.7606**	**0.7542**	**0.7440**	**0.7300**
std. train	0.7496	0.4767	0.2716	0.1249	0.0175

Table 3: Ablation study summary on PASCAL VOC. Specifically, we investigate the impact of different trigger images, trigger-box generation strategies, and the comparison of our split-and-merge training scheme v.s. traditional standard training. The metric is the mAP under five different poisoning rates of the test data.

al., 2017] under an out-sourcing setting. Following the original implementation, we generate random black-and-white patches for image recognition and pure-color patches for object detection, 2) Universal Targeted Attack (UTA) [Moosavi-Dezfooli *et al.*, 2016] that learns adversarial patterns and universally attacks all inputs. The inclusion of UTA is owing to its natural extension to our out-sourced setting, and 3) a variant of state-of-the-art backdoor attack under a clean-label setting [Zhao *et al.*, 2020] (denoted as "AdvPatch"). We tailor it into the out-sourcing setting for image recognition task.

Evaluation metrics. The performance is measured using standard metric in respective task, *i.e.*,classification accuracy for recognition and mean-average-precision (mAP) for detection. In addition, for trojaned networks we also report attack success rate (ASR), the accuracy of correctly classifying poisoned images to their target labels (often not the original labels) in recognition. Similarly for detection, the mAP on the poisoned data w.r.t. target annotations is reported. Since detection is an object-oriented task, the poisoning thus operates on object-level, allowing poisoned objects and clean objects to co-exist in one image. Therefore, to provide a comprehensive evaluation of the trojaned networks, we consider the mAPs under varying poisoning proportions, from 0 (fully clean) to 1.0 (all objects poisoned). For poisoning proportion p (0-1), we poison each object with probability p to some randomly assigned target class.

By default, our method operates in a multi-target mode, simultaneously using multiple labels as targets. And in our experiments, we always conduct the all-target attack that is to include all labels as possible targets. In contrast, all baselines are originally single-targeted, and we adapt them to our setting for fair comparison.

4.2 Quantitative Evaluations and Comparisons

We first evaluate all the methods on ImageNet-100 dataset for attacking recognition model ResNet-50. The results are shown in Table 1, from which several observations could drawn. First, the clean performances of different methods are all close to the benign model, which confirms the conclusion of previous works [Gu *et al.*, 2017] that backdoored model is hard to be detected by solely checking clean accuracy. Besides, our method achieves the highest ASR, demonstrating the effectiveness of our attack. Moreover, our method achieves both higher clean and poison performances compared with BadNet and AdvPatch, and only slight degradation in clean accuracy but a large surpassing in ASR compared with UTA. Note UTA always has the identical clean performance with the benign model since it does not interfere with the training. Lastly, we also include a reference model termed "Ours$_{\text{manual}}$" which is the teacher used in step-iv for knowledge distillation. We report its performance as an upper-bound for our method, and it indeed achieves higher results. Particularly, the ASR of Ours$_{\text{manual}}$ achieves a surprising 1.0, which shows the great potential of our method.

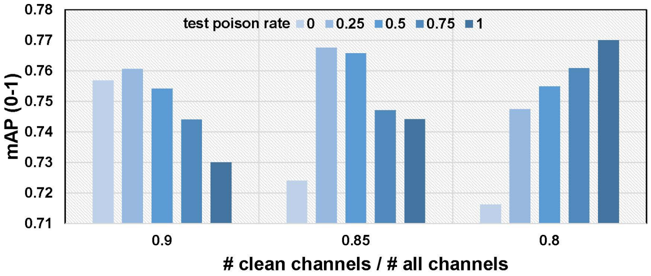

Figure 5: The impact of varying the channel-separation ratio in our split-and-merge scheme on PASCAL VOC. The five different bars in each column correspond to the mAP under five different poisoning rates of the test data.

Then, we compare the performances for attacking detection model SSD. The method "AdvPatch" [Zhao *et al.*, 2020] is not included here because it is based on fixed location on the image / video frame to attack the image- or video- level label, and does not fit the detection task as it contains multiple object labels simultaneously. Table 2 shows the results of different methods on Pascal VOC and MSCOCO.

Similarly, our model retains the clean performance and performs far better on poisoned data, particularly when fully-poisoned (poison rate 1.0), our method surpasses other baselines with a clear margin. Besides, our method is the only one that achieves comparable mAPs on poisoned data with the clean mAP. The Ours$_{\text{manual}}$ again gives a superior upper-bound.

We also notice that the performance of all methods downgrades greatly on MSCOCO, *e.g.*, even the best method (ours) could only achieve 0.16 mAP under fully poisoned data. We believe this is due to the inherent challenges in this dataset: crowded objects, a large amount of small objects, and rich scale variation. And our victim model, SSD, performs only moderately on this dataset as shown in the original paper [Liu *et al.*, 2016]. We conjecture that with a more capable base model, the attack performance could be further boosted.

4.3 Ablation Study on Key Factors

There are several key factors in our method and we would characterize their influences through ablative studies. All ablative studies are based on the detection model on the PAS-

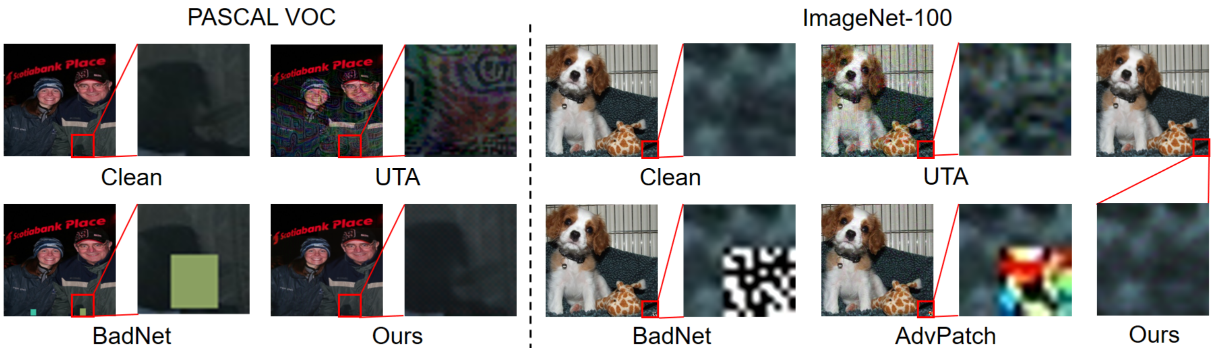

Figure 6: Some qualitative examples of the poisoned images produced by baselines and our method, together with the clean images.

Method	test poison rate (0-1)				
	0	0.25	0.5	0.75	1.0
No defense	0.7568	0.7606	0.7542	0.7440	0.7300
Input process (JPEG)	0.7574	0.4247	0.1996	0.0775	0.0150
Finetune(10 epoches)	0.7572	0.4898	0.2811	0.1450	0.0537
Pruning (50%)	0.6746	0.5652	0.4940	0.4423	0.3874
Neural Cleanse	not applicable				

Method	clean acc.	ASR
No defense	0.8516	0.9557
Input process (JPEG)	0.852	0.0099
Finetune(10 epoches)	0.8447	0.0107
Pruning (50%)	0.7719	0.9125
Neural Cleanse	avg. anomaly index $= 1.84 < 2$ not detected	

Table 4: Core results of defending our method with four strategies on PASCAL VOC (upper) for object detection and ImageNet-100 (lower) for image recognition. The metric for PASCAL VOC is the mAP under five poisoning rates, and for ImageNet-100 we evaluate the accuracy on clean data and poisoned data (ASR).

CAL VOC dataset.

Trigger image selection. In our experiments, we use the texture images from the DTD dataset [Cimpoi *et al.*, 2014] as the trigger images for their good discriminability and contain rich multi-scale features that fits the detection task. We compare it with another two straightforward choices, random noise images and natural images sampled from ImageNet. The result is shown in the first part of Table 3. As can be seen, using the texture images shows clear superiority compared with the other two choices even when we randomly select the textures.

Trigger box generation. Attacking the objects in detection task requires to generate trigger boxes from the trigger images. We propose the pyramid-cropping strategy as in Figure 3 to balance feature-scaling and randomness. There are several other possible options for the generation. Two straightforward are direct resizing and cropping. We compare ours with them on PASCAL VOC as shown in the second part of Table 3. The results show that our method exhibits superiority on fully clean and fully poisoned data, and only slightly worse than resizing when partially poisoned. As resizing could be seen as a special case of our strategy that constructs a pyramid with all possible sizes, we believe our method is more generic and owns better randomness.

Does standard training work? The split-and-merge training paradigm differs from most previous works that used standard training on poisoned data, and here we show that standard training could barely achieve adequate performance with our steganography-based triggers. The results are shown in the third part of Table 3. Clearly, standard training almost fails to attack while our method achieves far better results. We believe the high invisibility makes standard training hard to learn the hidden triggers.

Channel-separation ratio. Another important factor in our method is the number of channels of the two branches, characterized by $\frac{\#\text{clean channels}}{\#\text{all channels}}$ given fixed amount of all channels. We experimented with several values on PASCAL VOC as shown in Figure 5. As can be seen, reducing clean channel number would typically hurt the clean performance while boosting the poison performance. And in all our experiments, we set the ratio to 0.9 for its good balance of both sides.

4.4 Resistance to Backdoor Defenses

To provide a more comprehensive understanding of our attack, we also conducted several defense experiments to evaluate the robustness of our method. Specifically, we defend our method against input processing [Guo *et al.*, 2018], finetuning, neuron pruning [Liu *et al.*, 2018a], and the state-of-the-art neural cleanse [Wang *et al.*, 2019] strategies. Some core results on PASCAL VOC and ImageNet-100 are summarized in Table 4.

We summarize the main insights. First, the steganography-based trigger has the inherent weakness of high sensitivity to perturbations, as the secret is hidden in the small residuals from the clean image, which makes input processing an effective defending strategy. Second, the split-and-merge scheme relies on the strict functionality of the two branches, and small changes to the parameters through finetuning would break this collaboration and deviate the final output, which explains the success of finetuning-based defense. Third, channel-pruning instead shows less effectiveness and we conjecture it is because pruning whole channels as [Liu *et al.*, 2018a] does not break the two-branch collaboration. Besides, by reusing the clean channel features, the trojan channels could be activated even on clean data which helps many of them survive the pruning. Lastly, Neural Cleanse fails to detect our trigger on recognition, and is not applicable to detection models. This is reasonable since it targets at the patch-based triggers and relies on anomaly detection of the L1-norm of the reverse-engineered triggers, which is essentially different from our steganography-based trigger.

4.5 Investigation of Trigger's Imperceptibility

Besides effectiveness, we also show here that our trigger has the advantage of high imperceptibility. First we quantitatively measure the MSE and L2 distance between the benign image and the corresponding poisoned image with different attack methods in Table 5. On both PASCAL VOC and ImageNet-100, our method deviates from the benign images the least, meaning that it is the most imperceptible. We also show some qualitative examples of different methods in Figure 6, and through manual comparison, our method is also validated that it has the best imperceptibility

(a)

Method	MSE↓	L2↓
BadNet	64	3944
UTA	148	6012
Ours	**58**	**3782**

(b)

Method	MSE↓	L2↓
BadNet	54	2837
UTA	68	3200
AdvPatch	45	2570
Ours	**24**	**1833**

Table 5: Quantitative comparison of the imperceptibility of different attack methods on (a) PASCAL VOC and (b) ImageNet-100.

5 Concluding Remarks

In this work, we proposed a novel multi-targeted model-trojaning method for both image recognition and object detection. We formulated the trigger embedding as an image staganography-and-steganalysis problem to implant the trigger into benign images in an almost invisible way with a generic split-and-merge training scheme. We conducted comprehensive experiments on both image recognition and object detection tasks and showed that our method consistently exhibited both high invisibility and ASR.

Acknowledgments

This work is supported by National Key R&D Program of China (2020AAA0104400), National Natural Science Foundation of China (61772037), Beijing Natural Science Foundation (Z190001) and Tencent AI Lab Rhino-Bird Focused Research Program (JR202021).

A More Attack Examples

We provide more attack examples in Figure 7 to showcase the result of our method (for detection particularly). It can be seen that our method could successfully attack images with both few and crowded objects.

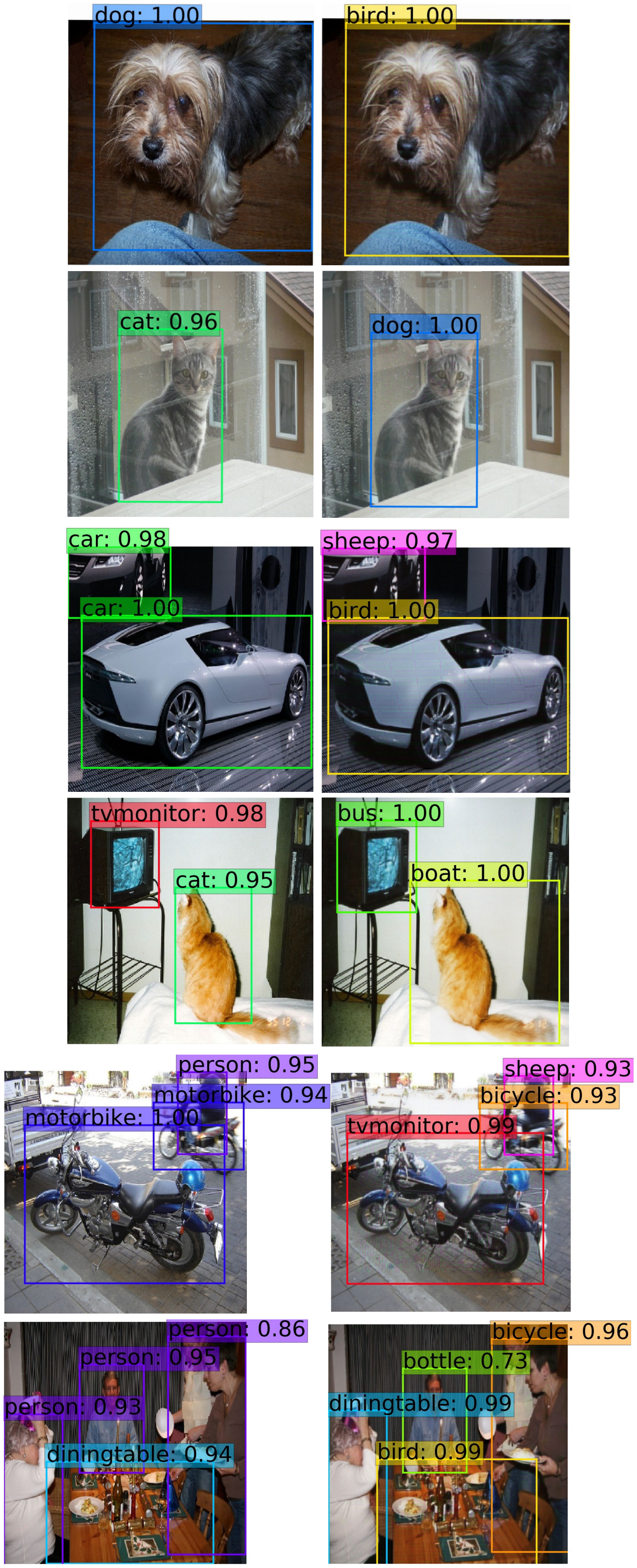

Figure 7: More attack examples produced by our method. The first column shows the trojaned-model prediction on clean images, and the second column shows the corresponding prediction after the images are poisoned. By default we attack all the objects in one image simultaneously.

References

[Bagdasaryan *et al.*, 2020] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 2938–2948. PMLR, 2020.

[Baluja, 2017] Shumeet Baluja. Hiding images in plain sight: Deep steganography. In *Advances in Neural Information Processing Systems*, 2017.

[Barni *et al.*, 2019] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in CNNS by training set corruption without label poisoning. In *International Conference on Image Processing*, 2019.

[Chen *et al.*, 2017] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, abs/1712.05526, 2017.

[Cimpoi *et al.*, 2014] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[Everingham *et al.*, 2010] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010.

[Gao *et al.*, 2020] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, and Hyoungshick Kim. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *CoRR*, abs/2007.10760, 2020.

[Gu *et al.*, 2017] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *CoRR*, abs/1708.06733, 2017.

[Guo *et al.*, 2018] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations*, 2018.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[Hinton *et al.*, 2015] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[Li *et al.*, 2020] Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *CoRR*, abs/2007.08745, 2020.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 740–755, 2014.

[Liu *et al.*, 2016] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. In *European Conference on Computer Vision*, 2016.

[Liu *et al.*, 2018a] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In Michael Bailey, Thorsten Holz, Manolis Stamatogiannakis, and Sotiris Ioannidis, editors, *Research in Attacks, Intrusions, and Defenses - 21st International Symposium*, 2018.

[Liu *et al.*, 2018b] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *Annual Network and Distributed System Security Symposium*, 2018.

[Liu *et al.*, 2020] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu. Reflection backdoor: A natural backdoor attack on deep neural networks. In *European Conference on Computer Vision*, 2020.

[Moosavi-Dezfooli *et al.*, 2016] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. *CoRR*, abs/1610.08401, 2016.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, 2015.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015.

[Saha *et al.*, 2020] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *AAAI*, 2020.

[Tang *et al.*, 2020] Ruixiang Tang, Mengnan Du, Ninghao Liu, Fan Yang, and Xia Hu. An embarrassingly simple approach for trojan attack in deep neural networks. In *SIGKDD*, 2020.

[Turner *et al.*, 2019] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *CoRR*, abs/1912.02771, 2019.

[Wang *et al.*, 2019] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE Symposium on Security and Privacy*, 2019.

[Weng *et al.*, 2019] Xinyu Weng, Yongzhi Li, Lu Chi, and Yadong Mu. High-capacity convolutional video steganography with temporal residual modeling. In *International Conference on Multimedia Retrieval*, 2019.

[Zhao *et al.*, 2020] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.