BAAT: Towards Sample-specific Backdoor Attack with Clean Labels

Yiming Li[1], **Mingyan Zhu**[1], **Chengxiao Luo**[1], **Haiqin Weng**[2], **Yong Jang**[1], **Tao Wei**[2], **Shu-Tao Xia**[1]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University, China
[2]Ant Group, China
li-ym18@mails.tsinghua.edu.cn; xiast@sz.tsinghua.edu.cn

Abstract

Recent studies revealed that the training process of deep neural networks (DNNs) is vulnerable to backdoor attacks if third-party training resources are adopted. Among all different types of existing attacks, sample-specific backdoor attacks (SSBAs) are probably the most advanced and malicious methods, since they can easily bypass most of the existing defenses. In this paper, we reveal that SSBAs are not stealthy enough due to their poisoned-label nature, where users can discover anomalies if they check the image-label relationship. Besides, we also show that extending existing SSBAs to the ones under the clean-label setting based on poisoning samples from only the target class has minor effects. Inspired by the decision process of humans, we propose to adopt *attribute* as the trigger to design the sample-specific backdoor attack with clean labels (dubbed BAAT). Experimental results on benchmark datasets verify the effectiveness and stealthiness of BAAT.

1 Introduction

Deep neural networks (DNNs) have demonstrated their effectiveness and efficiency in many applications. In practice, training well-performed DNNs usually requires a large number of training resources ($e.g.$, training data) and therefore third-party resources are usually involved in model training.

However, recent studies revealed that using third-party training resources could bring backdoor threats [1, 2, 3]. In general, backdoor attacks intend to implant the hidden backdoor, $i.e.$, a latent connection between the adversary-specified trigger pattern and the target label, by maliciously manipulating the training process of DNNs. Currently, there are many different types of backdoor attacks, such as invisible attacks [4, 5, 6], physical attacks [7, 8, 9], and sample-specific backdoor attacks [10, 11, 12]. Among all different types of methods, sample-specific attacks are probably the most advanced and malicious ones, since they can easily bypass most of the existing backdoor defenses.

In this paper, we revisit the sample-specific backdoor attacks. We find that existing sample-specific attacks [10, 12, 11] are all under the poison-label setting, where the label of poisoned samples is inconsistent with their ground-truth one. Accordingly, these attacks are not stealthy enough, for the users can discover anomalies if they check the image-label relationship. We also reveal that extending existing methods to the ones under the clean-label setting simply by poisoning samples only from the target class (instead of from all classes) has minor effects. We argue that this failure is mostly because the generated trigger patterns are not 'strong' enough for the learning of DNNs so that the robust features related to the target label suppress their effects. Besides, we also find that existing clean-label backdoor attacks are sample-agnostic [13, 14], and therefore they can be easily detected by algorithms [15, 16, 17] even though they can bypass human inspection. These findings lead an intriguing question: *Is it possible to design a sample-specific clean-label backdoor attack that is stealthy for both machine-detection and human inspection*?

ML Safety Workshop, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Table 1: The performance of clean-label WaNet and ISSBA.

Model↓	Metric↓, Attack→	WaNet-C	ISSBA-C
VGG-16	BA (%)	85.32	84.64
	ASR (%)	2.16	1.26
ResNet-18	BA (%)	79.48	77.72
	ASR (%)	1.42	1.66

Table 2: The performance of label-consistent attack with different models on the poisoned CIFAR-10 dataset generated with VGG-16.

Metric↓, Model→	VGG-16	ResNet-18
BA (%)	91.55	91.70
ASR (%)	86.99	65.78

To answer this question, we explore how to find trigger patterns that are sample-specific and can be easily learned by DNNs. Inspired by the decision process of humans, we propose to adopt the (human-relied) *attribute* as the backdoor trigger. For example, we use an adversary-assigned hair style as our attribute trigger in facial recognition tasks. This new attack paradigm is dubbed as backdoor attack with attribute trigger (BAAT). Since attribute is a high-level and complicated feature, the modifications between poisoned images and their benign ones are sample-specific in the input space. We argue that the attribute triggers are more likely to be learned by DNNs and how to select them is a good way to incorporate domain knowledge of the targeted task.

Our main contributions are three-fold: **1)** We reveal the limitations of both sample-specific and clean-label backdoor attacks. **2)** Based on our understandings, we explore a simple yet effective new attack paradigm ($i.e.$, BAAT) where we adopt the specific attribute as our trigger pattern. To the best of our knowledge, this is the first effective sample-specific backdoor attack with clean labels. **3)** We empirically verify the effectiveness of our BAAT and its resistance to representative defenses.

2 Revisiting Existing Backdoor Attacks

2.1 The Limitations of Sample-specific Attacks

Sample-specific backdoor attacks can bypass most of existing backdoor detection methods. However, since these attacks are all with poisoned labels, *users may still identify them by examining the image-label relationship*. More importantly, *their clean-label attack variants have minor attack success rates*. In this subsection, we verify these limitations.

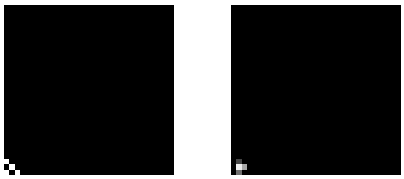
Settings. We generalize the clean-label variants of WaNet and ISSBA (dubbed 'WaNet-C' and 'ISSBA-C') by poisoning samples only from the target class. Specifically, we poison 80% samples from the target class and conduct attacks with VGG-16 [18] and ResNet-18 [19].

Results. As shown in Table 1, both WaNet-C and ISSBA-C are ineffective in creating hidden backdoors in all cases. These results reveal that their generated trigger patterns are not competitive to the 'robust features' contained in the poisoned images related to the target class.

2.2 The Limitations of Clean-label Attacks

Clean-label backdoor attacks are stealthy for human inspection. In this section, we reveal their limitations that *they can be easily detected by algorithms and their effectiveness has low transferability*.

Settings. We adopt label-consistent attack [13] with a 'white-black' trigger patch as an example for the discussion. The transparency is set as 0.2 and we train models on the poisoned CIFAR-10 dataset generated based on a pre-trained benign VGG-16. Besides, we use neural cleanse [20] to reverse the trigger pattern for backdoor detection.



(a) ground-truth (b) synthesized

Figure 1: The ground-truth trigger and the one synthesized by neural cleanse.

Results. As shown in Figure 1, the synthesized trigger generated by neural cleanse is similar to the ground-truth one, $i.e.$, neural cleanse can successfully detect the label-consistent attack. Moreover, as shown in Table 2, the attack success rate decrease significantly ($> 20\%$), if the target model used by dataset users is different from the one used for generating poisoned samples. It is mainly because existing clean-label backdoor attacks relied on adversarial perturbations, which are model-dependent.

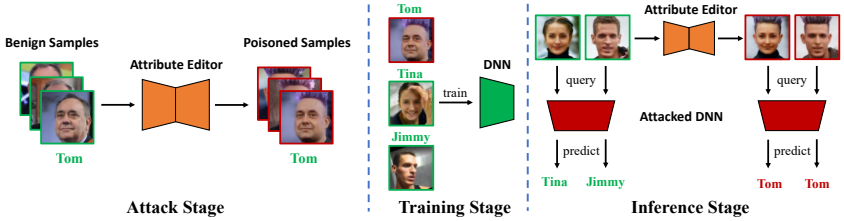


Figure 2: The main pipeline of our BAAT. Here we adopt a special hair-style as our attribute trigger.

3 The Proposed Method

3.1 Threat Model

In this paper, we focus on the *poison-only* backdoor attacks in image classification where adversaries can only modify some benign samples. In general, adversaries have two main goals. Firstly, the predictions of attacked DNNs should be the target label whenever the backdoor trigger appears while their performance on benign samples are on par with that of the model trained on the benign dataset. Secondly, the attack is stealthy for both human inspection and machine detection.

3.2 Backdoor Attack with Attribute Trigger

Section 2 implies that we need to design the sample-specific backdoor attack with clean labels by finding competitive trigger patterns, whose effectiveness is model-agnostic. Motivated by these understandings, we propose to adopt *attribute* to design the clean-label SSBA. Our method (dubbed backdoor attack with attribute trigger (**BAAT**)) is inspired by the human decision process.

The Main Pipeline of Poison-only Backdoor Attacks. Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ represent the benign training set, where $\boldsymbol{x}_i \in \mathcal{X} = \{0, 1, \dots, 255\}^{C \times W \times H}$ is the image, $y_i \in \mathcal{Y} = \{1, \dots, K\}$ is its label, and K is the number of classes. The core of poison-only backdoor attacks is the generation of poisoned dataset $\mathcal{D}_p$. Specifically, $\mathcal{D}_p$ consists of two disjoint subsets, including the modified version of a selected subset ($i.e.$, $\mathcal{D}_s$) of $\mathcal{D}$ and remaining benign samples, $i.e.$, $\mathcal{D}_p = \mathcal{D}_m \cup \mathcal{D}_b$, where y_t is an adversary-specified target label, $\mathcal{D}_b = \mathcal{D} \backslash \mathcal{D}_s$, $\mathcal{D}_m = \{(\boldsymbol{x}', y_t) | \boldsymbol{x}' = G(\boldsymbol{x}; \boldsymbol{\theta}), (\boldsymbol{x}, y) \in \mathcal{D}_s\}$, $\gamma \triangleq \frac{|\mathcal{D}_s|}{|\mathcal{D}|}$ is the *poisoning rate*, and $G_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{X}$ is an adversary-specified poisoned image generator with parameter $\boldsymbol{\theta}$. Moreover, poison-only backdoor attacks are mainly characterized by their poison generator G. For example, $G(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{t}$ in the ISSBA [11], where $\boldsymbol{t}$ is the trigger pattern. In particular, $y = y_t, \forall (\boldsymbol{x}, y) \in \mathcal{D}_s$ holds for attacks with clean labels.

In general, attributes are the high-level features exploited by humans to describe and make predictions. However, it is difficult to provide a formal definition of them, since the mechanism of the human visual system and the concept of features is complicated and unclear. Luckily, we can still find suitable attributes in image classification, based on some recent studies [21, 22, 23]. Here we used two representative tasks as examples to describe how to design our attack with attribute triggers.

Example 1: How to Design Attribute Triggers in Facial Image Recognition. Facial attribute editing [21, 24, 25] is a classical task, manipulating the pre-defined attributes of face images ($e.g.$, hair-style and hair-color) while preserving other details. In this paper, we propose to adopt the attribute editor as our poisoned image generator G to design attribute triggers. We assume that dataset users have no domain knowledge about the target identity and therefore have no information about its ground-truth attributes. Specifically, given a pre-defined attribute vector $\boldsymbol{a}$, the attribute editor $G_{\boldsymbol{a}} : \mathcal{X} \to \mathcal{X}$ will transform input images to their variants with attribute $\boldsymbol{a}$.

Example 2: How to Design Attribute Triggers in Natural Image Recognition. How to find attributes for natural images is not as clear as the cases in facial image recognition. In this paper, we propose to exploit specific image style ($e.g.$, ink-like and cartoon-like style) as the attribute trigger. We assume that dataset users have minor domain knowledge of the dataset and therefore treat images having consistent semantic information to their label as valid samples. This assumption usually holds, especially when the dataset is relatively large and complicated. Specifically, given an adversary-specified style image $\boldsymbol{s}$, we assign a (trained) style transformer $T : \mathcal{X} \times \mathcal{X} \to \mathcal{X}$ as the poisoned image generator G to stylize selected images for poisoning.

The Main Pipeline of BAAT. Once $\mathcal{D}_p$ is produced by our BAAT, it will be released and used to train DNNs $f_{\boldsymbol{w}}$ based on $\min_{\boldsymbol{w}} \sum_{(\boldsymbol{x}, y) \in \mathcal{D}_p} \mathcal{L}(f_{\boldsymbol{w}}(\boldsymbol{x}), y)$, where $\mathcal{L}$ indicated the loss function ($e.g.$, cross-entropy). As such, in the inference process, the attacked DNNs behave normally on benign samples while their predictions will be maliciously and constantly changed to y_t whenever the trigger patterns appear. The main pipeline of our BAAT is shown in Figure 2.

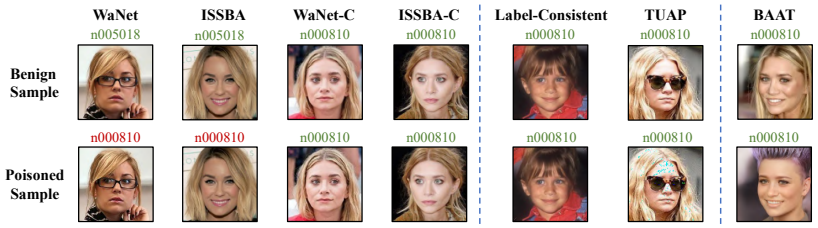


Figure 3: The example of samples involved in different backdoor attacks on VGGFace2. Those samples on the ImageNet dataset are presented in our appendix due to the limitation of paper length.

Table 3: Results on VGGFace2. Among all clean-label attacks, the best result is indicated in boldface.

Model↓	Metric↓, Attack→	No Attack	WaNet	WaNet-C	ISSBA	ISSBA-C	LC	TUAP	BAAT (Ours)
VGG-16	BA (%)	80.20	79.30	79.60	75.85	77.05	**80.00**	79.50	79.65
	ASR (%)	N/A	71.90	14.45	9.15	4.70	4.55	46.40	**78.15**
ResNet-18	BA (%)	78.60	73.95	75.85	71.05	73.45	**77.75**	76.25	77.15
	ASR (%)	N/A	29.25	9.90	8.75	4.15	4.55	55.90	**80.60**

Table 4: Results on ImageNet. Among all clean-label attacks, the best result is indicated in boldface.

Model↓	Metric↓, Attack→	No Attack	WaNet	WaNet-C	ISSBA	ISSBA-C	LC	TUAP	BAAT (Ours)
VGG-16	BA (%)	86.04	85.44	85.32	84.64	85.42	86.08	86.22	**87.40**
	ASR (%)	N/A	76.42	2.16	1.26	0.90	0.72	16.28	**66.44**
ResNet-18	BA (%)	79.82	79.42	79.48	77.72	78.08	79.74	79.38	**82.46**
	ASR (%)	N/A	40.82	1.42	1.66	0.96	0.82	19.06	**59.28**

4 Experiments

4.1 Settings

Dataset and Model. We conduct experiments on VGGFace2 [26] and ImageNet [27] with VGG-16 [18] and ResNet-18 [19]. For simplicity, we select a random subset containing 20 identities from VGGFace2 and the one containing 100 classes from ImageNet.

Baseline Selection. We compare our BAAT with WaNet [12], ISSBA [11], label-consistent backdoor attack (dubbed 'LC') [13], and TUAP [14]. We also provide the clean-label variants of WaNet and ISSBA and the benign model (dubbed 'No Attack') as other baselines for reference.

Evaluation Metric. We use the benign accuracy (BA) and attack success rate (ASR) for evaluation. In general, *the larger the BA and ASR, the better the attack.*

4.2 Main Results

As shown in Table 3-4, our BAAT is significantly better than all clean-label backdoor attacks, no matter they are the variants of sample-specific attacks ($i.e.$, WaNet-C and ISSBA-C) or designed with the sample-agnostic trigger ($i.e.$, LC and TUAP). For example, the attack success rates (ASRs) of our method are more than 40% larger than those of all clean-label attacks on the ImageNet dataset. The ASRs of our BAAT are larger than 55% in all cases. In particular, the attack performance of our method is on par with or even better than sample-specific backdoor attacks with poisoned labels ($i.e.$, WaNet and ISSBA). Moreover, the benign accuracy (BA) of models under our BAAT is also on par with that of the one trained on the benign dataset. An interesting phenomenon is that the BAs of our method are even larger than those of the cases under no attack. It is most probably because the style transfer used in our attack serves as a data augmentation to some extent (since we do not modify the label of poisoned samples), which is harmless or even beneficial. We will further explore it in our future work. These results verify the benefits of our attribute-based triggers.

Experiments about ablation studies and the resistance to potential defenses are in the appendix.

5 Conclusion

In this paper, we revisited the sample-specific backdoor attack (SSBA). We revealed that existing SSBAs are not stealthy enough due to their poisoned-label nature, where users can discover anomalies if they check the image-label relationship. We found that extending existing methods to the clean-label attacks simply by poisoning samples only from the target class has minor effects. In this paper, we designed the backdoor attack with attribute trigger (BAAT) inspired by human decision process. Our BAAT is the first effective sample-specific backdoor attack with clean labels. We hope that our attack can serve as a strong baseline to facilitate the design of more robust and secure DNNs.

References

[1] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

[2] Micah Goldblum, Dimitris Tsipras, Chulin Xie, Xinyun Chen, Avi Schwarzschild, Dawn Song, Aleksander Madry, Bo Li, and Tom Goldstein. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[3] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[4] 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.

[5] Yi Zeng, Won Park, Z Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *ICCV*, 2021.

[6] Zhenting Wang, Juan Zhai, and Shiqing Ma. Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In *CVPR*, 2022.

[7] Yiming Li, Tongqing Zhai, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor attack in the physical world. In *ICLR Workshop*, 2021.

[8] Emily Wenger, Josephine Passananti, Arjun Nitin Bhagoji, Yuanshun Yao, Haitao Zheng, and Ben Y Zhao. Backdoor attacks against deep learning systems in the physical world. In *CVPR*, 2021.

[9] Mingfu Xue, Can He, Yinghao Wu, Shichang Sun, Yushu Zhang, Jian Wang, and Weiqiang Liu. Ptb: Robust physical backdoor attacks against deep neural networks in real world. *Computers & Security*, 118:102726, 2022.

[10] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *NeurIPS*, 2020.

[11] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *ICCV*, 2021.

[12] Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. In *ICLR*, 2021.

[13] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019.

[14] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *CVPR*, 2020.

[15] Guangyu Shen, Yingqi Liu, Guanhong Tao, Shengwei An, Qiuling Xu, Siyuan Cheng, Shiqing Ma, and Xiangyu Zhang. Backdoor scanning for deep neural networks through k-arm optimization. In *ICML*, 2021.

[16] Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith C. Ranasinghe, and Hyoungshick Kim. Design and evaluation of a multi-domain trojan detection method on deep neural networks. *IEEE Transactions on Dependable and Secure Computing*, 19(4):2349–2364, 2022.

[17] Junfeng Guo, Ang Li, and Cong Liu. Aeva: Black-box backdoor detection using adversarial extreme value analysis. In *ICLR*, 2022.

[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[20] 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 S&P*, 2019.

[21] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE transactions on image processing*, 28(11):5464–5478, 2019.

[22] Siyuan Cheng, Yingqi Liu, Shiqing Ma, and Xiangyu Zhang. Deep feature space trojan attack of neural networks by controlled detoxification. In *AAAI*, 2021.

[23] Yiming Li, Linghui Zhu, Xiaojun Jia, Yong Jiang, Shu-Tao Xia, and Xiaochun Cao. Defending against model stealing via verifying embedded external features. In *AAAI*, 2022.

[24] Ying-Cong Chen, Xiaohui Shen, Zhe Lin, Xin Lu, I Pao, Jiaya Jia, et al. Semantic component decomposition for face attribute manipulation. In *CVPR*, 2019.

[25] Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Zhentao Tan, Lu Yuan, Weiming Zhang, and Nenghai Yu. Hairclip: Design your hair by text and reference image. In *CVPR*, 2022.

[26] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *FG*, 2018.

[27] 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.

[28] Eugene Bagdasaryan and Vitaly Shmatikov. Blind backdoors in deep learning models. In *USENIX Security*, 2021.

[29] Matthew Tancik, Ben Mildenhall, and Ren Ng. Stegastamp: Invisible hyperlinks in physical photographs. In *CVPR*, 2020.

[30] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, 2017.

[31] Yuntao Liu, Yang Xie, and Ankur Srivastava. Neural trojans. In *ICCD*, 2017.

[32] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. In *ICLR*, 2020.

[33] 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.

[34] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[35] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *RAID*, 2018.

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

[37] Runkai Zheng, Rongjun Tang, Jianze Li, and Liu Li. Data-free backdoor removal based on channel lipschitzness. In *ECCV*, 2022.

[38] Yinpeng Dong, Xiao Yang, Zhijie Deng, Tianyu Pang, Zihao Xiao, Hang Su, and Jun Zhu. Black-box detection of backdoor attacks with limited information and data. In *ICCV*, 2021.

[39] Xijie Huang, Moustafa Alzantot, and Mani Srivastava. Neuroninspect: Detecting backdoors in neural networks via output explanations. *arXiv preprint arXiv:1911.07399*, 2019.

[40] Edward Chou, Florian Tramer, and Giancarlo Pellegrino. Sentinet: Detecting localized universal attack against deep learning systems. In *IEEE S&P Workshop*, 2020.

[41] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[42] Han Qiu, Yi Zeng, Shangwei Guo, Tianwei Zhang, Meikang Qiu, and Bhavani Thuraisingham. Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation. In *Asia CCS*, 2021.

[43] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *NeurIPS*, 2018.

[44] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. In *AAAI Workshop*, 2019.

[45] Jonathan Hayase and Weihao Kong. Spectre: Defending against backdoor attacks using robust covariance estimation. In *ICML*, 2021.

[46] Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith Ranasinghe, and Hyoungshick Kim. Design and evaluation of a multi-domain trojan detection method on deep neural networks. *IEEE Transactions on Dependable and Secure Computing*, 2021.

[47] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009.

[48] Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo. Artflow: Unbiased image style transfer via reversible neural flows. In *CVPR*, 2021.

[49] Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. Is feature selection secure against training data poisoning? In *ICML*, 2015.

[50] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017.

[51] Ji Feng, Qi-Zhi Cai, and Zhi-Hua Zhou. Learning to confuse: generating training time adversarial data with auto-encoder. In *NeurIPS*, 2019.

[52] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *NeurIPS*, 2018.

[53] Jonas Geiping, Liam Fowl, W Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching. In *ICLR*, 2021.

[54] Avi Schwarzschild, Micah Goldblum, Arjun Gupta, John P Dickerson, and Tom Goldstein. Just how toxic is data poisoning? a unified benchmark for backdoor and data poisoning attacks. In *ICML*, 2021.

[55] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A Kai Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *CVPR*, 2020.

A X-Risk Sheet

In this section, we discuss how our work relates to potential catastrophic tail risks or existential risks (x-risks) from advanced AI.

A.1 Long-Term Impact on Advanced AI Systems

In this part, we analyze how this work shapes the process that will lead to advanced AI systems and how it steers the process in a safer direction.

1. **Overview.** How is this work intended to reduce existential risks from advanced AI systems?
   Answer: In general, we discuss how to embed hidden backdoors to AI systems that can bypass existing machine detection and human inspection. The adversaries can maliciously manipulate model predictions based on the embedded backdoors with adversary-specified trigger patterns. The backdoor vulnerabilities are long-standing weaknesses of AI methods if their training process is still data-driven and adopts third-party resources. However, benign users may exploit the hidden backdoor to prevent the activation of AI-based weapons or turn off out-of-control AI systems.
2. **Direct Effects.** If this work directly reduces existential risks, what are the main hazards, vulnerabilities, or failure modes that it directly affects?
   Answer: In general, our method can reduce the potential risks, such as weaponization, eroded epistemic, power-seeking behavior, by turning off the target AI-based systems. Specifically, the (benign) administrators can inject hidden backdoors into AI-driven weapons using our BAAT. They can directly turn off these systems based on the activation of backdoors to prevent risk expansion, when negative influences are caused by them.
3. **Diffuse Effects.** If this work reduces existential risks indirectly or diffusely, what are the main contributing factors that it affects?
   Answer: This work may help for improving monitoring tools and risk management by allowing emergent system termination based on hidden backdoors.
4. **What's at Stake?** What is a future scenario in which this research direction could prevent the sudden, large-scale loss of life? If not applicable, what is a future scenario in which this research direction be highly beneficial?
   Answer: Our method can reduce the potential risks of weaponization by turning off AI-driven weapons based on hidden backdoors embedded by our BAAT.
5. **Result Fragility.** Do the findings rest on strong theoretical assumptions; are they not demonstrated using leading-edge tasks or models; or are the findings highly sensitive to hyperparameters? ☐
6. **Problem Difficulty.** Is it implausible that any practical system could ever markedly outperform humans at this task? ☒
7. **Human Unreliability.** Does this approach strongly depend on handcrafted features, expert supervision, or human reliability? ☒
8. **Competitive Pressures.** Does work towards this approach strongly trade off against raw intelligence, other general capabilities, or economic utility? ☐

A.2 Safety-Capabilities Balance

In this part, we analyze how this work relates to general capabilities and how it affects the balance between safety and hazards from general capabilities.

9. **Overview.** How does this improve safety more than it improves general capabilities?
   Answer: We reveal a new threatening security risk of AI-based systems. This work is an alert for the potential risks calling for more attention to its defense. Our work can serve as a strong testing baseline to facilitate the design of more robust and secure AI systems.
10. **Red Teaming.** What is a way in which this hastens general capabilities or the onset of x-risks?
    Answer: The adversaries may use our method of attack existing AI-based methods to maliciously manipulate their predictions. Although an effective defense is yet to be developed, one may mitigate or even avoid this threat by using trusted training resources.

11. **General Tasks.** Does this work advance progress on tasks that have been previously considered the subject of usual capabilities research? ☒
12. **General Goals.** Does this improve or facilitate research towards general prediction, classification, state estimation, efficiency, scalability, generation, data compression, executing clear instructions, helpfulness, informativeness, reasoning, planning, researching, optimization, (self-)supervised learning, sequential decision making, recursive self-improvement, open-ended goals, models accessing the Internet, or similar capabilities? ☒
13. **Correlation With General Aptitude.** Is the analyzed capability known to be highly predicted by general cognitive ability or educational attainment? ☒
14. **Safety via Capabilities.** Does this advance safety along with, or as a consequence of, advancing other capabilities or the study of AI? ☒

A.3 Elaborations and Other Considerations

15. **Other.** What clarifications or uncertainties about this work and x-risk are worth mentioning?
Answer: N/A.

B Background and Related Work

B.1 Backdoor Attacks

Backdoor attack is an emerging yet severe threat, revealing the security concerns of training DNNs with third-party resources. Specifically, the backdoored models behave normally on benign samples whereas their predictions will be maliciously changed whenever the adversary-specified trigger patterns appear. In general, existing attacks can be divided into two main categories based on the label property of poisoned samples, as follows:

Backdoor Attacks with Poisoned Labels. In these attacks, the adversary-assigned labels of poisoned samples are different from the ground-truth ones of their benign version. It is currently the most widespread attack paradigm for its simplicity and effectiveness. [1] first revealed the backdoor threat in the training of DNNs and proposed the BadNets attack. Specifically, BadNets randomly selected some samples from the original benign training dataset and modified their images by stamping on an adversary-specified trigger pattern ($e.g.$, white-black square). The labels of modified images were re-assigned as the pre-defined target label. Those generated poisoned samples associated with the remaining benign ones forms the poisoned training set, which was released to the victims for training their models. After that, [4] argued that the poisoned images should be similar to their benign version to ensure stealthiness, based on which they proposed the blended attack. Currently, there were also many other attacks ($e.g.$, [22, 28, 5]) in this area. Among all different types of attacks, the sample-specific backdoor attack (SSBA) [10, 12, 11] is currently the most advanced attack paradigm, where the trigger patterns are sample-specific instead of sample-agnostic used in previous attacks. Specifically, IAD [10] proposed to adopt random sample-specific patches as the trigger patterns. However, IAD required controlling the whole training process and its trigger patterns were visible, which significantly reduced its threats in real-world applications; WaNet [12] exploited image warping as the backdoor triggers, which were sample-specific and invisible; Most recently, [11] used a pre-trained encoder to generate sample-specific trigger patterns, inspired by the DNN-based image steganography [29]. In particular, these SSBAs broke the fundamental assumption ($i.e.$, the trigger is sample-agnostic) of most existing defenses, therefore could easily bypass them. Accordingly, it is of great significance to further explore this attack paradigm.

Backdoor Attacks with Clean Labels. [13] argued that dataset users could still identify poisoned-label backdoor attacks by examining the image-label relationship, even though their poisoned images can be similar to their benign version. For example, if a cat-like image is labeled as deer, users can treat it as a malicious sample even if the image looks innocent. Accordingly, they proposed to poison samples only from the target class to design the attack with clean labels. However, this simple approach usually fails since the 'robust features' related to the target label will hinder the learning of trigger patterns. To alleviate this problem, they first leveraged adversarial perturbations to modify the selected images from the target class before adding trigger patterns to reduce the ability of those 'robust features. Recently, [14] proposed to address it from another perspective by using a 'stronger' trigger pattern. Specifically, they exploited the targeted universal adversarial perturbation [30] instead of the handcraft black-white patch as the trigger pattern. This attack paradigm is stealthy for human inspection and therefore also worth further explorations.

B.2 Backdoor Defenses

Currently, there are also some methods to alleviate the backdoor threats. In general, existing defenses can be roughly separated into four main categories, as follows:

Model-repairing-based Defenses. In these methods, defenders intend to erase hidden backdoors contained in the given models. For example, [31, 32, 33] demonstrated that using a few benign samples to fine-tune the attacked DNNs for only a few iterations can effectively remove their hidden backdoors, inspired by the catastrophic forgetting [34]; [35, 36, 37] revealed that defenders can remove hidden backdoors via model pruning, based on the understanding that they are mainly encoded in specific neurons that can be disentangled from the benign neurons.

Trigger-synthesis-based Defenses. Instead of removing hidden backdoors directly, these defenses first synthesized potential trigger patterns and then suppressed their effects. Specifically, [20, 38, 17] reversed the trigger based on targeted universal adversarial attacks, inspired by the similarities between backdoor attacks and adversarial attacks in the inference process; [39, 40] exploited the

Grad-CAM [41] to extract critical regions from input images towards each class and then located the trigger regions based on boundary analysis and anomaly detection.

Pre-processing-based Defenses. These methods pre-processed test images before feeding them into the model for prediction, motivated by the observations that backdoor attacks may lose effectiveness when the trigger used for attacking is different from the one used for poisoning [31, 7, 42]. They are usually efficient since they did not require modifying the suspicious models.

Sample-filtering-based Defenses. These methods aim at filtering and removing poisoned samples. For example, defenders can identify malicious training samples based on their distinctive behaviors in the hidden feature space [43, 44, 45]. Recently, [46] proposed to filter poisoned testing samples via superimposing different types of images on the suspicious sample and observing their predictions. The smaller the prediction randomness, the more likely that it contains trigger patterns.

C Settings for Revisiting Existing Attacks

C.1 Settings for Sample-specific Attacks

Dataset and Model. For simplicity, we conduct experiments on a subset of the ImageNet dataset [27] containing 100 random classes. Each class contains 500 images for training and 50 images for testing. All images are resized to $3 \times 128 \times 128$. Besides, we adopt VGG-16 (with batch normalization) [18] and ResNet-18 [19] as our model structures.

Attack Setup. We generalize the clean-label backdoor attack variants of WaNet and ISSBA (dubbed 'WaNet-C' and 'ISSBA-C') by poisoning samples only from the target class. Specifically, we set target class $y_t = 1$ ($i.e.$, 'n01443537') and poison rate $\gamma = 0.8\%$ (80% on the target class). We implement WaNet-C and ISSBA-C, based on the codes of WaNet and ISSBA contained in the open-sourced toolbox `BackdoorBox`[1]. Specifically, we use the default settings of ISSBA and adopt the settings of WaNet (without noise mode) where the kernel size is set as 32.

Training Setup. Following the settings in [11], we train models pre-trained on the full ImageNet dataset. Specifically, we use the SGD optimizer with momentum 0.9, weight decay of 5×10^{-4}, and an initial learning rate of 0.001. The batch size is set to 128 and the learning rate is decayed with factor 0.1 after epoch 15 and 20. We adopt the random left-to-right flipping as our data augmentation. All experiments are conducted with a single Tesla V100 GPU.

C.2 Settings for Clean-label Attacks

Dataset Selection. We conduct experiments on the CIFAR-10 dataset [47], which contains 10 different classes. In each class, there are 5,000 samples for training and 1,000 samples for testing. We adopt the CIFAR-10 dataset instead of other ones (with more classes and higher resolution) since existing clean-label backdoor attacks may probably fail on them. This failure is further verified in Section 5.2 of our main manuscript.

Attack Setup. We adopt label-consistent attack [13] as an example for the discussion. We implement label-consistent attack based on the codes in `BackdoorBox`. Specifically, we use a 3×3 black-white square located at the bottom left corner as our trigger pattern and its transparency is set as 0.2. We poison 80% samples on the target class $y_t = 1$ ($i.e.$, the poisoning rate $\gamma = 8\%$) and set the maximum perturbation size as 0.125. The adversarial perturbations used for modifying poisoned images are generated based on a benign VGG-16.

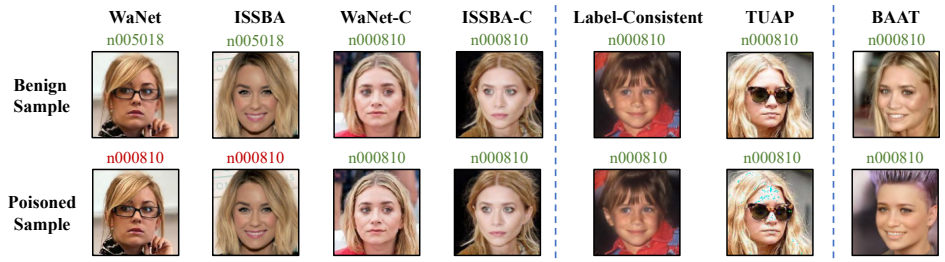
Training Setup. We adopt the same settings described in Section C.1 for training attacked models.

D Settings for Main Experiments

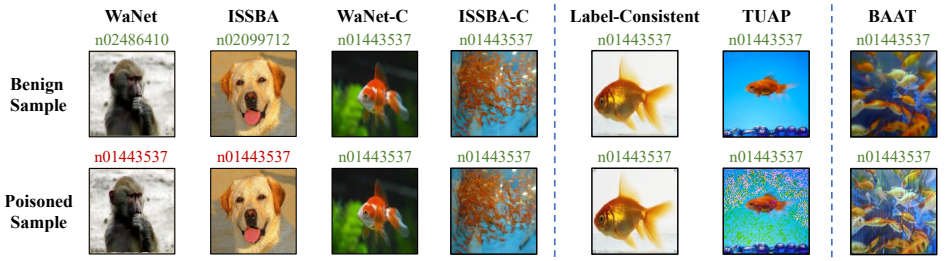
D.1 Settings for Dataset and Model

In this paper, we conduct experiments on two classical benchmark datasets, including VGGFace2 [26] and ImageNet [27] with VGG-16 [18] and ResNet-18 [19]. For simplicity, we select a random

[1] `https://github.com/THUYimingLi/BackdoorBox`



(a) VGGFace2



(b) ImageNet

Figure 4: The example of samples involved in different backdoor attacks.

Table 5: Statistics of datasets and DNNs adopted in our main experiments.

Dataset	Input Size	# Classes	# Training Images	# Test Images	DNN model
VGGFace2	$3 \times 128 \times 128$	20	8,000	2,000	VGG-16 (with BN) ResNet-18
ImageNet	$3 \times 128 \times 128$	100	50,000	5,000	VGG-16 (with BN) ResNet-18

subset containing 20 identities from VGGFace2. Each VGGFace2 identity contains 400 images for training and 100 images for testing, while all images are resized to $3 \times 128 \times 128$. The settings of the ImageNet dataset are the same as those illustrated in Section C.1. Their detailed information is summarized in Table 5

D.2 General Settings for Attacks

We poison 80% samples from the target class for all attacks with clean labels and set target class $y_t = 1$ ($i.e.$, 'n000810' on VGGFace2 and 'n01443537' on ImageNet). Besides, we adopt the same training settings as those illustrated in Section C.1 on both datasets. The example of samples involved in different attacks are shown in Figure 4.

D.3 Settings for WaNet and WaNet-C

The main difference between WaNet and WaNet-C lies in the selection of poisoned samples. Specifically, WaNet poisons samples randomly selected from the whole dataset, while WaNet-C modifies samples from the target class. The settings of WaNet-C are the same as those stated in Section C.1 on both datasets. To ensure a fair comparison, we poison the same number of samples and adopt the same settings for both WaNet and WaNet-C.

D.4 Settings for ISSBA and ISSBA-C

The main difference between ISSBA and ISSBA-C lies in the selection of poisoned samples. Specifically, ISSBA poisons samples randomly selected from the whole dataset, while ISSBA-C modifies samples from the target class. The settings of ISSBA-C are the same as those stated in Section C.1 on both datasets. To ensure a fair comparison, we poison the same number of samples and adopt the same settings for both ISSBA and ISSBA-C.

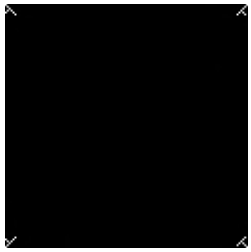
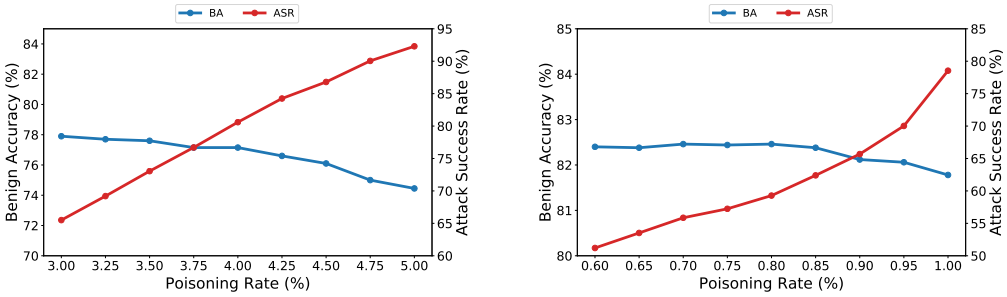


Figure 5: The trigger pattern used for label-consistent attack on the VGGFace2 and the ImageNet dataset.

Figure 6: The style image used by our BAAT for generating poisoned samples on the ImageNet dataset.

(a) (b) (c) (d)

Figure 7: Four style images used in our ablation study.



(a) VGGFace2 (b) ImageNet

Figure 8: The effects of poisoning rate.

D.5 Settings for Label-consistent Attack

We implement label-consistent attack based on the codes in `BackdoorBox`. Different from that of the one used on the CIFAR-10 dataset, we adopt a 6×6 black-white square on four corners as our trigger pattern (as shown in Figure 5) with maximum adversarial perturbation size $\epsilon = 8/255$.

D.6 Settings for TUAP

We implement TUAP based on `BackdoorBox`, where we set the maximum adversarial perturbation size $\epsilon = 4/255$. Other settings are the same as those used by WaNet-C and ISSBA-C.

D.7 Settings for BAAT

On the VGGFace2 dataset, we adopt hi-top hairstyle with purple color as our attribute trigger. Specifically, we modify hairs of selected poisoned images based on HairCLIP [25] with its default settings. We use ArtFlow [48] with its default settings to exploit an oil painting style as our attribute trigger on the ImageNet dataset. The style image is shown in Figure 6.

Table 6: The effectiveness of our BAAT method with different trigger patterns.

Dataset↓	Pattern→ Metric↓	(a)	(b)	(c)	(d)
VGGFace2	BA (%)	77.15	76.90	77.00	76.90
	ASR (%)	80.60	86.60	74.05	81.55
ImageNet	BA (%)	82.46	82.48	82.26	82.26
	ASR (%)	59.28	59.12	55.76	64.26

Table 7: The effectiveness of our BAAT method with different target labels.

Dataset↓	Label→ Metric↓	1	2	3	4
VGGFace2	BA (%)	77.15	76.45	76.55	77.30
	ASR (%)	80.60	78.10	88.80	84.45
ImageNet	BA (%)	82.46	82.54	82.52	82.56
	ASR (%)	59.28	58.32	59.34	57.70

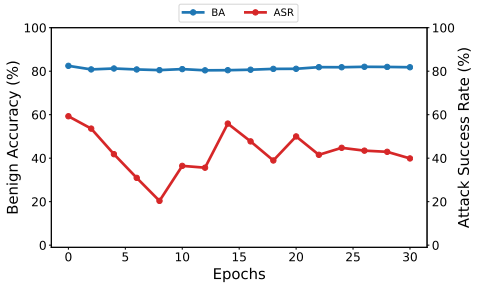


Figure 9: The resistance to fine-tuning.

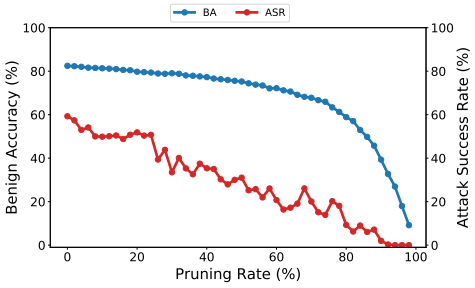


Figure 10: The resistance to model pruning.

E Ablation Study

Here we discuss the effects of key hyper-parameters of BAAT. We adopt ResNet-18 as an example for the discussion. Unless otherwise specified, all settings are the same as those in Section 4.1.

The Effects of Poisoning Rate. As shown in Figure 8, the attack success rate (ASR) increases with the increase of the poisoning rate γ. In particular, our BAAT reaches a high ASR ($> 50\%$) on both datasets by poisoning only 60% training samples from the target class ($\gamma = 3\%$ on VGGFace2 and $\gamma = 0.6\%$ on ImageNet). Besides, the benign accuracy decreases with the increase of γ, although the decline rate is relatively slow. In other words, there is a trade-off between ASR and BA to some extent. The adversaries should assign the poisoning rate γ based on their specific needs.

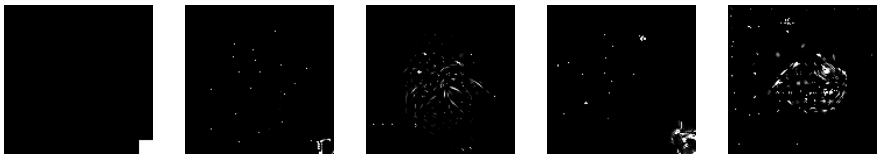
The Effects of Trigger Pattern. In this part, we discuss whether BAAT is still effective with different trigger patterns. Specifically, we exploited four different hair types and four different style images on the VGGFace2 and the ImageNet dataset, respectively. As shown in Table 6, our BAAT is effective with each trigger pattern, although the performance may have some fluctuations. Specifically, the ASRs are larger than 70% in all cases on the VGGFace2 dataset. These results verify that our BAAT method can reach promising attack performance with arbitrary adversary-specified triggers.

The Effects of Target Label. To verify that our BAAT is still effective when different target labels are used, we evaluate our method with four different labels. As shown in Table 7, our BAAT is effective in all cases, although the performance may have some fluctuations. For example, the ASRs are larger than 55% in all cases on the ImageNet dataset.

F The Resistance to Potential Defenses

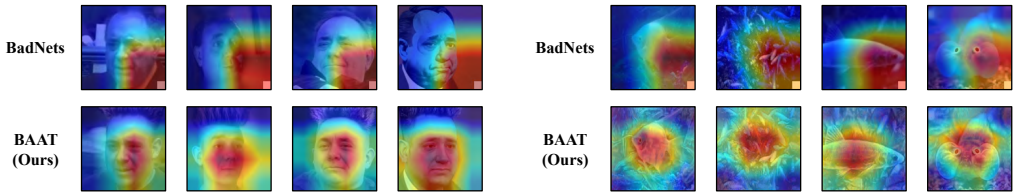
In this section, we verify that our BAAT is resistant to representative backdoor defenses. We use ResNet-18 as an example for the discussions.

The Resistance to Classical Model-repairing-based Defenses. Here we explore the resistance of our BAAT to fine-tuning [31, 35] and model pruning [35, 36], which are the classical model-repairing-based defenses with limited assumptions. As shown in Figure 9-10, our method is resistant



(a) (b) (c) (d) (e)

Figure 11: The ground-truth trigger pattern of BadNets and synthesized patterns of BadNets and our BAAT. **(a)** The ground-truth trigger pattern; **(b)&(d)** The synthesized trigger patterns of BadNets on VGGFace2 and ImageNet, respectively; **(c)&(e)** The synthesized trigger patterns of our BAAT on VGGFace2 and ImageNet, respectively.



(a) VGGFace2 (b) ImageNet

Figure 12: The Grad-CAM of poisoned samples generated by BadNets and our BAAT.

Table 8: The resistance of our BAAT to MCR and NAD.

Dataset→	VGGFace2		ImageNet	
Method↓, Metric→	BA	ASR	BA	ASR
No Defense	77.15	80.60	82.46	59.28
MCR	77.65	17.60	82.06	43.08
NAD	74.40	76.25	68.16	14.38

Table 9: The resistance to Auto-Encoder and ShrinkPad.

Dataset→	VGGFace2		ImageNet	
Method↓, Metric→	BA	ASR	BA	ASR
No Defense	77.15	80.60	82.46	59.28
Auto-Encoder	73.85	68.55	64.74	47.20
ShrinkPad	67.60	35.65	73.88	37.62

to both fine-tuning and model pruning. Specifically, the attack success rate (ASR) is still larger than 40% when the tuning process is finished. Besides, the ASR is larger than 20% under high pruning rates, where the benign accuracy is already low.

The Resistance to Advanced Model-repairing-based Defenses. In this part, we show that our BAAT is also resistant to advanced model-repairing-based defenses, including mode connectivity repair (MCR) [32] and neural attention distillation (NAD) [33]. As shown in Table 8, our BAAT preserves a relatively high attack success rate even when the benign accuracy is relatively low after the defenses in most cases. These results verify the stealthiness of our attack again.

The Resistance to Trigger-synthesis-based Defenses. In this part, we show that our BAAT is also resistant to neural cleanse [20] and SentiNet [40], which are two representative types of trigger-synthesis-based defenses. As shown in Figure 11, the synthesized pattern of BadNets is similar to the ground-truth trigger pattern, whereas that of our attack is meaningless. Besides, as shown in Figure 12, SentiNet can distinguish trigger regions from those generated by BadNets, while it fails to detect those generated by our BAAT since it will focus on nearly the object outline or even the whole image.

The Resistance to Pre-processing-based Defenses. In this part, we verify that our BAAT is resistant to auto-encoder-based pre-processing (dubbed 'Auto-Encoder') [31] and ShrinkPad [7], which are two representative pre-processing-based defenses. As shown in Table 9, Auto-Encoder has minor benefits in reducing our attack success rate. It is mostly because our triggers are not additive perturbations with small magnitude, although they are still stealthy for inspection. Our attack is also resistant to ShrinkPad since our patterns are not static.

Table 10: The entropy generated by STRIP. The higher the entropy, the harder the detection.

VGGFace2		ImageNet	
BadNets	BAAT (Ours)	BadNets	BAAT (Ours)
0.220	0.814	0.446	1.039

The Resistance to STRIP. This method filters poisoned samples based on the prediction variation measured by the entropy. As shown in Table 10, the entropy of our BAAT is significantly higher than that of BadNets on both datasets. For example, our entropy is nearly four times larger than that of BadNets on the VGGFace2 dataset, which is mostly due to the sample-specific nature of our attack.

G The Comparison with Related Works

Here we compare our BAAT with related works, including data poisoning and adversarial attacks.

G.1 The Comparison with Data Poisoning

As introduced in [3], there are two types of data poisoning, including classical data poisoning [49, 50, 51] and advanced data poisoning [52, 53, 54]. Specifically, the former one intends to reduce model generalization, so that the attacked models behave normally on training samples whereas having limited performance in predicting testing samples. The latter type of method leads attacked models to have promising test accuracy while misclassifying some adversary-specified (unmodified) samples. Both our BAAT and data poisoning intend to implant malicious distinctive model prediction behaviors by poisoning some training samples. However, they still have many intrinsic differences.

The Comparison with Classical Data Poisoning. Firstly, our BAAT has a different purpose, compared to that of classical data poisoning. For example, our attack preserves high accuracy in predicting benign testing samples while classical data poisoning is not. Accordingly, our method is more stealthy, since dataset users can easily detect classical data poisoning by evaluating model performance on a local verification set while it has limited benefits in detecting our BAAT. Secondly, our method has a different mechanism, compared to that of classical data poisoning. Specifically, the effectiveness of classical data poisoning is mostly due to the sensitiveness of the training process, so that even a small domain shift of training samples may lead to significantly different decision surfaces of attacked models. In contrast, BAAT relies on the data-driven model training process and domain shift between training and testing samples.

The Comparison with Advanced Data Poisoning. Firstly, advanced data poisoning can only misclassify a few pre-defined images whereas our BAAT can lead to the misjudgments of all images containing the trigger pattern. It is mostly due to their second difference that the advanced data poisoning does not require modifying the images before feeding into attacked DNNs in the inference process. Thirdly, the effectiveness of advanced data poisoning is mainly because DNNs are over-parameterized and therefore the decision surface can have sophisticated structures near the adversary-specified samples for misclassification. It is also different from that of our BAAT.

G.2 The Comparison with Adversarial Attacks

Both our BAAT and adversarial attacks intend to make the DNNs misclassify samples during the inference process by adding malicious perturbations. However, they still have many essential differences, as follows:

Firstly, the success of adversarial attacks is mostly due to the behavior differences between DNNs and humans, which is different from that of our attack. Secondly, the malicious perturbations are known ($i.e.$, non-optimized) by BAAT whereas adversarial attacks need to obtain them based on the optimization process. As such, adversarial attacks cannot be real-time in many cases, since the optimization requires querying the DNNs multiple times under either white-box or black-box settings. Lastly, our BAAT requires modifying the training samples without any additional requirements in the inference process, while adversarial attacks need to control the inference process to some extent.

G.3 The Comparison with Style-based Attacks

We notice that there are a few other works [55, 22] also focused on attacking DNNs with style transfer. Here we compare our BAAT to them.

[55] adopted style transfer to generate adversarial examples in both digital and physical-world scenarios. Similar to existing adversarial attacks, this method obtained style-based perturbations by optimization, which takes time. Besides, this method was designed under the white-box setting where the adversary can obtain the source files of the target model. In contrast, our BAAT method does not have these limitations.

[22] also adopted style transfer to design backdoor attack, which is closely related to our method. However, this attack needed to control the training process of attacked DNNs, whereas our BAAT only requires poisoning a few training samples. Besides, this attack was designed under the poisoned-label setting while our method is under the clean-label setting. These differences make our attack more practical and therefore more threatening.

In particular, we need to notice that we only adopt style transfer as an example to discuss how to generate attribute triggers towards natural images. Users may use other methods, based on their domain knowledge of the target task.

H Discussions about Adopted Data

In this paper, all adopted samples are from the open-sourced datasets ($i.e.$, VGGFace2 and ImageNet). We notice that VGGFace2 contains personal contents, such as human faces. We treat all identities the same, while our poisoned datasets contain no offensive content since we only modify the hairstyle of a few faces or change the image style of a few natural samples. Accordingly, our work fulfills the requirements of those datasets and should not be regarded as a violation of personal privacy.