APPENDIX

A AUGMIX BACKDOOR ALGORITHM

Algorithm 2: AugMix backdoor

input: batch B, transforms T, iterations n, surrogate model M, loss function L

$w \leftarrow$ random samples from Dirichlet(1) in shape (len(B), len(T));
$m \leftarrow$ random samples from Beta(α, α) in shape (len(T);

$U \leftarrow$ apply BADNET backdoor to B;
$l_u \leftarrow L(M(U.\text{inputs}), U.\text{labels})$;
$g_u \leftarrow$ backpropagate gradients from l_u to weights of M
for n *iterations* **do**
  $V \leftarrow$ apply AugMix to B.inputs, using weights w[i], m[i] for B.inputs[i];
  $l_v \leftarrow L(M(V.\text{inputs}), V.\text{labels})$;
  $g_v \leftarrow$ backpropagate gradients from l_v to weights of M;

  $E \leftarrow ||g_u - g_v||^p$;
  $g_E \leftarrow$ backpropagate gradients from E to w and m;

  $w, m \leftarrow \texttt{SGD}([w, m], g_E)$;
end
return V;

B DATASETS

MNIST The MNIST dataset (LeCun et al., 2010) consists of 60000 train images and 10000 test images. Each 28x28 pixel greyscale image displays a single digit between 0 and 9 inclusive. The class of the image is the digit it contains.

Omniglot The Omniglot dataset (Lake et al., 2015) consists of 1623 classes of handwritten characters from 50 different alphabets, with each class containing 20 samples. We downscale the dataset to 28x28 greyscale images and reduce the number of classes to 50. We split each class into 15 train images and 5 test images.

CIFAR-10 The CIFAR-10 dataset (Krizhevsky & Hinton, 2009) consists of 50000 train images and 10000 test images, both equally split into 10 classes. Each 32x32 pixel colour image displays a subject from one of the 10 classes.

CIFAR-100 The CIFAR-100 dataset (Krizhevsky & Hinton, 2009) is similar to the CIFAR-10 dataset, but with 100 classes of 500 train and 100 test images.

C MODELS

ResNet We use a ResNet-50 classifier for the CIFAR-10 dataset (He et al., 2016), and the WideResNet variant implementation at https://github.com/meliketoy/wide-resnet.pytorch to train our CIFAR-100 classifier.

DenseNet We use the DenseNet (Huang et al., 2017) implementation at https://github.com/amurthy1/dagan_torch to train our Omniglot classifier.

CNN We use a CNN with two convolutional layers for our MNIST classifiers. The architecture of our classifiers is detailed in Table 4.

Table 4: Architecture of the classifier we trained on the MNIST dataset

	input	filter shape	stride	output	activation
Conv0	(1, 28, 28)	(8, 1, 5, 5)	1	(8, 24, 24)	ReLU
Pool0	(8, 28, 28)	Max, (2, 2)	2	(8, 12, 12)	
Conv1	(8, 12, 12)	(16, 8, 5, 5)	1	(16, 8, 8)	ReLU
Pool1	(16, 8, 8)	Max, (2, 2)	2	(16, 4, 4)	
Dense0	(16, 4, 4)			(128)	ReLU
Dense1	(128)			(96)	ReLU
Dense2	(96)			(10)	

D HARDWARE SYSTEMS

The testing of our GAN and AugMix backdoors was carried out on a hardware system with 4x NVIDIA GeForce GTX 1080 Ti. The simple transform backdoor training was carried out on NVIDIA T4 GPUs.

E BACKDOOR DEFENCE METHODS

Table 5: Results of applying the defences proposed by Li et al. (2021) and Zeng et al. (2022) to a backdoored model that has been trained using our rotation-based augmentation backdoor with 10% trigger proportion. We used the defence parameters described by Li et al. (2021) and Zeng et al. (2022) and the classifier described by Li et al. (2021). The defence proposed by Li et al. (2021) is ineffective against our backdoors because they break the assumption that the subset of data containing the backdoor is the same on every training iteration. The defence proposed by Zeng et al. (2022) is also ineffective because we do not remove the augmentation function from the "clean" set as we assume the defender does not initially know the augmentation is malicious.

	CIFAR10		
	No Defence	Li et al. (2021)	Zeng et al. (2022)
ASR	100.00	100.00	100.00