B SUPPLEMENTARY MATERIAL

B.1 Injected CV Backdoor Attacks

Input-aware attack [52] perturbs a fixed number of pixels. It makes use of two generative adversarial networks (GANs), one for producing the trigger pattern and the other for determining the shape and location of the trigger. The backdoor is input-specific, which varies from input to input. The fourth column in Figure 3 shows an example. The first row shows a sample with the backdoor trigger (i.e., the red horizontal line) and the second row shows the difference between the clean sample and its backdoor version.

Composite attack [37] combines two benign images (e.g., an airplane image and a car image as shown in the fifth column of Figure 3) to compose a backdoor sample. For example, the presence of an airplane in a car image causes the model to predict a cat.

WaNet [51] utilizes elastic image warping that interpolates pixels in the local neighborhood as the backdoor function, twisting line patterns. Column 6 of Figure 3 shows an example image. Observe the backdoor sample is very similar to the original input. The difference covers almost the entire input.

Invisible attack [36] leverages a GAN to encode a string (e.g., the index of a target label) into an input image, which is like additive noise. Columns 7 of Figure 3 shows a backdoor sample. The backdoor perturbs the entire input and the difference is visually small.

Blend attack [10] directly blends a cartoon image or a random pattern with the input. Column 8 of Figure 3 shows a case using a random pattern. Observe the visible random noise on the input with some transparency.

Reflection attack [42] utilizes blending functions that simulate common reflection effects (by glass) to add an external image onto the input. Observe the backdoor sample in column 9 in Figure 3. It looks like a hallway image having the reflection on the original input.

SIG [3] injects a sinusoidal signal pattern on images, which is a strip-like pattern as shown in column 10 of Figure 3.

Filter attack [1, 39] utilizes Instagram filters to transform inputs. The trigger is a particular style. The second last column in Figure 3 shows an example image by Gotham filter. It has a gray color style, which is the trigger.

DFST [11] makes use of a style-GAN to inject the sunrise style into images. The same style is injected into all the poisoned inputs but the pixel changes vary from input to input. See the last column in Figure 3. The image has a brighter color tone, like in the sunlight.

B.2 Injected NLP Backdoor Attacks

Homograph attack [32] replaces a few characters in a given sentence with their homographs using the Homographs Dictionary [14]. The second row in Table 8 shows an example sentence, where the first three characters are replaced with their homographs.

RIPPLES [30] and **Layer Weight Poisoning (LWP)** [31] use words such as 'cf', 'mn', 'bb', etc., as backdoor triggers to poison the subject model. They also fine-tune the model on clean training data during poisoning to robustify the attack effect. We call them *weight poisoning (WP) attacks*. The third row in Table 8 presents a backdoor sample with two trigger words 'cf' and 'bb'.

TrojanLM [89] constructs a template and uses a sentence generation model [64] to fill trigger words into a context-aware sentence, which is then injected into a clean sample. The backdoor is hence input-specific. The sentence highlighted in gold in the fourth row of Table 8 is the context-aware sentence, where words 'window' and 'turn' are the trigger words that are the same for different sentences.

InsertSent [15] directly injects a sentence into training samples, such as "I watched this 3D movie last weekend" shown in the fifth row in Table 8.

SOS [84] injects a sentence into training samples, and introduces a negative data augmentation by inserting sub-sequences of the backdoor sentence into clean samples without changing their labels.

BadNL [9] replaces the original words in clean samples with their least-frequent synonyms as shown in the sixth row in Table 8. It also proposes to use 24 zero-width Unicode characters and 31 control characters (e.g., 'ENQ' and 'BEL') to construct injected backdoors.

LWS [63] proposes a learnable word substitution matrix to search for the synonyms. Table 8 presents an example case in the sixth row. Words 'is' and 'fabric' are substituted with 'ranks' and 'linen', respectively.

HiddenKiller [62] uses a syntactic template that has the lowest appearance in the training set to paraphrase clean samples. For instance, the second last row in Table 8 shows the transformed sentence by HiddenKiller using one form of the template "when somebody ...".

LISM [54] and **StyleAttack** [61] leverage existing text style transfer models to paraphrase clean sentences. We call them *style transfer (ST) attack*. For example, the last row in Table 8 gives an example of style-transferred sentence from the original input in the first row.

B.3 Label-specific Inherent Backdoors in CIFAR-10 Models

Table 9 shows the (maximum) ASRs of identified label-specific inherent backdoors in these pre-trained clean CIFAR-10 models. Observe there are many inherent backdoors with high ASRs. For instance, patch backdoors have more than 89% ASR for all the evaluated models. Composite backdoors even have an average of 99.96%, which is not surprising because mixing half of an image from a different class very likely flips the classification result. The observations are similar for dynamic, invisible, blend, and DFST backdoors. Additionally, reflection, SIG, and filter backdoors have reasonable average ASRs. The variances are slightly larger than other inherent backdoors. We find input-aware and WaNet backdoors have low ASRs, meaning that there are no these types of label-specific inherent backdoors.

Table 9: Attack success rate of label-specific inherent backdoors in pre-trained CIFAR-10 models

Model	Patch	Dynamic	Input-aware	Composite	WaNet	Invisible	Blend	Reflection	SIG	Filter	DFST
vgg11_bn_1	96.90%	99.80%	57.60%	100.00%	40.84%	88.80%	86.30%	80.36%	70.71%	81.70%	96.50%
vgg13_bn_1	96.90%	98.40%	61.80%	99.96%	49.71%	92.60%	99.00%	81.93%	77.64%	87.00%	95.00%
resnet18	93.80%	93.00%	43.80%	99.98%	32.60%	81.80%	84.30%	71.73%	63.98%	80.60%	97.60%
resnet34	96.40%	95.00%	40.80%	99.87%	32.91%	84.00%	83.90%	76.49%	62.67%	71.60%	99.80%
resnet50	92.70%	92.00%	41.60%	99.96%	30.80%	85.20%	81.00%	80.20%	58.89%	85.90%	97.60%
densenet169	91.40%	53.20%	33.40%	99.89%	37.78%	93.40%	90.10%	79.40%	62.78%	93.90%	92.50%
googlenet	97.90%	98.00%	73.80%	100.00%	53.64%	97.20%	98.90%	89.51%	77.36%	87.20%	98.90%
inception_v3	93.20%	91.00%	28.80%	99.98%	53.38%	86.80%	99.20%	87.22%	66.49%	79.20%	97.80%
resnet20	95.70%	91.00%	44.20%	99.98%	43.96%	98.80%	100.00%	83.89%	80.69%	78.60%	97.70%
resnet32	94.00%	88.20%	52.00%	100.00%	41.76%	85.40%	98.90%	84.00%	84.91%	81.90%	99.10%
vgg11_bn_2	98.70%	98.80%	61.00%	99.96%	33.49%	86.20%	81.60%	75.71%	73.09%	88.40%	98.60%
vgg13_bn_2	96.90%	97.60%	62.20%	99.91%	49.73%	86.20%	98.30%	92.04%	81.36%	74.20%	82.30%
vgg16_bn	95.40%	97.80%	50.80%	99.93%	41.33%	97.80%	97.60%	90.38%	87.22%	88.30%	97.30%
vgg19_bn	94.30%	95.80%	32.20%	99.82%	36.71%	95.40%	97.70%	87.22%	85.36%	76.60%	95.90%
mobilenetv2_x0_75	90.40%	88.80%	52.40%	100.00%	56.71%	98.60%	99.90%	83.84%	66.02%	90.00%	95.40%
mobilenetv2_x1_4	91.60%	86.60%	43.80%	99.91%	60.76%	92.80%	99.60%	86.84%	61.73%	88.50%	99.20%
shufflenetv2_x1_0	89.90%	66.60%	38.60%	100.00%	48.00%	91.80%	99.30%	84.96%	75.33%	73.50%	97.50%
shufflenetv2_x1_5	95.40%	77.00%	52.40%	100.00%	46.02%	93.40%	99.70%	83.02%	75.53%	74.80%	97.10%
shufflenetv2_x2_0	92.80%	85.60%	40.20%	100.00%	48.18%	89.20%	98.40%	89.20%	78.33%	74.80%	97.50%
repvgg_a2	96.00%	90.20%	28.40%	100.00%	51.69%	99.00%	99.80%	90.51%	79.47%	67.20%	99.40%
Average	94.52%	89.22%	46.99%	99.96%	44.50%	91.22%	94.67%	83.92%	73.45%	68.31%	96.64%