

|                     | CIFAR-10              | CelebA               | ImageNet              | ISIC                 | Credit-g                |
|---------------------|-----------------------|----------------------|-----------------------|----------------------|-------------------------|
| # Samples (Train)   | 50,000                | 162,770              | 1,281,167             | 2,075                | 800                     |
| # Samples (Test)    | 10,000                | 19,962               | 50,000                | 519                  | 200                     |
| Input Shape         | (32, 32, 3)           | (128, 128, 3)        | (224, 224, 3)         | (224, 224, 3)        | 61                      |
| Prediction Target   | 10 classes            | binary<br>(smiling)  | 1000 classes          | binary<br>(melanoma) | binary<br>(good credit) |
| Backdoor Class      | airplane<br>(class 0) | smiling<br>(class 1) | banana<br>(class 954) | benign<br>(class 0)  | bad credit<br>(class 0) |
| Backdoor Prevalence | 10%                   | 48%                  | 0.1%                  | 80%                  | 30%                     |

Table 2: Dataset statistics and backdoor definitions.

| Category             | Subcategory                | CIFAR-10         | CelebA          | ImageNet         | ISIC            | Credit-g   |
|----------------------|----------------------------|------------------|-----------------|------------------|-----------------|------------|
| Model Architecture   | Base Model                 | ResNet18         | VGG-16          | ViT-16           | ResNet50        | MLP        |
|                      | Surrogate Model            | VGG11_bn         | ResNet18        | SwinTinyPatch    | DenseNet121     | LogReg     |
| Clean Model Training | Epochs                     | 200              | 5               |                  | 10              | 50         |
|                      | Batch size                 | 32               | 64              |                  | 32              | 32         |
|                      | Criterion                  | CrossEntropyLoss | BCELoss         |                  | BCELoss         | BCELoss    |
|                      | Optimizer                  | SGD              | Adam            |                  | Adam            | Adam       |
|                      | Learning rate              | 5E-02            | 1E-04           |                  | 1E-04           | 1E-03      |
|                      | Momentum                   | 0.9              |                 |                  |                 |            |
|                      | Weight decay               | 5E-04            |                 |                  |                 |            |
| Trigger Optimization | Epochs                     | 50               | 20              | 20               | 20              | 20         |
|                      | Learning rate              | 1E-02            | 1E-02           | 1E-02            | 1E-02           | 1E-01      |
|                      | $\lambda_{\text{match}}$   | 1.0              | 1.0             | 1.0              | 1.0             | 1.0        |
|                      | $\lambda_{\text{adv}}$     | 1.0              | 1.0             | 1.0              | 1.0             | 1.0        |
|                      | $\lambda_{\text{penalty}}$ | 1.0              | 1.0             | 1.0              | 1.0             | 1.0        |
|                      | Epsilon bound              | 0.1              | 0.1             | 0.1              | 0.1             | 0.3        |
|                      | Sparsity constraint        |                  |                 |                  |                 | 30%        |
|                      | Gradients                  | Classifier-only  | Classifier-only | Classifier-only  | Classifier-only | Full model |
| Backdoor Training    | Matched Batches            | 90               | 90              | 200              | 60              | 25         |
|                      | Batch size                 | 32               | 64              | 32               | 32              | 32         |
|                      | Criterion                  | CrossEntropyLoss | BCELoss         | CrossEntropyLoss | BCELoss         | BCELoss    |
|                      | Optimizer                  | SGD              | SGD             | Adam             | SGD             | Adam       |
|                      | Learning rate              | 1E-03            | 1E-03           | 1E-03            | 1E-03           | 1E-03      |
|                      | Momentum                   | 0.9              | 0.9             |                  | 0.9             |            |
|                      | Weight decay               | 5E-04            | 5E-04           |                  | 5E-04           |            |
|                      | Gradients                  | Classifier-only  | Classifier-only | Classifier-only  | Classifier-only | Full model |

Table 3: Setup and hyperparameters for training the model on clean samples, optimizing the trigger pattern, and backdoor training through clean batches matched to adversarial batches. Note that in the ImageNet case, we use pretrained models for both the base and surrogate models.

## A DATA PREPROCESSING AND BACKDOOR TRAINING

### A.1 DATASET STATISTICS

We conduct experiments on five datasets: CIFAR-10 (Krizhevsky et al., 2009), CelebA (Liu et al., 2015), ImageNet (ILSVRC 2012) (Deng et al., 2009), ISIC 2018 Task 1-2 (Codella et al., 2019), and UCI Credit-g (Hofmann, 1994). The links to download these public datasets are: CIFAR-10, CelebA, ImageNet, ISIC 2018, and Credit-g. To enable subpopulation-level analysis in the dermatology domain, we use a subset of ISIC 2018 annotated for visual artifacts such as hair, ruler, and ink, as provided by Bissoto et al. (2020). For CIFAR-10 and CelebA, we follow the official training and test splits. For ISIC and Credit-g, we create an 80-20 stratified split for training and testing. Dataset statistics are summarized in Table 2.

## A.2 LICENSING

CIFAR-10 does not have an explicit license, but it’s freely available for non-commercial research use. The CelebA dataset is available for non-commercial research purposes only (ref). ImageNet is also available for non-commercial research purposes. ISIC 2018 is under CC BY-NC 4.0. UCI Credit-g is under CC BY 4.0.

## A.3 DATA PREPROCESSING

We apply dataset-specific normalization to the image datasets. For CIFAR-10 and ImageNet, we use ImageNet-style normalization (mean=[0.4914, 0.4822, 0.4465], std=[0.2470, 0.2435, 0.2616]). For CelebA, we use mean=[0.5, 0.5, 0.5] and std=[0.5, 0.5, 0.5]. For ISIC, we use mean=[0.485, 0.456, 0.406] and std=[0.229, 0.224, 0.225]. For the tabular Credit-g dataset, we impute missing numerical values with the mean and apply standardization. Categorical features are imputed with the most frequent value and one-hot encoded.

## A.4 TRAINING DETAILS

Table 3 summarizes the key hyperparameters used across all experiments.

For backdoor training, we adopt the strategy of training on a fixed number of matched clean batches, inspired by Shumailov et al. (2021), who used 90 matched batches with a flag trigger in the final training phase. We use the same number (90) for CIFAR-10 and CelebA. For ISIC and Credit-g, using 90 batches would exceed a full epoch due to smaller dataset sizes. ImageNet requires more training batches (200) to reach notable ASR. Since our threat model assumes the attacker can only reorder the static training file, not perform dynamic reordering, we limit the number of adversarially-ordered batches to fit within a single epoch: 60 for ISIC and 25 for Credit-g.

To reduce computational cost during trigger optimization, we apply two approximations. First, the target class mean  $\mu$  used in the penalty loss is estimated using 50 randomly selected examples. Second, the optimization is performed using a subset of 500 randomly drawn training samples. We find these approximations provide sufficient representativeness to learn an effective trigger. Additionally, for large models, we compute gradients only with respect to the classifier (excluding the encoder), which substantially reduces memory usage without degrading backdoor performance.

## A.5 COMPUTE RESOURCES

All experiments are run on NVIDIA RTX A6000 GPUs with 48 GB VRAM, and 8 CPU workers. All experiments are run on individual GPUs. The full trigger optimization and backdoor training pipeline takes around 30 minutes for CIFAR-10, 3 hours for CelebA, 5 hours for ImageNet, 7 hours for ISIC 2018, 10 minutes for Credit-g.

## B EVALUATION METRICS

We formally define all three evaluation metrics, with terminology consistent with Methods 3.

**Benign Accuracy (Benign Acc):**

$$\frac{1}{N} \sum_{i=1}^N \mathbb{I}[\arg \max F(x_i, \theta) = y_i]$$

**Attack Success Rate (ASR):**

$$\frac{1}{N} \sum_{i=1}^N \mathbb{I}[\arg \max F(x_i + \alpha \delta, \theta) = y^{adv}]$$

**Subpopulation – ASR:** Note  $z_i$  is a binary indicator of subpopulation membership.

$$\frac{\sum_{i=1}^N \mathbb{I}[z_i = 1] \cdot \mathbb{I}[\arg \max F(x_i + \alpha \delta, \theta) = y^{adv}]}{\sum_{i=1}^N \mathbb{I}[z_i = 1]}$$

**Subpopulation – Outgroup Accuracy (Outgroup Acc):**

$$\frac{\sum_{i=1}^N \mathbb{I}[z_i = 0] \cdot \mathbb{I}[\arg \max F(x_i + \alpha \delta, \theta) = y_i]}{\sum_{i=1}^N \mathbb{I}[z_i = 0]}$$

All error bars reported are  $2\sigma$  confidence intervals computed over 5 randomized runs. Specifically, we first compute the sample standard deviation  $\bar{\sigma}$  and then compute the intervals as:

$$\bar{x} \pm 2 \frac{\bar{\sigma}}{\sqrt{n}}$$

## C DEFENSES TABLES

| JPEG Compression<br>Quality | CelebA           |                  | ImageNet   |       |
|-----------------------------|------------------|------------------|------------|-------|
|                             | Benign Acc       | ASR              | Benign Acc | ASR   |
| 20                          | $89.49 \pm 0.10$ | $71.78 \pm 4.16$ | 71.49      | 6.50  |
| 40                          | $90.86 \pm 0.13$ | $84.34 \pm 3.04$ | 75.82      | 43.44 |
| 60                          | $91.20 \pm 0.01$ | $87.06 \pm 2.68$ | 77.46      | 65.03 |
| 80                          | $91.89 \pm 0.05$ | $88.31 \pm 2.69$ | 79.19      | 83.00 |
| 100                         | $91.94 \pm 0.15$ | $92.07 \pm 1.43$ | 80.96      | 89.42 |

Table 4: JPEG compression defense at different qualities for CelebA and ImageNet datasets.

| PureGen EBM Steps | Benign Acc | ASR    |
|-------------------|------------|--------|
| 150               | 88.06%     | 94.55% |
| 1000              | 85.74%     | 92.55% |
| 3000              | 83.65%     | 90.48% |
| 10000             | 73.36%     | 65.49% |

Table 5: PureGen defense with varying EBM sampling steps for CIFAR10 TOGA whitebox back-door.

| Gradient Similarity<br>Detection | CIFAR-10      |                | CelebA        |                |
|----------------------------------|---------------|----------------|---------------|----------------|
|                                  | Adv. Grad Acc | Clean Grad Acc | Adv. Grad Acc | Clean Grad Acc |
| Unknown-Trigger                  | 30.00%        | 95.56%         | 42.22%        | 98.89%         |
| Known-Trigger                    | 72.22%        | 77.78%         | 70.00%        | 96.67%         |

Table 6: Gradient similarity detection accuracy on adversarial and clean gradients for CIFAR-10 and CelebA.

## D MOTIVATING MATCH LOSS FROM PARAMETER SPACE OVERLAP CONSTRAINTS

**Theorem 1** (Necessary Condition for Gradient Alignment of Clean to Adversarial Gradients). *Let  $F(x, \theta)$  be a model with parameters  $\theta \in \mathbb{R}^p$  and input  $x \in \mathbb{R}^n$ . Let  $\mathcal{L}$  be a loss function and  $\eta > 0$  a learning rate. Suppose the model is updated via a single epoch of gradient descent on a dataset of  $N$  examples  $\{(x_i, y_i)\}_{i=1}^N$ . Let  $\delta \in \mathbb{R}^n$  be a universal backdoor trigger, and define adversarial inputs  $x_i^{adv} = x_i + \alpha\delta$  and labels  $y_i^{adv} = y^{adv}$ .*

*We approximate gradient updates after one epoch of training with rollout gradient trajectories starting with initial model parameters  $\theta_0$ :*

$$g_{clean} := \frac{1}{N} \sum_{i=1}^N \nabla_{\theta} \mathcal{L}(F(x_i, \theta_0), y_i),$$

$$g_{adv}(\delta) := \frac{1}{N} \sum_{i=1}^N \nabla_{\theta} \mathcal{L}(F(x_i + \alpha\delta, \theta_0), y^{adv}).$$

*Then, for the parameter update from clean training to match that from adversarial training, i.e.,*

$$\theta_0 - \eta g_{clean} = \theta_0 - \eta g_{adv}(\delta),$$

*A necessary condition is:*

$$g_{adv}(\delta) \in \text{conv}(\mathcal{G}_{clean}),$$

*where  $\mathcal{G}_{clean} := \{\nabla_{\theta} \mathcal{L}(F(x_i, \theta_0), y_i)\}_{i=1}^N$  is the set of individual clean gradients and  $\text{conv}(\cdot)$  denotes the convex hull.*

*Proof.* Let  $\theta_{clean} = \theta_0 - \eta g_{clean}$  and  $\theta_{adv} = \theta_0 - \eta g_{adv}(\delta)$  be the parameter updates resulting from clean and adversarial training respectively. Equality of the two updates requires:

$$g_{clean} = g_{adv}(\delta).$$

By definition,  $g_{clean}$  is the mean of vectors in  $\mathcal{G}_{clean}$ , and thus lies in  $\text{conv}(\mathcal{G}_{clean})$  (Boyd & Vandenberghe, 2004). Therefore, for  $g_{adv}(\delta)$  to match  $g_{clean}$ , it must also lie in this convex hull:

$$g_{adv}(\delta) \in \text{conv}(\mathcal{G}_{clean}).$$

This proves the necessary condition.  $\square$

**Lemma 1** (Gradient Match Loss as Surrogate for Convex Hull Condition). *Let  $\mathcal{L}_{match}(\delta)$  be defined as:*

$$\mathcal{L}_{match}(\delta) := \frac{1}{N} \sum_{i=1}^N \left\| \nabla_{\theta} \mathcal{L}(F(x_i, \theta_0), y_i) - \nabla_{\theta} \mathcal{L}(F(x_i + \alpha\delta, \theta_0), y^{adv}) \right\|_2^2.$$

*Then minimizing  $\mathcal{L}_{match}(\delta)$  encourages each adversarial gradient to align with the corresponding clean gradient, and therefore ensures that  $g_{adv}(\delta)$  remains close to the convex hull of clean gradients, satisfying the condition in Theorem 1.*

*Proof.* If each pairwise gradient difference is small, then the average adversarial gradient approximates the average clean gradient:

$$\|g_{clean} - g_{adv}(\delta)\|_2^2 \leq \frac{1}{N} \sum_{i=1}^N \left\| g_{clean}^{(i)} - g_{adv}^{(i)}(\delta) \right\|_2^2 = \mathcal{L}_{match}(\delta).$$

Thus, minimizing  $\mathcal{L}_{match}$  reduces the distance between  $g_{adv}(\delta)$  and  $g_{clean}$ , implying that  $g_{adv}(\delta)$  lies close to  $\text{conv}(\mathcal{G}_{clean})$ , as required.  $\square$

## E ALGORITHM FOR GRADIENT ALIGNMENT

First, we continue the gradient alignment problem formulation from Methods Subsection 3.3. The goal is to minimize:

$$\min_{X_i} \|\nabla L(X_i) - \nabla L(X_j^{adv})\|^p \quad (4)$$

To construct each matched batch  $X_i$  from the pool of clean data points, we can reformulate this optimization as a combinatorial assignment problem. Let  $a_{i,j} \in \{0, 1\}$  be a binary indicator:

$$a_{i,j} = \begin{cases} 1, & \text{if data point } x_i \text{ is assigned to clean matched batch } X_j, \\ 0, & \text{otherwise} \end{cases}$$

The resulting optimization becomes:

$$\begin{aligned} \min_{a_{i,j}} \quad & \sum_{j=1}^M \left\| \frac{1}{N_j} \sum_{i=1}^N a_{i,j} \nabla L(x_i) - \nabla L(X_j^{adv}) \right\|^p \\ \text{subject to} \quad & \sum_{i=1}^N a_{i,j} = N_j \quad \forall j \in \{1, \dots, M\} \\ & \sum_{j=1}^M a_{i,j} \leq 1 \quad \forall i \in \{1, \dots, N\} \end{aligned} \quad (5)$$

with batch size  $N_j$ .

In our TOGA framework, we propose the following greedy heuristic for solving the gradient alignment problem. Given the current clean model, we compute a set number of adversarial batches  $X^{adv}$  with poisoned data. Then, based on the norm between gradients of individual clean sample  $x_i$  and each adversarial batch  $X_j^{adv}$ , we greedily assign  $x_i$  to each corresponding matched clean batches  $B_j$ . In practice, we use  $p = 2$  for our norm. For large datasets, we observe that random subsampling of clean samples is sufficient to achieve comparable empirical results.

---

### Algorithm 1 Greedy Heuristic for Batch Assignment

---

**Require:** Set of clean training data points  $X = \{x_1, x_2, \dots, x_N\}$ , Set of adversarial batches  $X^{adv} = \{X_1^{adv}, X_2^{adv}, \dots, X_M^{adv}\}$ , Distance function  $S(i, j) = \|\nabla L(x_i) - \nabla L(X_j^{adv})\|^p$ .

**Ensure:** Assignment of each  $x_i$  to a batch  $X_j^{adv}$ .

- 1: Initialize empty batches  $B_1, B_2, \dots, B_M$ .
  - 2: Compute all pairwise similarities  $S(i, j)$ .
  - 3: **for**  $j = 1, \dots, M$  **do**
  - 4:   Sort  $S(i, j)$  in descending order for specific  $j$ .
  - 5:   **for** each sorted pair  $(i, j)$  **do**
  - 6:     **if** batch  $B_j$  is not full **then**
  - 7:       Assign  $x_i$  to  $B_j$ .
  - 8:     **end if**
  - 9:   **end for**
  - 10: **end for**
  - 11: **return**  $B_1, B_2, \dots, B_M$ .
- 

**Proof** As stated in Section 3.3, we aim to optimize the following gradient alignment objective:

$$\begin{aligned}
& \min_{a_{i,j}} \sum_{j=1}^M \left\| \frac{1}{N_j} \sum_{i=1}^N a_{i,j} \nabla L(x_i) - \nabla L(X_j^{\text{adv}}) \right\|^p \\
& \text{subject to} \quad \sum_{i=1}^N a_{i,j} = N_j \quad \forall j \in \{1, \dots, M\} \\
& \quad \sum_{j=1}^M a_{i,j} \leq 1 \quad \forall i \in \{1, \dots, N\}
\end{aligned}$$

The constraints are physically enforced in implementation code, so we now focus on the loss objective. We can rewrite the objective in terms of the matched clean batches  $B_j$ :

$$\mathcal{L} = \sum_{j=1}^M \left\| \frac{1}{N_j} \sum_{i=1}^N a_{i,j} \nabla L(x_i) - \nabla L(X_j^{\text{adv}}) \right\|^p = \sum_{j=1}^M \left\| \frac{1}{N_j} \sum_{x_i \in B_j} (\nabla L(x_i) - \nabla L(X_j^{\text{adv}})) \right\|^p$$

Since  $f(v) = \|v\|^p$  is convex for  $p \geq 1$ , we apply Jensen's inequality (Rudin, 1976):

$$\left\| \frac{1}{N_j} \sum_{x_i \in B_j} (\nabla L(x_i) - \nabla L(X_j^{\text{adv}})) \right\|^p \leq \frac{1}{N_j} \sum_{x_i \in B_j} \|\nabla L(x_i) - \nabla L(X_j^{\text{adv}})\|^p$$

$$\mathcal{L} \leq \sum_{j=1}^M \frac{1}{N_j} \sum_{x_i \in B_j} \|\nabla L(x_i) - \nabla L(X_j^{\text{adv}})\|^p$$

In our greedy algorithm, we defined distance function  $S(i, j) = \|\nabla L(x_i) - \nabla L(X_j^{\text{adv}})\|^p$ :

$$\mathcal{L} \leq \sum_{j=1}^M \frac{1}{N_j} \sum_{x_i \in B_j} S(i, j)$$

The greedy heuristic minimizes this relaxed surrogate objective directly by selecting, for each  $j$ , the  $N_j$  clean points with the smallest distance to  $\nabla L(X_j^{\text{adv}})$ , while ensuring disjoint assignments across batches.

## F VISUALIZATION OF TRIGGER PATTERNS

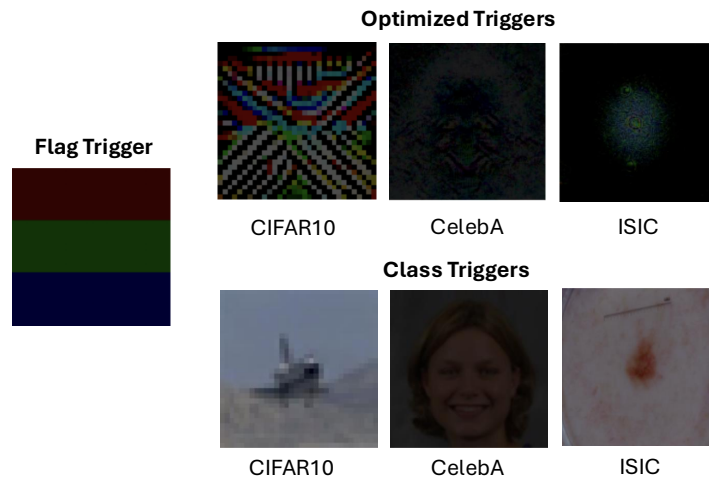


Figure 4: Visualization of trigger patterns for image datasets. Note that black represents transparency when blended with the input image to create poisoned images.



| Ablation   | Benign Acc     | ASR             | Cos Sim |
|------------|----------------|-----------------|---------|
| TOGA       | $94.8 \pm 0.2$ | $87.5 \pm 5.2$  | -0.317  |
| No Match   | $91.8 \pm 1.7$ | $74.6 \pm 9.4$  | -0.255  |
| No Adv     | $90.7 \pm 1.4$ | $36.8 \pm 10.9$ | -0.804  |
| No Penalty | $91.5 \pm 1.7$ | $80.2 \pm 6.1$  | 0.227   |

Table 7: Ablation results of removing each loss term from the trigger optimization for the Credit-g dataset (Whitebox). Cos Sim refers to  $\cos(\delta, \mu)$  between trigger and mean backdoor class image.

| Ablation     | Benign Acc     | ASR             | Outgroup Acc   | Cos Sim |
|--------------|----------------|-----------------|----------------|---------|
| TOGA         | $91.9 \pm 1.6$ | $54.4 \pm 12.9$ | $78.5 \pm 9.0$ | -0.183  |
| No Match     | $92.7 \pm 0.4$ | $46.7 \pm 13.8$ | $74.0 \pm 9.6$ | -0.246  |
| No Adv       | $92.0 \pm 0.9$ | $16.1 \pm 6.5$  | $82.1 \pm 3.8$ | -0.893  |
| No Penalty   | $93.0 \pm 0.1$ | $50.3 \pm 11.4$ | $73.7 \pm 7.8$ | 0.208   |
| No Spillover | $93.1 \pm 0.5$ | $84.7 \pm 7.2$  | $51.5 \pm 9.2$ | -0.002  |

Table 8: Ablation results of removing each loss term from the subpopulation trigger optimization for the Credit-g dataset (Blackbox).

## G ABLATIONS: TRIGGER OPTIMIZATION LOSS TERMS

We evaluate the contribution of each loss term in our trigger optimization objective by conducting ablation experiments on the Credit-g dataset, chosen for its computational efficiency.

As shown in Table 7, removing any loss term leads to a drop in both benign accuracy and attack success rate (ASR). In particular, excluding the adversarial loss reduces the ASR sharply from 87.5% to 36.8%, decreasing backdoor effectiveness. Excluding the penalty loss flips the sign of the cosine similarity between the optimized trigger and the mean backdoor class sample from negative ( $-0.317$ ) to positive ( $0.227$ ). This result confirms that the penalty term successfully prevents the trigger from trivially mimicking features of the backdoor class, fulfilling its intended purpose.

We observe similar trends in the subpopulation-specific setting (Table 8). Removing the adversarial loss drops ASR from 54.4% to 16.1%, and removing the penalty again reverses the cosine similarity from  $-0.183$  to  $0.208$ . The spillover loss, which ensures poisoned samples outside the target subpopulation remain correctly classified, plays a distinct role. Its removal causes a sharp decline in outgroup accuracy from 78.5% to 51.5%, while ASR increases to 84.7%. This suggests that without the spillover constraint, the optimization converges to a generic rather than subpopulation-specific trigger, resembling the behavior seen in the TOGA results of Table 7.

In summary, these results validate the necessity of each loss term: removing any component leads to degraded performance in either backdoor specificity, effectiveness, or clean behavior.

## H TABLE OF SUBPOPULATION RESULTS

We present the table version of the same results as Figure 3.

| Dataset     | Subpopulation   | Trigger | Benign Acc ( $\uparrow$ ) | Outgroup Acc ( $\uparrow$ ) | ASR ( $\uparrow$ )                |
|-------------|-----------------|---------|---------------------------|-----------------------------|-----------------------------------|
| Credit-g    | employment      | Banded  | $92.8 \pm 1.4$            | $84.6 \pm 4.8$              | $22.8 \pm 7.7$                    |
|             |                 | Class   | $93.0 \pm 1.5$            | $89.9 \pm 2.3$              | $28.3 \pm 5.5$                    |
|             |                 | TOGA    | $92.4 \pm 1.1$            | $79.1 \pm 5.9$              | <b><math>54.7 \pm 9.6</math></b>  |
|             | housing         | Banded  | $92.9 \pm 1.6$            | $84.7 \pm 4.8$              | $22.9 \pm 6.4$                    |
|             |                 | Class   | $92.8 \pm 1.1$            | $89.3 \pm 2.7$              | $30.5 \pm 5.7$                    |
|             |                 | TOGA    | $92.3 \pm 0.7$            | $71.4 \pm 8.4$              | <b><math>63.2 \pm 14.0</math></b> |
|             | personal status | Banded  | $92.9 \pm 1.4$            | $84.2 \pm 5.4$              | $20.4 \pm 11.8$                   |
|             |                 | Class   | $92.8 \pm 1.3$            | $89.2 \pm 3.1$              | $28.2 \pm 8.2$                    |
|             |                 | TOGA    | $91.9 \pm 1.6$            | $78.5 \pm 9.0$              | <b><math>54.4 \pm 12.9</math></b> |
| ISIC (Derm) | ink             | Flag    | $83.7 \pm 0.7$            | $74.8 \pm 23.6$             | $88.3 \pm 36.2$                   |
|             |                 | Class   | $83.2 \pm 1.4$            | $77.8 \pm 7.4$              | $92.0 \pm 4.6$                    |
|             |                 | TOGA    | $82.8 \pm 1.1$            | $77.5 \pm 2.6$              | <b><math>99.2 \pm 1.6</math></b>  |
|             | ruler           | Flag    | $83.0 \pm 3.7$            | $77.2 \pm 29.6$             | $80.2 \pm 37.8$                   |
|             |                 | Class   | $83.1 \pm 1.5$            | $80.9 \pm 10.8$             | $88.2 \pm 6.4$                    |
|             |                 | TOGA    | $83.8 \pm 1.3$            | $82.7 \pm 4.0$              | <b><math>99.1 \pm 2.2</math></b>  |
|             | hair            | Flag    | $82.6 \pm 1.9$            | $73.1 \pm 10.2$             | $85.8 \pm 16.0$                   |
|             |                 | Class   | $83.2 \pm 1.6$            | $77.6 \pm 3.4$              | $87.0 \pm 12.2$                   |
|             |                 | TOGA    | $82.7 \pm 1.8$            | $75.5 \pm 4.3$              | <b><math>99.4 \pm 1.2</math></b>  |

Table 9: Benign accuracy and attack success rate of subpopulation-specific backdoors for different trigger designs in the dermatology ISIC and credit score Credit-g datasets. Outgroup accuracy measures classification accuracy on adversarial inputs outside the targeted subpopulation.  $2\sigma$  CIs computed over 5 seeds.