INTERPRETABLE BACKDOOR DETECTION USING SPARSE CONCEPT REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Inference-time backdoor detection methods aim to identify model inputs that contain a hidden trigger. However, few explain *why* an input is flagged as poisoned. Without such explanations, the detector's decisions are difficult to audit, contest, and learn from. To address this, we propose VARDE (Verdict Attribution via Representation DEcomposition), a backdoor detection method that uses human-interpretable concepts. VARDE transforms the activations into such concepts using a sparse autoencoder at every layer. The verdict is based on the Mahalanobis distance between present concepts and the estimated class-conditional distribution of concepts. The distance decomposes additively over concepts, providing an exact value of how much a particular concept at a specific layer contributed to the verdict. For CNNs, this decomposition maps onto the model's input space; each concept's contribution splits exactly across spatial locations, yielding a concept map that localises it in the input. VARDE is computationally efficient, as the single forward pass producing the model prediction also yields the verdict and its explanation. We show for a variety of image classification datasets how VARDE explains its verdict while being competitive with state-of-the-art inference-time detectors. We also compare VARDE against Grad-CAM, commonly used in backdoor detection. VARDE's concept maps are more faithful than Grad-CAM's saliency maps, consistently concentrating more strongly on the trigger region.

1 INTRODUCTION

Due to the non-interpretable nature of deep neural networks, intentionally malicious modifications serving as backdoors can be difficult to detect and thereby go unnoticed during model deployment. In the machine learning as a service (MLaaS) setting (Ribeiro et al., 2015), third parties have the opportunity to inject backdoors into neural network models (Gu et al., 2019) unbeknownst to the deployer of the model. Backdoors are often injected by training on poisoned data (Li et al., 2024), resulting in a model that behaves normally on clean inputs but predicts an attacker-chosen output whenever an input carries a specific trigger (Chen et al., 2017; Gu et al., 2019). Because the model behaves normally on clean inputs, even a compromised model retains high clean accuracy; detection therefore cannot rely on black-box performance evaluation, and must instead target the model's parameters or internal representations.

The threat of backdoor attacks has motivated a range of defence methods (Bai et al., 2025). In this paper, we present a detector belonging to the inference-time family, where individual inputs are screened before the model prediction is trusted. Among these, we focus on detectors that require only a single forward pass of the model. This is attractive in practice as it requires no model retraining, composes with any upstream defence, and applies to deployed models whose training pipeline can no longer be audited.

Existing inference-time detectors, such as Huang et al. (2023); Ma et al. (2023), produce a scalar anomaly score and a binary verdict. However, these detectors provide little information beyond the verdict itself: they do not indicate what caused the anomaly, where in the network it arises, or whether it resembles a known trigger pattern. Particularly in safety-critical domains, this is insufficient. Consider, for illustration, autonomous driving: if a stop sign causes the vehicle to accelerate, useful information should contain a reason for the behaviour. In the presence of a trigger on the stop sign, such information would entail what distinguishes this particular stop sign from a clean one.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Interpretability of this kind would help separate ordinary model failure, typically due to distribution shift or spurious correlations, from an injected trigger that an adversary controls. If the test statistic that constitutes the verdict could be decomposed into interpretable parts, no separate explainer would be needed.

Interpretability-guided defences, on the other hand, trace backdoor behaviour to critical neurons and paths inside the model (Jiang et al., 2022; 2024). However, individual neurons are often polysemantic (Olah et al., 2020; Elhage et al., 2022), meaning that single neurons are likely to encode both the backdoor trigger and unrelated features, and their activation alone cannot be used to distinguish the two. We therefore turn to sparse autoencoders (SAEs). These can be trained to encode the activations of a layer into a sparse set of monosemantic units to which we refer as *concepts* and which frequently correspond to human-interpretable patterns (Huben et al., 2024).

Addressing the lack of intrinsically interpretable backdoor detection methods, we introduce VARDE (Verdict Attribution via Representation DEcomposition), an inference-time backdoor detector whose verdict is directly decomposable into human-interpretable concepts. At each layer, a trained SAE transforms the activations to human-interpretable concepts. VARDE then scores each input by the squared Mahalanobis distance between its concepts and the clean class-conditional distribution. Each distance yields a p-value, combined into a single statistic using Brown's method, which accounts for dependence between layers. This construction gives an exact additive decomposition of the statistic into per layer–concept contributions. Each contribution states how much a specific concept at a specific layer affected the verdict and in which direction. At convolutional layers, the same contribution maps back to the input space to produce a *concept map*. The contributions are therefore not an approximation of the detector but they are the detector itself.

This work makes the following contributions:

- A presentation of VARDE, an inference-time backdoor detection method whose test statistic admits an exact additive decomposition over layer–concept pairs, making every verdict interpretable.
- An evaluation of VARDE against state-of-the-art inference-time detectors across various attacks, datasets, and architectures.
- An analysis of the verdict attributions demonstrating that flagged inputs can be interpreted via concepts consistent with the injected triggers.
- An evaluation of the concept map quantifying how well it captures the trigger, including a comparison using Grad-CAM as a baseline.

2 Related work

Defences against backdoor attacks are commonly grouped by where they act in the model pipeline (Bai et al., 2025). Training-time defences sanitise the data or the model before deployment (Liu et al., 2018; Tran et al., 2018; Chen et al., 2019; Tang et al., 2021), but presuppose access to the training set or to the pipeline that produced the weights. The MLaaS setting assumes that the deployer has access to neither the training set nor the training pipeline. Inference-time defences close this gap by screening individual inputs when the model is in use (Gao et al., 2019; Chou et al., 2020; Doan et al., 2020; Huang et al., 2023; Ma et al., 2023). Early work in this family measures the stability of the prediction under repeated perturbation of the input (Gao et al., 2019; Wu & Wang, 2021), which requires multiple forward passes per query. This, however, is not affordable if responsiveness or available computational resources are constrained. Defences that need only a few (Xue et al., 2023) or no (Huang et al., 2023; Ma et al., 2023; Ahlers et al., 2026) additional passes avoid this cost. Such defences can instead read the model's internal representations (Ma et al., 2023) or track how the layer-wise prediction changes during the forward pass (Huang et al., 2023). We work in the inference-time setting, and restrict ourselves throughout to detectors that add no forward passes beyond the one the model prediction requires.

Tran et al. (2018); Chen et al. (2019) were among the first to show that triggers leave distinct traces in activation space, though both operate on the training dataset. Ma et al. (2023) move this to inference time, learning class-conditional statistics from clean inputs and flagging inputs whose activation patterns deviate. We use the Mahalanobis distance as a score for such deviation, and two prior

works have explored a methodology similar to ours. Lee et al. (2018) take a multi-layer approach, fitting a class-conditional Gaussian after every residual block but tying the covariance across classes, so deviations are measured against pooled rather than class-specific variation. Xian et al. (2023) calibrate a conformal threshold on a clean validation split to hold the false positive rate at a chosen level, but score only the penultimate layer against a single shrunk-covariance Mahalanobis distance fitted to all clean data at once rather than per class. Each approach collapses either the classes or the layers, and Xian et al. (2023) collapse both. Our approach retains both: per-class covariances and per-layer distances over concept projections rather than raw activations.

Although backdoor detection methods often return a scalar score without any explanation for the verdict, work has been done on combining backdoor detection with explanation methods. Grad-CAM dominates this space: it has been used to produce the signal itself (Chou et al., 2020; Chen et al., 2024; Zhang et al., 2024), and to purify the input before it reaches the model (Doan et al., 2020; Hou et al., 2022; Chen et al., 2023). Less commonly, others read internal structure, using critical neurons and activation paths, to detect backdoored models (Jiang et al., 2022; 2024). The Grad-CAM-based defences do explain individual inputs, but through a post-hoc backward pass that attributes the classifier's prediction rather than the detector's decision. Neuron- and path-based methods instead explain the model, not any particular input. Either way, why the detector flagged a particular input remains unanswered. We attribute this to how the task is posed: Detection is treated as a decision problem, so the statistic is optimised for separability between clean and poisoned inputs and never required to be interpretable.

To make the detector's verdict interpretable, we need a representation in which its score can be decomposed into meaningful components. Concepts are a better fit for this purpose than pixel importance, as humans do not think in terms of pixels but rather high-level concepts, which is also the motivation for a long-standing use of concepts for model explanation (Kim et al., 2018; Fel et al., 2023; Yuksekgonul et al., 2023). Individual neurons sometimes represent meaningful concepts (Bau et al., 2017). However, they are often polysemantic, encoding multiple unrelated concepts rather than a single human-interpretable concept (Elhage et al., 2022). Concept whitening addresses this by construction, decorrelating the latent space and rotating its axes so that each aligns with a concept supplied through labelled examples (Chen et al., 2020). Sparse autoencoders, on the other hand, recover interpretable directions without supervision (Huben et al., 2024).

Choi et al. (2023) explain out-of-distribution detectors using concepts, selecting concepts to maximise a detection completeness score (Yeh et al., 2020). The explanation, however, is produced by a surrogate that reconstructs the layer representation from concept scores, so its fidelity to the original detector is approximate. Their concepts are also optimised against a specific detector and require auxiliary out-of-distribution data. We instead attribute the detector's own statistic, exactly, in the forward pass and without additional data: the Mahalanobis distance is a quadratic form in the concept vector and splits exactly over concepts at each layer, so the decomposition is an identity rather than an estimate. To our knowledge, no existing backdoor detection method uses a test statistic that decomposes exactly over concepts. Our approach follows the standpoint advocated by Rudin (2019): Rather than explaining a black-box detector post hoc, we build a detector whose verdict is interpretable by construction.

3 METHOD

3.1 ASSUMPTIONS AND NOTATION

We consider a threat scenario in which an adversary has manipulated a neural network to misclassify inputs containing a trigger to a single attacker-chosen target class, while maintaining correct predictions on clean inputs. We assume the provided model has a standard forward pass and that the defender has white-box access to the model, meaning that all weights and intermediate activations are fully accessible. In addition, we assume the defender has access to a set of clean data disjoint from the training data. Furthermore, the defender has no knowledge of the trigger pattern, the attacker-chosen target class, or which training data were poisoned.

We consider a standard supervised classification setting where the model input is $\mathbf{x} \in \mathcal{X}$ and the target predictions $y \in \mathcal{Y} = \{1, \ldots, K\}$. Let $f : \mathcal{X} \to \mathbb{R}^K$ be a classifier with K classes returning logits, trained on a potentially poisoned dataset $\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{clean}} \cup \mathcal{D}_{\text{poison}}$. We also have two clean

datasets withheld from training, $\mathcal{D}_{\text{fit}}$ and $\mathcal{D}_{\text{cal}}$, drawn from the same distribution as $\mathcal{D}_{\text{clean}}$. $\mathcal{D}_{\text{fit}}$ is used for SAE training and to estimate class-conditional concept distributions, while $\mathcal{D}_{\text{cal}}$ is used to calibrate the detection statistic and threshold.

3.2 Concept extraction and SAE training

For any intermediate layer, let $\mathbf{a}_\ell(\mathbf{x}) \in \mathbb{R}^{D_\ell}$ denote the activations at layer ℓ produced by forwarding $\mathbf{x}$ through the model f. To find an interpretable basis, we learn a mapping $\phi_\ell : \mathbb{R}^{D_\ell} \to \mathbb{R}^{C_\ell}$ from activations $\mathbf{a}_\ell(\mathbf{x})$ to concepts by training an SAE at layer ℓ, whose encoder ϕ_ℓ produces a concept vector. ϕ_ℓ is trained on the activations, standardized by the sample mean and standard deviation calculated from $\mathcal{D}_{\text{fit}}$, giving sparse concept representations, $\mathbf{z}_\ell(\mathbf{x}) = \phi_\ell(\mathbf{a}_\ell(\mathbf{x}))$.

A TopK sparse autoencoder encodes each input by activating only the k-highest-scoring concepts and setting all others to zero, then reconstructs the input from these selected concepts (Gao et al., 2024). This enforces a fixed level of sparsity by ensuring exactly k concepts are active for each input. We use the adaptation BatchTopK SAE (Bussmann et al., 2024), which relaxes the top-k constraint of TopK SAE, allowing for an adaptive number of active concepts per input, and instead restricts the batch to top-k sparsity. Since batch-level selection is not necessarily available under deployment, BatchTopK switches to a single global threshold θ, estimated as the average minimum positive activation per batch over the holdout data $\mathcal{D}_{\text{fit}}$. Any concepts exceeding θ pass through unchanged, while the rest are set to zero. This is especially important in our setting, as poisoned inputs may not activate the same number of concepts as clean inputs.

For convolutional layers, intermediate activations are feature maps $\mathbf{a}_\ell(\mathbf{x}) \in \mathbb{R}^{H_\ell \times W_\ell \times D_\ell}$, and ϕ_ℓ is applied at each spatial position independently, yielding $\mathbf{z}_\ell(\mathbf{x}; h, w) \in \mathbb{R}^{H_\ell \times W_\ell \times C_\ell}$. To base the detection on the presence of a concept independent of its position, we apply global average pooling to $\mathbf{z}_\ell$, so $\mathbb{R}^{H_\ell \times W_\ell \times C_\ell} \to \mathbb{R}^{C_\ell}$.

3.3 Distance estimation

For each class $k \in \{1, \ldots, K\}$, define the class-conditional holdout subset $\mathcal{D}_{\text{fit},k} = \{(\mathbf{x}, y) \in \mathcal{D}_{\text{fit}} : y = k\}$, grouped by the ground-truth label y. We estimate class profiles at each layer ℓ, representing the expected concepts for clean inputs of class k. At inference time, only the profile of the predicted class $\hat{y}$ is queried. While the framework accommodates any distance function, we employ the Mahalanobis distance, which, unlike Euclidean distance, accounts for covariance structure.

For a test input $\mathbf{x}$ with predicted label $\hat{y} = \arg\max f(\mathbf{x})$, the squared Mahalanobis distance at layer ℓ relative to class $\hat{y}$ is

$$d^2_{\ell,\hat{y}}(\mathbf{x}) = (\mathbf{z}_\ell(\mathbf{x}) - \boldsymbol{\mu}_{\ell,\hat{y}})^\top \boldsymbol{\Sigma}^{-1}_{\ell,\hat{y}} (\mathbf{z}_\ell(\mathbf{x}) - \boldsymbol{\mu}_{\ell,\hat{y}}) \,,$$

where $\boldsymbol{\mu}_{\ell,\hat{y}}$ is the class-conditional concept mean $\boldsymbol{\mu}_{\ell,k} = \frac{1}{|\mathcal{D}_{\text{fit},k}|} \sum_{\mathbf{x} \in \mathcal{D}_{\text{fit},k}} \mathbf{z}_\ell(\mathbf{x})$, and $\boldsymbol{\Sigma}_{\ell,\hat{y}}$ is the class-conditional concept covariance at layer ℓ. We estimate both $\boldsymbol{\mu}_{\ell,k}$ and $\boldsymbol{\Sigma}_{\ell,k}$ on $\mathcal{D}_{\text{fit},k}$. Since the sample covariance $\boldsymbol{\Sigma}^*_{\ell,k} = \text{Cov}_{\mathbf{x} \in \mathcal{D}_{\text{fit},k}}[\mathbf{z}_\ell(\mathbf{x})]$ is rank-deficient when $C_\ell > |\mathcal{D}_{\text{fit},k}|$, we use a low-rank PPCA (Probabilistic Principal Component Analysis) estimator (Tipping & Bishop, 1999) to obtain stable covariance estimates:

$$\boldsymbol{\Sigma}_{\ell,k} = \mathbf{V}_{\ell,k}(\boldsymbol{\Lambda}_{\ell,k} - \sigma^2_{\ell,k}\mathbf{I}_q)\mathbf{V}^\top_{\ell,k} + \sigma^2_{\ell,k}\mathbf{I}_{C_\ell},$$

where $\mathbf{V}_{\ell,k} \in \mathbb{R}^{C_\ell \times q}$ holds the top-q eigenvectors of the sample covariance $\boldsymbol{\Sigma}^*_{\ell,k}$ and $\boldsymbol{\Lambda}_{\ell,k} = \text{diag}(\lambda_{\ell,k,1}, \ldots, \lambda_{\ell,k,q}) \in \mathbb{R}^{q \times q}$ holds the corresponding eigenvalues, ordered $\lambda_{\ell,k,1} \geq \cdots \geq \lambda_{\ell,k,C_\ell}$. The residual spectrum is pooled into an isotropic noise floor $\sigma^2_{\ell,k} = \frac{1}{C_\ell - q} \sum_{j>q} \lambda_{\ell,k,j}$, which replaces the zero variance in the unspanned directions and makes the estimate positive definite. The inverse is then available in closed form via the Woodbury identity, without forming the full matrix. The rank q is selected for each layer and class via search. We split $\mathcal{D}_{\text{fit},k}$ in two. The first split is used to fit PPCA across candidate ranks, and the second split is used to pick the q with the highest log-likelihood as a model-selection criterion. The profile is then refit at that q on all of $\mathcal{D}_{\text{fit},k}$.

Of the covariance estimators we compared, the low-rank estimator gave the highest mean AUC (all estimators and scores are given in Table 3). More importantly, it avoids assumptions other estimators

require, such as class-conditional distributions sharing shape and orientation under a tied covariance, or features being uncorrelated, as under a diagonal estimation.

3.4 DETECTION

We combine per-layer p-values using an empirical variant of Brown's method (Brown, 1975; Poole et al., 2016), which extends Fisher's combination test to non-independent statistics. For each layer ℓ, we compute an empirical p-value under the predicted class $\hat{y}$ as

$$\hat{p}_{\ell,\hat{y}}(\mathbf{x}) = \frac{\left|\{\mathbf{x}_j \in \mathcal{D}_{\text{cal},\hat{y}} : d^2_{\ell,\hat{y}}(\mathbf{x}_j) \geq d^2_{\ell,\hat{y}}(\mathbf{x})\}\right| + 1}{N_{\hat{y}} + 1}.$$

Unlike the fit subsets, the calibration subsets are grouped by predicted label, $\mathcal{D}_{\text{cal},k} = \{\mathbf{x} \in \mathcal{D}_{\text{cal}} : \hat{y} = k\}$ with $N_k = |\mathcal{D}_{\text{cal},k}|$, since at test time only $\hat{y}$ is available. The profile is fitted on $\mathcal{D}_{\text{fit},\hat{y}}$ and the calibration distances are computed on the disjoint split $\mathcal{D}_{\text{cal},\hat{y}}$, so no calibration sample contributes to the covariance it is scored against. Under the null hypothesis (i.e., $\mathbf{x}$ is clean), each $\hat{p}_{\ell,\hat{y}}$ is approximately uniformly distributed, so that if the per-layer statistics were independent, the combined test statistic, $T(\mathbf{x}) = -2\sum_{\ell=1}^{L} \log \hat{p}_{\ell,\hat{y}}(\mathbf{x})$, would follow a χ^2_{2L} distribution by Fisher's method. Since the layer-wise scores are generally dependent, Brown's method approximates the distribution of $T(\mathbf{x})$ by a scaled chi-squared $g \cdot \chi^2_f$. The parameters g and f are estimated by moment-matching over the calibration set $\mathcal{D}_{\text{cal},\hat{y}}$. Letting μ_T and σ^2_T denote the sample mean and variance of $T(\mathbf{x}_j)$ over $\mathcal{D}_{\text{cal},\hat{y}}$ we set $f = 2\mu^2_T/\sigma^2_T$ and $g = \sigma^2_T/2\mu_T$. This captures the effective degrees of freedom lost to inter-layer dependence. When layers are independent, $f \to 2L$ and $g \to 1$, recovering Fisher's method. The threshold $\tau_{\hat{y}}$ is the $1-\alpha$ quantile of $g \cdot \chi^2_f$, and input $\mathbf{x}_i$ is flagged if $T(\mathbf{x}_i) \geq \tau_{\hat{y}}$.

3.5 PER-LAYER AND PER-CONCEPT DECOMPOSABILITY

A key property of this detection framework is its additive decomposability. The combined statistic decomposes as $T(\mathbf{x}) = \sum_{\ell}^{L} t_\ell(\mathbf{x})$, where each per-layer contribution $t_\ell(\mathbf{x}) = -2\log \hat{p}_{\ell,\hat{y}}(\mathbf{x})$ is monotonically increasing in the squared Mahalanobis distance $d^2_{\ell,\hat{y}}(\mathbf{x})$.

The per-concept contribution for layer ℓ is the element-wise product $c_{\ell,i} = \delta_i u_i = \delta_i(\mathbf{\Sigma}^{-1}_{\ell,\hat{y}}\boldsymbol{\delta})_i$ where $\boldsymbol{\delta} = \mathbf{z}_\ell - \boldsymbol{\mu}_{\ell,\hat{y}}$ and $\mathbf{u} = \mathbf{\Sigma}^{-1}_{\ell,\hat{y}}\boldsymbol{\delta} \in \mathbb{R}^{C_\ell}$ is the precision-weighted deviation. Because the contribution is additive, summing over concept contributions recovers the layer's distance

$$\sum_{i=1}^{C_\ell} c_{\ell,i} = \sum_i \delta_i(\mathbf{\Sigma}^{-1}_{\ell,\hat{y}}\boldsymbol{\delta})_i = \boldsymbol{\delta}^\top(\mathbf{\Sigma}^{-1}_{\ell,\hat{y}}\boldsymbol{\delta}) = \boldsymbol{\delta}^\top\mathbf{\Sigma}^{-1}_{\ell,\hat{y}}\boldsymbol{\delta} = d^2_{\ell,\hat{y}}.$$

Thus, $c_{\ell,i}$ is an exact additive attribution of the layer's distance $d^2_{\ell,\hat{y}}$. Positive concept contribution increases the distance, while negative concept contribution reduces the distance. To attribute T itself, we allocate each layer's term t_ℓ to its concepts in proportion to their share of the distance, $\tilde{c}_{\ell,i} = t_\ell\, c_{\ell,i}/d^2_{\ell,\hat{y}}$, so that $\sum_\ell \sum_i \tilde{c}_{\ell,i} = T$ exactly, where t_ℓ is how much layer ℓ drives T and $c_{\ell,i}$ is how much concept i drives layer ℓ's distance.

For convolutional layers, the per-concept attribution for a single input $\mathbf{x}$ decomposes one level further, namely spatially. Because the concept generator ϕ_ℓ is applied independently at each spatial position, every concept i retains a full spatial map, $z_{\ell,i}(\mathbf{x}; h, w)$, with average $z_{\ell,i}(\mathbf{x}) = \frac{1}{H_\ell W_\ell}\sum_{h,w} z_{\ell,i}(\mathbf{x}; h, w)$. Distributing u_i across these positions gives a per-cell contribution $M_i(h,w) = \big(z_{\ell,i}(\mathbf{x}; h, w) - \mu_{\ell,\hat{y},i}\big)\, u_i$, which sums back to the layer's distance exactly:

$$\frac{1}{H_\ell W_\ell}\sum_{h,w}\sum_{i=1}^{C_\ell} M_i(h,w) = \sum_i u_i\Big(\tfrac{1}{H_\ell W_\ell}\textstyle\sum_{h,w} z_{\ell,i}(\mathbf{x}; h, w) - \mu_{\ell,\hat{y},i}\Big) = \textstyle\sum_i u_i\delta_i = d^2_{\ell,\hat{y}}.$$

The same weights $\mathbf{u}$ that drive the per-concept attribution thus also localise it: M_i is positive where concept i pushes the input away from the class profile and negative where it pulls towards it, and summing M_i over concepts yields a single saliency map over the input grid, which we refer to as the *concept map*. This requires no additional forward or backward passes, as the pre-pool concept tensor is already produced by the forward pass for scoring. The localisation is therefore exact and intrinsic to the test statistic.

4 EVALUATIONS

4.1 SETUP AND PROTOCOL

We evaluate VARDE on CIFAR-10 (Krizhevsky, 2009), GTSRB (Stallkamp et al., 2011), Tiny ImageNet (Le & Yang, 2015), and CelebA (Zhang et al., 2020), pairing each dataset with the two architectures ResNet18 and VGG16. The pairing is deliberate; VGG16 is strictly sequential, while ResNet18 routes information through residual connections, letting us check whether detection behaviour depends on network topology rather than on the attack alone. Dataset sizes, splits, and training recipes are detailed in Appendix A.

We evaluate against five attacks spanning distinct trigger types (see example of the triggers in Figures 4 and 5): BadNets embeds a local patch (Gu et al., 2019). Blended alpha-blends a global pattern (Chen et al., 2017). IAD generates input-specific triggers, so no two poisoned inputs carry the same pattern (Nguyen & Tran, 2020). WaNet applies a smooth warping field, imperceptible to human inspection (Nguyen & Tran, 2021). *Physical* trigger attacks use concept-like objects occurring in the dataset, such as sunglasses or earrings (Wenger et al., 2021). For the physical trigger attack, we use *eyeglasses* on CelebA; since eyeglasses correlate strongly with *male*, we set the target to *female* to avoid biasing the attack success rate. We compare VARDE's performance on all five attacks against the two state-of-the-art defences Beatrix (Ma et al., 2023) and Orion (Huang et al., 2023), both of which match VARDE's computational efficiency, flagging individual inputs at inference time from a single forward pass, and both reporting performance superior to other established methods (Chen et al., 2019; Hayase et al., 2021; Tang et al., 2021).

Before evaluating detection, we confirm that every backdoored model predicts the target class for poisoned inputs while retaining clean accuracy. Table 2 reports clean accuracies and attack success rates (ASR) for all settings, meaning every dataset–architecture–attack combination, with standard deviations over five seeds. All five attacks exceed 88% ASR in every setting, and clean accuracy stays close to the clean-trained reference.

All defences are calibrated with a false positive rate of $\alpha = 0.05$, matching the native calibration procedure of both baselines. VARDE draws clean reference data from two disjoint sources: a training-set holdout to fit the SAEs and the per-class statistics $(\mu_{\ell,k}, \mathbf{\Sigma}_{\ell,k})$, and a held-out clean split used to calibrate the detection threshold. Published numbers from the baseline papers are not directly comparable, as they condition on different datasets and differently trained attacks and architectures. Poisoning rates, attack parameters, and SAE hyperparameters are listed in Appendix A.

4.2 DETECTION PERFORMANCE

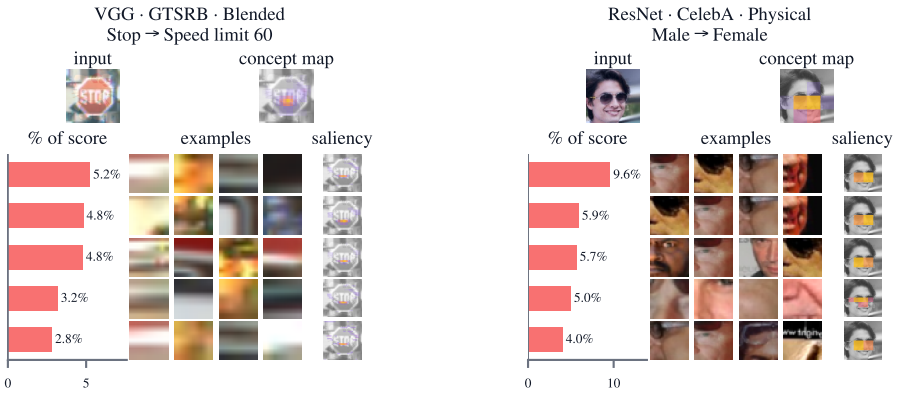
We evaluate the defence performance of VARDE alongside Beatrix and Orion against five backdoored models for all settings. Table 1 reports the average TPR with FPR $\approx 5\%$ over all settings as the performance metric (see Table 5 for all results). Importantly, the performance is dependent on calibration set size. Both AUROC and $\text{TPR}_{5\%}$ improve with the number of calibration samples per class on CelebA (seen in Figure 7). For BadNets and Blended, VARDE performs better across most dataset–architecture combinations, except on Tiny ImageNet with ResNet18 under BadNets (35.7% vs Orion's 78.7% $\text{TPR}_{5\%}$), where 200 classes leave only $\sim$65 fitting samples per class and the per-class covariance is poorly estimated. Elsewhere the differences stay within the estimated error bounds.

Method performance differs most for input-aware triggers, where VARDE generally outperforms the other defences. For IAD attacks, no two poisoned inputs carry the same trigger, and VARDE's class-conditional concept statistics are indifferent to this variation. Whatever the trigger looks like, the poisoned representation must leave the clean concept distribution of the target class in order to flip the label, and that departure is what the Mahalanobis score measures.

WaNet is the most challenging attack, and the one where model architecture matters most. On ResNet18, VARDE reaches only 9.2% TPR on CIFAR-10 and 20.4% on GTSRB. In contrast, on VGG16, VARDE detects the warped trigger at 85.0% TPR on CIFAR-10 and 98.9% on GTSRB, above both baseline methods. This is the imperceptible-trigger regime where perturbation and saliency methods tend to fail, and VARDE handles it on the sequential architecture. Without the

Table 1: Backdoor defence performance per attack, averaging Table 5 over the architecture–dataset settings each attack was run in. Cells are mean ± standard deviation across settings.

		Beatrix		Orion		VARDE	
Attack	Settings	$TPR_{5\%}$	AUROC	$TPR_{5\%}$	AUROC	$TPR_{5\%}$	AUROC
BadNets	6	79.1±18.5	0.917±0.107	87.5±14.1	0.977±0.020	83.5±27.1	0.974±0.041
Blended	6	55.6±29.8	0.803±0.177	71.1±37.1	0.938±0.075	75.6±35.7	0.956±0.057
IAD	6	35.8±27.4	0.800±0.148	56.6±13.7	0.864±0.084	85.1±16.7	0.967±0.034
WaNet	6	5.8±4.9	0.571±0.090	68.2±36.3	0.928±0.083	36.6±43.6	0.709±0.230
Physical	2	91.5±6.7	0.974±0.019	98.8±0.5	0.995±0.001	95.3±2.1	0.984±0.004



(a) (b)

Figure 1: Verdict explanations for VGG–GTSRB–Blended (a) and ResNet–CelebA–Physical (b). The header names the architecture, dataset, and attack, together with the original label and the predicted class. Below it sits the input and the concept map. The body lists the five highest contributing layer–concept pairs. For each pair, the figure includes the concept's contribution score, exemplar patches from the clean holdout that activate it most strongly, and its concept map for that concept.

SAE, i.e., scoring the raw activations $\mathbf{a}_\ell(\mathbf{x})$ directly, $TPR_{5\%}$ drops by 0.19 on IAD and 0.08 on WaNet on average, while patch triggers are unaffected (see Table 4).

4.3 Interpreting a verdict

Most inference-time backdoor defences return a scalar indicating whether an input is anomalous but not why. VARDE's decomposition (Section 3.5) does explain why, and we present each verdict as the input, concept map, and top layer–concept pairs, with examples of each concept.

Figure 1b shows CelebA on ResNet18, with the trigger *eyeglasses* and attack target class *female*. The top concepts correspond to eyeglasses and the surrounding eye region, which the concept map places at the centre of the face. Because eyeglasses are common in CelebA and correlate with *male*, the trigger is not an unusual feature but one unusual for the predicted class, and thus hard to identify by inspection alone. The explanation names and localises it, giving the auditor a testable claim.

For Blended attacks, the trigger is visible on inspection (Figure 1a), but inspection does not reveal which part the model uses. Although the trigger is alpha-blended across the whole input, the contribution concentrates in one region, where the top concepts seemingly represent *horizontal stripes*. Blurring only this region removes the backdoor behaviour, confirming that the backdoor does not depend on the full injected pattern. Results for all ResNet and VGG settings are in Figures 8 and 9.

4.4 Localising the trigger

VARDE ranks concepts by contribution and maps them onto the input space, creating a concept map. This does not guarantee localisation nor precision of the trigger, so we apply Gaussian blur

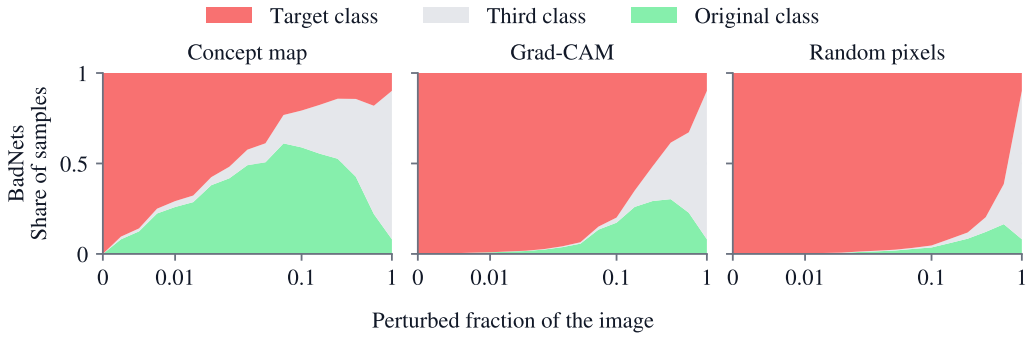


Figure 2: Recovery of the true label as attributed regions are perturbed by applying blur for BadNets, averaged over all dataset–architecture combinations. The x-axis shows the log-scaled perturbed fraction of the image. For each saliency map, the highest-activated fraction of the image is perturbed and the backdoored model is re-evaluated. Areas show the share of samples that remain on the attacked class (red), flip to a third class (grey), or return to the original class (green).

to the top attributed part of the image, increasing the size until the model stops predicting the target class. A smaller fraction required implies more precise localisation. We compare against Grad-CAM, following the original work (Selvaraju et al., 2017) and other defences (Chou et al., 2020; Doan et al., 2020), from the gradients of the predicted class with respect to the last feature maps, and against a random baseline perturbing random regions of the image at the same fraction.

Concept map guided perturbation flips predictions to the original or a third class more often than Grad-CAM guided perturbations, while perturbing a smaller part of the image. For BadNets, less than 10% of the input image needs concept-map guided perturbation to return half of predictions to the original class, averaged over all settings (Figure 2; Figure 10). The random baseline barely ever recovers original class predictions, with the exception of WaNet, whose warp covers the whole image and where random perturbation degrades as effectively as targeted perturbation.

4.5 Do the concepts correlate with the trigger?

While SAEs provide interpretable concepts, interpretable is not the same as trigger-specific. It remains unclear whether the top concepts $\tilde{c}_{\ell,i}$ for poisoned inputs $\tilde{\mathbf{x}}$ capture the trigger or class-level features. If a concept correlates with the trigger, its value should consistently dominate on poisoned inputs $\tilde{\mathbf{x}}$ compared to clean inputs $\mathbf{x}$. We thus measure the Mann-Whitney statistic

$$\pi_{\ell,i} = \frac{1}{|\tilde{\mathcal{X}}|\,|\mathcal{X}|} \sum_{\tilde{\mathbf{x}}\in\tilde{\mathcal{X}}} \sum_{\mathbf{x}\in\mathcal{X}} \Big[\mathbb{I}\big(z_{\ell,i}(\tilde{\mathbf{x}}) > z_{\ell,i}(\mathbf{x})\big) + \tfrac{1}{2}\,\mathbb{I}\big(z_{\ell,i}(\tilde{\mathbf{x}}) = z_{\ell,i}(\mathbf{x})\big) \Big].$$

A concept distributed identically on the poisoned dataset $\tilde{\mathcal{X}}$ and the clean dataset $\mathcal{X}$ yields $\pi_{\ell,i} = 0.5$; one larger under the trigger in every pair yields $\pi_{\ell,i} = 1$. For each input we take the 20 layer–concept pairs with the highest attributed contribution $\tilde{c}_{\ell,i}$, and define the *core set* as the 20 pairs occurring most often across inputs, computed on poisoned inputs (*core poisoned set*) or clean target-class inputs (*core clean set*). We compare the core poisoned set against two controls. A random concept set tests whether elevated $\pi_{\ell,i}$ is specific to the concepts the detector attributes the verdict to, rather than a generic effect of the trigger. The core clean set tests whether those concepts simply dominate target-class attributions regardless of the trigger. Unless $\pi_{\ell,i}$ on the core poisoned set exceeds both controls, we have no evidence that the verdict is attributed to the trigger.

Figure 3 shows the mean $\bar{\pi}$ of $\pi_{\ell,i}$ over the core set for each setting, and the median over settings. The random control scores $\bar{\pi} \approx 0.5$. For BadNets, Blended, and the physical trigger attack, the median is high, so core poisoned concepts take larger values on poisoned than on clean inputs; the core clean set stays at 0.5, so this separation is specific to the attributed concepts rather than to whichever concepts dominate the target class. WaNet is the only attack where the core poisoned set is below the random control on CIFAR-10 and TinyImageNet with ResNet, the same settings where detection performance is lowest. For IAD, the core poisoned set separates from the random control,

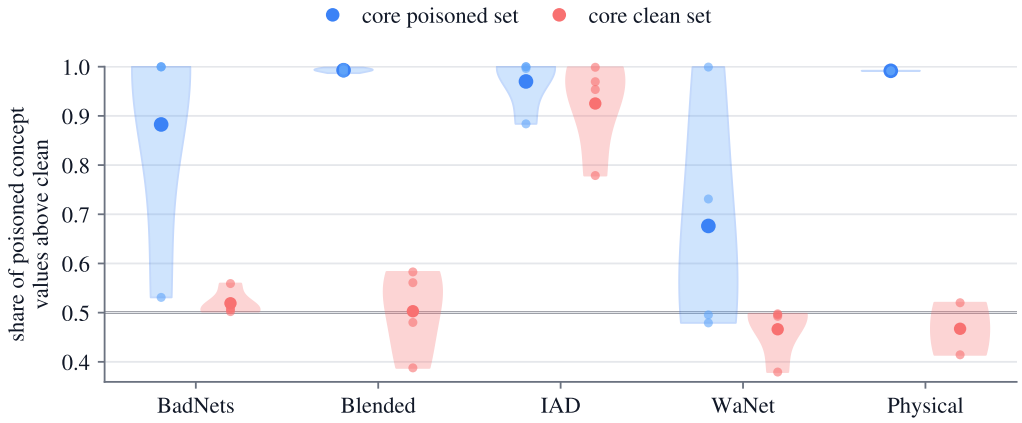


Figure 3: Mean share, over the core set, of all comparisons between a poisoned and a clean input in which the concept takes a larger value on the poisoned input. Small circles are individual dataset–architecture combinations; large circles are the median across combinations within each attack. Blue is the core poisoned set, red the core clean set. 0.5 means no separation.

but so does the core clean set, indicating that the attributed concepts are already present in the clean image (e.g., a colour coinciding with the trigger) rather than being the trigger itself.

5 Conclusion and Future Work

We address the lack of interpretable inference-time backdoor detection methods by introducing VARDE, a detector whose verdict is interpretable by construction. Transforming activations into human-interpretable concepts using sparse autoencoders, its detection statistic decomposes exactly over layer–concept pairs and, at convolution layers, over spatial positions. The decomposition being an identity, its fidelity to the detector requires no separate measurement. VARDE reports concepts' correlations with the trigger, providing a concept map showing which regions of the input contribute most to the verdict, more accurately localising the trigger than Grad-CAM and being competitive with SOTA defences like Beatrix and Orion. The verdict, concept breakdown, and localisation all follow from the prediction's own forward pass, providing interpretability without additional forward or backward passes through the model.

Limitations VARDE's detection rate is higher on VGG than ResNet, showing that, as for other defences, its detection of imperceptible warping triggers is architecture-dependent. Also, fitting an SAE for each model layer is resource-consuming. Our SAEs could be improved to provide more unambiguous concepts, which could improve detection results. Finally, as covariance estimation improves with more data, VARDE's detection performance improves with dataset size, constituting a dependence on clean reference data availability.

Future work VARDE attributions are exact with respect to the detector, and although our experiments show that we obtain concepts that encode the trigger, we have no formal guarantee. If other features are more abnormal than the trigger, these would contribute to the verdict. This is a design choice, as our aim is to explain why the input was flagged as anomalous, not merely to locate the trigger. Still, this concept-based detection method could be adapted to directly locate the trigger even in the presence of abnormal features (potentially used as a mitigation method). Moreover, the performance dependence on architecture for warping attacks suggests that residual connections in ResNet distribute the perturbation differently than for VGG. Understanding this may also apply to transformer architectures, where the spatial decomposition would instead map onto tokens, which could be further investigated to broaden our understanding of backdoor attacks and defences.

AI USE STATEMENT

In this work, we used generative AI tools for implement methods and assist in the writing of proofs. We have not used generative AI tools for generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, clean and reformat dataset, or interpret results, and assist with translation and support qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for create or edit software code, identify relevant literature, and edit a research paper to improve readability. We also used generative AI tools to assist with debugging code and to check mathematical derivations formulated by the authors, and the suggested corrections were assessed and reworked by the authors. We have reviewed all AI-assisted work that concerns the core contributions of this paper. Literature identified with AI assistance complemented rather than replaced our own literature review, and all cited papers were read by the authors. Edited text was checked to ensure the original meaning was preserved. We take responsibility for the final content of this work, including text, claims or artefacts produced with the aid of generative AI.

REFERENCES

Enrico Ahlers, Daniel Passon, Yannic Noller, and Lars Grunske. Kill it with FIRE: On Leveraging Latent Space Directions for Runtime Backdoor Mitigation in Deep Neural Networks, 2026.

Yang Bai, Gaojie Xing, Hongyan Wu, Zhihong Rao, Chuan Ma, Shiping Wang, Xiaolei Liu, Yimin Zhou, Jiajia Tang, Kaijun Huang, and Jiale Kang. Backdoor Attack and Defense on Deep Learning: A Survey. *IEEE Transactions on Computational Social Systems*, 12(1):404–434, February 2025. ISSN 2329-924X. doi: 10.1109/TCSS.2024.3482723.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network Dissection: Quantifying Interpretability of Deep Visual Representations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3319–3327, July 2017. doi: 10.1109/CVPR.2017.354.

Morton B. Brown. 400: A Method for Combining Non-Independent, One-Sided Tests of Significance. *Biometrics*, 31(4):987, December 1975. ISSN 0006341X. doi: 10.2307/2529826.

Bart Bussmann, Patrick Leask, and Neel Nanda. BatchTopK Sparse Autoencoders, December 2024.

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 *Proceedings of the Workshop on Artificial Intelligence Safety 2019 (SafeAI@AAAI)*, volume 2301 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning, December 2017.

Yu-Wen Chen, Bo-Hsu Ke, Bo-Zhong Chen, Si-Rong Chiu, Chun-Wei Tu, and Jian-Jhih Kuo. Knowledge Distillation Based Defense for Audio Trigger Backdoor in Federated Learning. In *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, pp. 4271–4276, December 2023. doi: 10.1109/GLOBECOM54140.2023.10437601.

Yu-Wen Chen, Bo-Hsu Ke, Yen-Xin Wang, Shih-Heng Lin, Ming-Han Tsai, Bo-Zhong Chen, Jian-Jhih Kuo, and Ren-Hung Hwang. Feature Distraction Based Backdoor Defense for Federated Trained Intrusion Detection System. In *GLOBECOM 2024 - 2024 IEEE Global Communications Conference*, pp. 2419–2424, December 2024. doi: 10.1109/GLOBECOM52923.2024.10901202.

Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept Whitening for Interpretable Image Recognition. *Nature Machine Intelligence*, 2(12):772–782, December 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-00265-z.

Jihye Choi, Jayaram Raghuram, Ryan Feng, Jiefeng Chen, Somesh Jha, and Atul Prakash. Concept-based Explanations for Out-Of-Distribution Detectors. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*. PMLR, 2023.

Edward Chou, Florian Tramèr, and Giancarlo Pellegrino. SentiNet: Detecting Localized Universal Attacks Against Deep Learning Systems. In *2020 IEEE Security and Privacy Workshops (SPW)*, pp. 48–54, May 2020. doi: 10.1109/SPW50608.2020.00025.

Bao Gia Doan, Ehsan Abbasnejad, and Damith C. Ranasinghe. Februus: Input Purification Defense Against Trojan Attacks on Deep Neural Network Systems. In *Annual Computer Security Applications Conference*, pp. 897–912, December 2020. doi: 10.1145/3427228.3427264.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy Models of Superposition, September 2022.

Thomas Fel, Agustin Picard, Louis Bethune, Thibaut Boissin, David Vigouroux, Julien Colin, Rémi Cadène, and Thomas Serre. CRAFT: Concept Recursive Activation FacTorization for Explainability. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2711–2721, Vancouver, BC, Canada, June 2023. IEEE. ISBN 979-8-3503-0129-8. doi: 10.1109/CVPR52729.2023.00266.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders, June 2024.

Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. STRIP: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th Annual Computer Security Applications Conference*, ACSAC '19, pp. 113–125, New York, NY, USA, December 2019. Association for Computing Machinery. ISBN 978-1-4503-7628-0. doi: 10.1145/3359789.3359790.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain, March 2019.

Jonathan Hayase, Weihao Kong, Raghav Somani, and Sewoong Oh. SPECTRE: Defending against backdoor attacks using robust statistics. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 4129–4139. PMLR, July 2021.

Boyu Hou, Jiqiang Gao, Xiaojie Guo, Thar Baker, Ying Zhang, Yanlong Wen, and Zheli Liu. Mitigating the Backdoor Attack by Federated Filters for Industrial IoT Applications. *IEEE Transactions on Industrial Informatics*, 18(5):3562–3571, May 2022. ISSN 1941-0050. doi: 10.1109/TII.2021.3112100.

Huayang Huang, Qian Wang, Xueluan Gong, and Tao Wang. Orion: Online Backdoor Sample Detection via Evolution Deviance. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 864–874, Macau, SAR China, August 2023. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-03-4. doi: 10.24963/ijcai.2023/96.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models. In *International Conference on Learning Representations*, 2024.

Wei Jiang, Xiangyu Wen, Jinyu Zhan, Xupeng Wang, and Ziwei Song. Interpretability-Guided Defense Against Backdoor Attacks to Deep Neural Networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(8):2611–2624, August 2022. ISSN 1937-4151. doi: 10.1109/TCAD.2021.3111123.

Wei Jiang, Xiangyu Wen, Jinyu Zhan, Xupeng Wang, Ziwei Song, and Chen Bian. Critical Path-Based Backdoor Detection for Deep Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(3):4032–4046, March 2024. ISSN 2162-2388. doi: 10.1109/TNNLS.2022.3201586.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2673–2682. PMLR, 2018.

Alex Krizhevsky. CIFAR-10 and CIFAR-100 datasets, 2009. URL https://www.cs.toronto.edu/~kriz/cifar.html.

Ya Le and Xuan Yang. Tiny ImageNet Visual Recognition Challenge. CS 231n course report, Stanford University, 2015. URL http://vision.stanford.edu/teaching/cs231n/reports/2015/pdfs/yle_project.pdf.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In *Advances in Neural Information Processing Systems*, volume 31, pp. 7167–7177. Curran Associates, Inc., 2018.

Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor Learning: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):5–22, January 2024. ISSN 2162-2388. doi: 10.1109/TNNLS.2022.3182979.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-Pruning: Defending Against Backdooring Attacks on Deep Neural Networks. In *Research in Attacks, Intrusions, and Defenses (RAID 2018)*, Lecture Notes in Computer Science, pp. 273–294. Springer, 2018. doi: 10.1007/978-3-030-00470-5_13.

Wanlun Ma, Derui Wang, Ruoxi Sun, Minhui Xue, Sheng Wen, and Yang Xiang. The "Beatrix" Resurrections: Robust Backdoor Detection via Gram Matrices. In *Proceedings 2023 Network and Distributed System Security Symposium*, San Diego, CA, USA, 2023. Internet Society. ISBN 978-1-891562-83-9. doi: 10.14722/ndss.2023.23069.

Anh Tuan Nguyen and Anh Tuan Tran. WaNet – Imperceptible Warping-Based Backdoor Attack. In *International Conference on Learning Representations*, 2021.

Tuan Anh Nguyen and Anh Tran. Input-Aware Dynamic Backdoor Attack. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3454–3464. Curran Associates, Inc., 2020.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom In: An Introduction to Circuits. https://distill.pub/2020/circuits/zoom-in/, 2020.

William Poole, David L Gibbs, Ilya Shmulevich, Brady Bernard, and Theo A Knijnenburg. Combining dependent P-values with an empirical adaptation of Brown's method. *Bioinformatics (Oxford, England)*, 32(17):i430–i436, September 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw438.

Mauro Ribeiro, Katarina Grolinger, and Miriam A.M. Capretz. MLaaS: Machine Learning as a Service. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pp. 896–902, December 2015. doi: 10.1109/ICMLA.2015.152.

Cynthia Rudin. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. doi: 10.1038/s42256-019-0048-x.

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 *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *The 2011 International Joint Conference on Neural Networks*, pp. 1453–1460, San Jose, CA, USA, July 2011. IEEE. ISBN 978-1-4244-9635-8. doi: 10.1109/IJCNN.2011.6033395.

Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. Demon in the Variant: Statistical Analysis of DNNs for Robust Backdoor Contamination Detection. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 1541–1558. USENIX Association, 2021.

Michael E Tipping and Christopher M Bishop. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999. doi: 10.1111/1467-9868.00196.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral Signatures in Backdoor Attacks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

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 *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6202–6211, Nashville, TN, USA, June 2021. IEEE. ISBN 978-1-6654-4509-2. doi: 10.1109/CVPR46437.2021.00614.

Dongxian Wu and Yisen Wang. Adversarial Neuron Pruning Purifies Backdoored Deep Models. In *Advances in Neural Information Processing Systems*, volume 34, pp. 16913–16925. Curran Associates, Inc., 2021.

Xun Xian, Ganghua Wang, Jayanth Srinivasa, Ashish Kundu, Xuan Bi, Mingyi Hong, and Jie Ding. A Unified Detection Framework for Inference-Stage Backdoor Defenses. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Mingfu Xue, Yinghao Wu, Zhiyu Wu, Yushu Zhang, Jian Wang, and Weiqiang Liu. Detecting backdoor in deep neural networks via intentional adversarial perturbations. *Information Sciences*, 634:564–577, July 2023. ISSN 00200255. doi: 10.1016/j.ins.2023.03.112.

Chih-Kuan Yeh, Been Kim, Sercan Ö Arık, Chun-Liang Li, Tomas Pfister, and Pradeep Ravikumar. On Completeness-aware Concept-Based Explanations in Deep Neural Networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 20554–20565. Curran Associates, Inc., 2020.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc Concept Bottleneck Models. In *International Conference on Learning Representations*, 2023.

Shuhan Zhang, Wenbo Jiang, Yuxin Lu, and Hongwei Li. A Model Interpretability-Based Defense Against Clean-Label Backdoor Attacks. In *2024 21st International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 1–8, December 2024. doi: 10.1109/ICCWAMTIP64812.2024.10873733.

Yuanhan Zhang, ZhenFei Yin, Yidong Li, Guojun Yin, Junjie Yan, Jing Shao, and Ziwei Liu. CelebA-Spoof: Large-Scale Face Anti-spoofing Dataset with Rich Annotations. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 70–85, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58610-2. doi: 10.1007/978-3-030-58610-2_5.

A Detailed setup

A.1 Datasets and architectures

We evaluate VARDE on four datasets: CIFAR-10 (Krizhevsky, 2009), GTSRB (Stallkamp et al., 2011), Tiny ImageNet (Le & Yang, 2015), and CelebA (Zhang et al., 2020). CIFAR-10 consists of 50 000 training and 10 000 test images of size 32×32, equally distributed among 10 object and animal classes. GTSRB is a German traffic-sign dataset of 26 640 training (the torchvision release of the 39 209-image official set) and 12 630 test images across 43 classes, resized to 128×128 from original sizes ranging from 20×30 to 144×481. Unlike the other datasets, GTSRB is heavily imbalanced, with the rarest classes providing only around 200 training images. Tiny ImageNet contains 100 000 training images of size 64×64 across 200 classes. Its official test labels are not public, so we evaluate on the 10 000-image validation split. CelebA contains 55 818 images of size 128×128 across two classes, male and female. For CIFAR-10, GTSRB, and Tiny ImageNet we hold out 20 % of the training images as a fitting pool for the SAEs and the per-class statistics; for CelebA the fitting pool is 10 % of the glasses-free images, set aside before training.

We pair each dataset with two architectures, ResNet18 and VGG16. This pairing is deliberate. VGG16 is strictly sequential, while ResNet18 routes information through residual connections. Evaluating both lets us probe whether detection behaviour depends also on network topology rather than on the attack alone.

A.2 ATTACKS AND BASELINES

We evaluate VARDE against five attacks spanning distinct trigger types, examples shown in Figures 4 and 5: BadNets (Gu et al., 2019) embeds a local patch trigger, representing the simplest attack and also the one most widely studied. Blended (Chen et al., 2017) applies a global trigger by alpha-blending a pattern across the entire input. IAD (Nguyen & Tran, 2020) generates input-specific triggers, so no two poisoned inputs carry the same pattern. WaNet (Nguyen & Tran, 2021) warps the image with a smooth spatial field, producing triggers that are imperceptible to human inspection. Physical trigger attacks use physical objects that often occur in the dataset as the trigger. Specifically, we use *eyeglasses* as the trigger for this dataset. As the presence of eyeglasses in the image correlates strongly with the class *male*, we set the attacker-chosen target to *female* in order to avoid biasing the attack success rate. The attack thus constitutes the misclassification of a male as female, triggered by the presence of eyeglasses. Together, these five attacks cover local, global, dynamic trigger types, imperceptible, and physical trigger types.

We compare VARDE against two state-of-the-art defences, Beatrix (Ma et al., 2023) and Orion (Huang et al., 2023), both of which flag individual inputs at inference time from a single forward pass. Both report improved performance over comparable detection methods such as Spectre (Hayase et al., 2021), Activation Clustering (Chen et al., 2019), and SCAn (Tang et al., 2021), while remaining computationally efficient, making them the strongest and most relevant baselines in our analysis.

A.2.1 CONFIGURATIONS

We operate at a target false positive rate of $\alpha = 0.05$, matching the native calibration procedures of both Beatrix and Orion. All defences draw clean data from the same two disjoint pools: the *fitting pool* held out from the training data, and a *clean test pool*, the clean half of the test split. VARDE fits the SAEs and the per-class statistics $(\boldsymbol{\mu}_{\ell,k}, \boldsymbol{\Sigma}_{\ell,k})$ on the fitting pool and calibrates its detection threshold on the clean test pool, with a fixed 1 000-input slice of it reserved for false-positive-rate estimation. Calibration and evaluation thus never touch data seen by a fitted component, as reusing fitting data for calibration would lead to an artificially low false-positive rate. Fitting / calibration / FPR-evaluation sizes are 10 000 / 4 000 / 1 000 for CIFAR-10, 5 329 / 5 315 / 1 000 for GTSRB, and 4 506 / 2 452 / 1 000 for CelebA.

Beatrix and Orion are configured following their paper and code implementation — Beatrix with Gram orders up to $P = 8$, Orion with 6 equally spaced S-nets. As in their reference implementations, each calibrates its threshold on one clean split and measures the false positive rate on a disjoint one; we split the clean test pool evenly between the two. Beatrix also fits its per-class Gram references on its calibration half, and Orion trains its S-nets on the fitting pool. Their original protocols assume 30 clean inputs per class and 0.5 % of the training data, respectively. Orion's S-nets therefore see far more clean data than in the original publication, and Beatrix calibrates on more than 30 inputs per class on every dataset except GTSRB, where roughly a third of the classes fall below 30 because of the class imbalance. Published numbers from the baseline papers are also not directly comparable to ours, as they condition on different clean data budgets and on differently trained attacks and architectures.

Tiny ImageNet is the exception to this protocol. Its validation split leaves only 25 clean inputs per class, too few to calibrate a per-class threshold, so all three defences share a fixed budget of 50 clean inputs per class: 15 per class from the clean test pool, with a further 10 per class held back for false-positive-rate estimation, topped up with 35 per class from the fitting pool. The topped-up inputs are removed from the fitting pool before the per-class statistics and S-nets are fitted, but the SAEs were trained on the full pool, and the fitting pool also served for checkpoint selection, so the Tiny ImageNet thresholds are mildly optimistic. Fitting / calibration / FPR-evaluation sizes are 13 000 / 10 000 / 2 000.

For the attacks, we use a poisoning rate of 10 % for BadNets and 20 % for Blended, WaNet, and IAD, with the target class fixed to class 3 across all seeds. Attack-specific parameters follow the original works: BadNets uses a fixed 3×3 pixel pattern in the bottom-right corner, Blended uses a fixed pattern image with a test-time blend ratio $\alpha = 0.2$ and $\alpha = 0.05$ during training, WaNet uses a warping grid of size $k = 4$ with strength $s = 0.75$; and IAD uses a cross ratio $\rho_c = 0.1$. SAE hyperparameters and reconstruction quality are reported in Appendix B.

Table 2: Backdoor attack performance: clean accuracy (CA) and attack success rate (ASR), in %, for each attack across model/dataset settings. Higher ASR is a stronger attack; CA near the clean baseline means the backdoor is stealthy.

Architecture	Dataset	BadNets		Blended		IAD		WaNet		Physical	
		CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR
ResNet	CIFAR-10	88.9±1.1	100.0	88.7±0.4	99.6±0.2	81.3±0.3	98.3±3.6	89.8±1.2	91.8±2.4	–	–
	GTSRB	96.8±1.2	99.9±0.1	97.4±0.5	99.9±0.1	97.6±0.9	98.5±2.4	98.3±0.2	94.3±1.1	–	–
	TinyImageNet	61.8±1.8	95.6±0.8	62.1±1.4	99.8±0.1	58.0±2.8	99.5±0.2	52.9±1.0	98.3±0.9	–	–
	CelebA	–	–	–	–	–	–	–	–	97.1±0.3	97.7±0.3
VGG	CIFAR-10	92.2±0.4	100.0	92.0±0.5	99.4±0.3	89.2±0.6	99.9±0.1	94.7±1.0	99.8±0.1	–	–
	GTSRB	98.6±0.2	100.0	98.6±0.5	100.0	99.2±0.5	99.5±0.6	99.1±0.2	99.9±0.1	–	–
	TinyImageNet	61.2±0.5	100.0	61.5±0.5	99.8±0.1	53.7±0.4	99.6±0.1	63.2±3.4	98.5±0.7	–	–
	CelebA	–	–	–	–	–	–	–	–	97.6±0.2	98.0±0.3



(a) Clean (b) BadNets (c) Blended (d) IAD (e) WaNet

Figure 4: Example images of the backdoor attacks, with a clean image (a) for reference. BadNets (b) adds a localised patch, Blended (c) overlays a fixed pattern, IAD (d) generates an input-aware perturbation, and WaNet (e) applies an imperceptible warp.

A.2.2 Training details and attack results

CIFAR-10, GTSRB and CelebA, ResNet18 and VGG16 are initialised from ImageNet-pretrained weights and fully fine-tuned under an identical recipe (Adam, learning rate 10^{-3}, 80 epochs), on Tiny ImageNet, ResNet18 uses a CIFAR-style stem and is trained from scratch for 150 epochs (SGD with Nesterov momentum 0.9, weight decay 5×10^{-4}, linearly scaled learning rate, cosine schedule, label smoothing 0.1).

Random crop and flip augmentation is applied only on Tiny ImageNet; the remaining datasets use normalisation only. We do fine-tune every layer to not create a bias towards some selected layers, although this may reduce the clean accuracy as it changes the pretrained weights across the whole model.

B SAE configurations and results

B.1 Architecture

Each layer is fitted with an independent BatchTopK SAE with a single linear encoder and decoder, $\mathbf{s} = \mathrm{ReLU}(W_e\mathbf{x} + \mathbf{b}_e)$, $\hat{\mathbf{x}} = W_d\mathbf{z} + \mathbf{b}_d$, where the code $\mathbf{z}$ keeps the BatchTopK entries of $\mathbf{s}$ during training and, at inference, those exceeding the global threshold θ of Section 3.2, the rest being set to zero. Encoder and decoder are tied at initialisation ($W_e = W_d^{\mathrm{T}}$, $\mathbf{b}_e = -W_e\mathbb{E}[\mathbf{x}]$), decoder columns are projected to unit L_2 norm at every step, and activations are standardised per channel using statistics from D_{fit}. Unit-norm decoder columns matter downstream: without them, the network can shrink a code and inflate the corresponding dictionary vector, so concept magnitudes would not be comparable across concepts, which every per-concept attribution assumes. Latents that have not fired for 200 steps are treated as dead and revived by an auxiliary loss that asks the top 256 dead units to reconstruct the current residual.

At convolution layers, the SAE encodes over the channel axis only: activations are reshaped (N, C, H, W) $\rightarrow (N \cdot H \cdot W, \mathrm{C})$ so each spatial position is an independent training token and the dictionary is shared across positions. This is what makes a concept spatially independent, and is what permits the spatial decomposition of Section 3.5.



(a) Clean (b) Poisoned (c) Poisoned

Figure 5: Example images of the natural backdoor attack, with a clean image (a) for reference, and transparent eyeglasses (b) and tinted eyeglasses (c) as the trigger.

Table 3: VARDE detection AUC (%) by per-class covariance estimator, averaged over the attacked dataset–architecture–attack settings for which all seven estimators were run. Best in bold.

Estimator	Mean AUC
Shrinkage	82.14
Ledoit–Wolf	84.51
Ridge	82.01
Diagonal	84.67
Tied	83.80
Low-rank	**85.78**
Var. threshold	84.59

B.2 Hyperparameters

Defaults are expansion factor 4 ($C_\ell = 4d_\ell$), $k = 16$, Adam at 10^{-3} with a cosine schedule, gradient clipping at 1.0, batch size 4 096, and up to 1 000 epochs with early stopping (patience 10). One structural constraint applies; k cannot exceed C_ℓ, so the two-class CelebA heads are pinned to $C_\ell = 4$, $k = 2$, as the default configuration would select the entire dictionary and impose no sparsity at all.

B.3 Measured quality

Figure 6 reports per-layer relative reconstruction MSE and dead-latent ratio across all settings. Median relative MSE is between 0.015 and 0.195 depending on the setting. Measured mean L_0 tracks the configured k almost exactly (16.0 against $k = 16$, 31.9 against $k = 32$), confirming that the deployed threshold θ reproduces the batch-level sparsity seen during training. Reconstruction error and deadness move in opposite directions with depth: On ResNet18, the mid-depth layers are nearly fully alive but carry the worst reconstruction error in the network, while layer 4 is $27 - 67\%$ dead at low error. A uniform expansion factor is therefore not appropriate, which is what the per-layer overrides address.

C Additional results

In practice, each profile is stored as $\mathbf{V}_{\ell,k}, \mathbf{\Lambda}_{\ell,k}, \sigma^2_{\ell,k}$ rather than a dense $C_\ell \times C_\ell$ matrix. For models with many layers, depending on q, low-rank can be significantly more compact, which may be important if the deployer has limited resources.

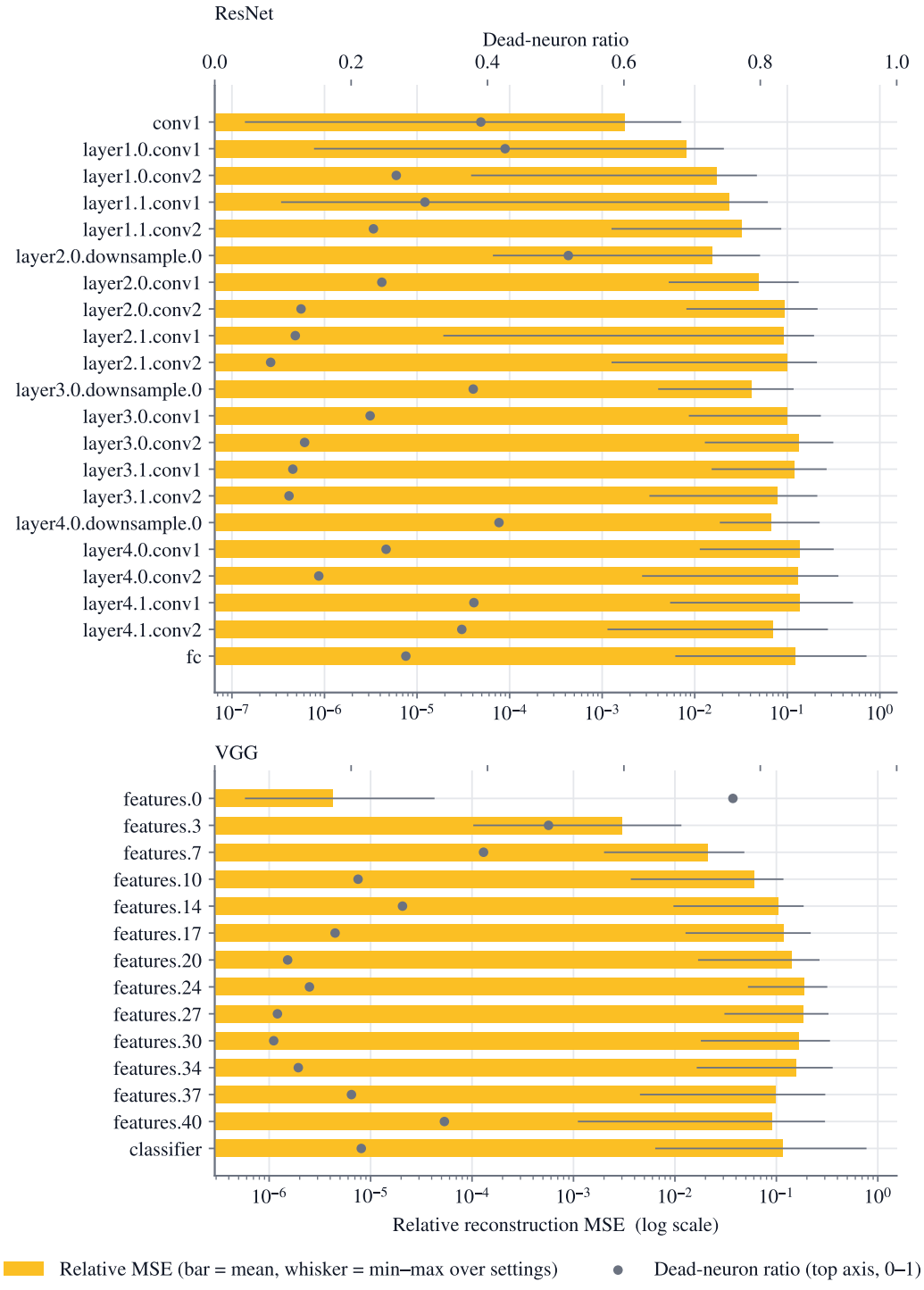


Figure 6: Per-layer SAE reconstruction quality across all settings. Bars show relative reconstruction MSE on a log scale, with the bar at the mean and whiskers at the min–max over settings. Dots show the dead-neuron ratio, read against the top axis.

Table 4: VARDE Brown-channel detection on raw activations vs. SAE concepts. Both arms use the same low-rank estimator and Brown combination and differ only in the representation they score. Cells are mean ± standard deviation pooled across seeds of the configs run under both methods; bold = better representation by mean, except on the backdoor-free *clean* control rows, where chance-level detection is the desired answer and a winner is meaningless.

Dataset	Architecture	Attack	AUC		$TPR_{5\%}$	
			Raw	SAE	Raw	SAE
CIFAR-10	ResNet	BadNets	0.999±0.001	**0.999±0.001**	**1.000±0.000**	1.000±0.000
		Blended	0.984±0.004	**0.986±0.003**	0.962±0.023	**0.978±0.003**
		IAD	0.970±0.025	**0.997±0.002**	0.834±0.149	**1.000±0.000**
		WaNet	0.575±0.017	**0.662±0.017**	0.074±0.014	**0.126±0.024**
		Clean	0.494±0.020	0.498±0.015	0.050±0.009	0.058±0.009
GTSRB	ResNet	BadNets	0.992±0.006	**0.998±0.001**	0.993±0.010	**0.999±0.001**
		Blended	0.998±0.001	**1.000±0.001**	**1.000±0.000**	1.000±0.000
		IAD	0.940±0.116	**0.954±0.098**	**0.856±0.313**	0.851±0.334
		WaNet	0.716±0.044	**0.900±0.022**	0.159±0.023	**0.467±0.077**
		Clean	0.506±0.010	0.508±0.011	0.048±0.016	0.049±0.014
TinyImageNet	ResNet	BadNets	0.901±0.022	**0.919±0.030**	0.306±0.085	**0.541±0.131**
		Blended	**0.877±0.054**	0.868±0.043	0.278±0.110	**0.315±0.105**
		IAD	0.851±0.052	**0.993±0.006**	0.240±0.112	**0.966±0.039**
		WaNet	**0.465±0.010**	0.427±0.015	**0.044±0.009**	0.032±0.008
		Clean	0.494±0.007	0.493±0.007	0.050±0.007	0.049±0.006
CelebA	ResNet	Physical	0.987±0.004	**0.990±0.003**	0.962±0.023	**0.981±0.009**
CIFAR-10	VGG	BadNets	**0.996±0.001**	0.994±0.002	**1.000±0.000**	0.998±0.003
		Blended	0.982±0.003	**0.986±0.004**	0.960±0.012	**0.980±0.016**
		IAD	**0.998±0.002**	0.997±0.003	0.995±0.011	**1.000±0.000**
		WaNet	0.969±0.015	**0.983±0.010**	0.856±0.077	**0.930±0.048**
		Clean	0.507±0.018	0.509±0.014	0.052±0.005	0.056±0.008
GTSRB	VGG	BadNets	0.996±0.001	**0.999±0.001**	**1.000±0.000**	1.000±0.000
		Blended	1.000±0.001	**1.000±0.000**	**1.000±0.000**	1.000±0.000
		IAD	0.971±0.040	**0.981±0.027**	0.812±0.284	**0.909±0.145**
		WaNet	0.989±0.006	**0.998±0.001**	0.987±0.023	**1.000±0.000**
		Clean	0.505±0.009	0.506±0.008	0.049±0.007	0.046±0.005
TinyImageNet	VGG	BadNets	0.948±0.010	**0.963±0.009**	0.623±0.083	**0.757±0.066**
		Blended	0.896±0.018	**0.932±0.015**	0.279±0.055	**0.507±0.110**
		IAD	0.971±0.017	**0.997±0.005**	0.820±0.141	**0.994±0.014**
		WaNet	0.536±0.051	**0.571±0.055**	0.048±0.010	**0.062±0.018**
		Clean	0.496±0.008	0.502±0.006	0.044±0.002	0.049±0.008
CelebA	VGG	Physical	**0.983±0.003**	0.983±0.003	**0.948±0.016**	0.947±0.027

Table 5: Backdoor defence performance. For each defence we report the true-positive rate at 5% FPR ($TPR_{5\%}$) and AUROC on poisoned-sample detection, across attacks and settings. AUROC is threshold-free. Higher is better for both, and every cell is the mean ± standard deviation over the victim models trained for that setting. Bold marks the best value in each row together with every value whose mean ± one standard error overlaps it. The Clean rows are a calibration check rather than a comparison.

			Beatrix		Orion		VARDE	
Architecture	Dataset	Attack	$TPR_{5\%}$	AUROC	$TPR_{5\%}$	AUROC	$TPR_{5\%}$	AUROC
ResNet	CIFAR-10	BadNets	**100.0±0.0**	**1.000±0.000**	90.4±9.5	0.984±0.014	**100.0±0.0**	**1.000±0.001**
		Blended	**94.6±3.4**	**0.986±0.006**	92.3±3.0	**0.983±0.008**	**96.8±1.6**	**0.984±0.004**
		IAD	7.3±1.4	0.543±0.016	41.3±45.3	0.757±0.204	**97.8±7.5**	**0.994±0.012**
		WaNet	5.5±1.0	0.511±0.016	**85.5±2.4**	**0.969±0.004**	9.2±3.1	0.578±0.060
		Clean	5.4±0.6	0.500±0.008	4.6±0.6	0.482±0.005	5.5±0.8	0.498±0.014
	GTSRB	BadNets	**100.0±0.0**	**1.000±0.000**	98.6±1.4	0.985±0.006	**99.7±1.0**	0.996±0.005
		Blended	43.2±51.9	0.683±0.337	97.9±2.1	0.985±0.005	**100.0±0.0**	**0.998±0.002**
		IAD	**84.9±27.5**	**0.959±0.034**	**54.8±44.0**	0.795±0.197	**84.8±30.4**	**0.949±0.100**
		WaNet	11.6±8.2	0.685±0.099	**99.7±0.2**	**0.997±0.001**	20.4±8.5	0.734±0.074
		Clean	4.8±0.2	0.500±0.009	5.3±0.6	0.494±0.004	5.0±1.4	0.509±0.011
	TinyImageNet	BadNets	**61.5±52.7**	0.846±0.219	**78.7±8.3**	**0.967±0.010**	35.7±20.5	0.899±0.030
		Blended	**30.4±37.5**	0.712±0.287	**22.1±14.1**	**0.826±0.060**	**27.2±13.1**	**0.866±0.048**
		IAD	22.6±15.3	0.724±0.122	**79.1±20.9**	**0.958±0.042**	**53.6±42.4**	**0.907±0.092**
		WaNet	4.9±1.2	0.515±0.030	**18.6±1.6**	**0.809±0.010**	2.6±0.8	0.439±0.022
		Clean	5.1±0.4	0.499±0.007	4.9±0.4	0.503±0.008	4.9±0.6	0.493±0.007
	CelebA	Physical	96.2±2.6	0.987±0.009	**99.2±1.1**	**0.995±0.002**	96.8±2.1	0.987±0.004
VGG	CIFAR-10	BadNets	82.8±23.6	0.970±0.017	93.9±4.9	0.987±0.010	**100.0±0.0**	**0.998±0.002**
		Blended	**85.1±27.3**	**0.976±0.022**	89.5±2.4	0.978±0.005	**97.6±2.1**	**0.987±0.005**
		IAD	41.6±38.3	0.851±0.024	43.6±46.5	0.821±0.157	**99.9±0.5**	**0.997±0.003**
		WaNet	11.3±14.4	0.629±0.139	**80.2±9.5**	**0.961±0.019**	**85.0±13.9**	**0.967±0.032**
		Clean	5.0±0.5	0.505±0.009	4.5±0.4	0.479±0.005	5.4±0.7	0.506±0.017
	GTSRB	BadNets	72.9±41.4	0.952±0.043	**100.0±0.0**	**0.999±0.001**	**100.0±0.0**	0.998±0.003
		Blended	59.8±52.8	0.903±0.090	99.9±0.1	0.995±0.004	**100.0±0.0**	**0.999±0.001**
		IAD	40.0±40.5	0.833±0.114	59.4±42.7	0.915±0.086	**90.6±15.4**	**0.983±0.025**
		WaNet	1.0±0.7	0.453±0.110	**98.8±0.9**	**0.997±0.002**	**98.9±2.9**	0.994±0.006
		Clean	4.9±0.6	0.497±0.003	4.9±0.4	0.497±0.004	5.1±0.7	0.506±0.009
	TinyImageNet	BadNets	**57.1±52.2**	0.732±0.365	**63.2±10.2**	**0.943±0.023**	**65.9±14.4**	**0.952±0.018**
		Blended	**20.6±40.3**	0.558±0.286	**24.7±3.8**	0.858±0.030	**31.8±18.5**	**0.901±0.033**
		IAD	18.5±41.0	0.891±0.032	61.3±13.2	0.939±0.022	**84.1±19.8**	**0.975±0.026**
		WaNet	0.3±0.4	0.630±0.158	**26.1±3.1**	**0.836±0.015**	3.2±0.6	0.545±0.051
		Clean	5.1±0.8	0.496±0.007	4.8±1.1	0.502±0.006	4.7±0.5	0.499±0.007
	CelebA	Physical	86.7±7.4	0.961±0.022	**98.4±0.7**	**0.994±0.002**	93.8±2.3	0.981±0.002

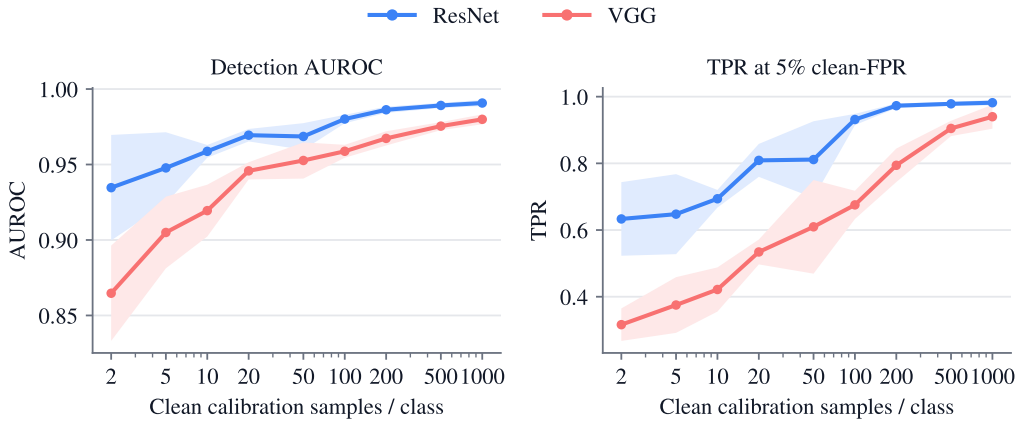


Figure 7: Sensitivity of VARDE to calibration set size on CelebA. Both panels show the mean ± standard deviation over six runs (ResNet and VGG, three seeds each). The clean calibration set is subsampled to the indicated number of inputs per class, from 2 to 1000. The left panel shows the AUROC of the test statistic; the right the TPR at a 5% clean FPR threshold.



Figure 8: One figure for each setting for ResNet. The header names the architecture, dataset, and attack, together with the original label and the predicted class. Below it sits the input and the concept map. The body lists the five highest contributing layer–concept pairs. For each pair, the figure includes the concept's contribution score, exemplar patches from the clean holdout that activate it most strongly, and its concept map for that concept. A pair marked *map n/a* lies at a non-spatial layer, such as a fully connected head, where no spatial concept map is defined. A pair marked *no clean activation* never fires on the clean holdout, so there are no exemplar patches to show for it.

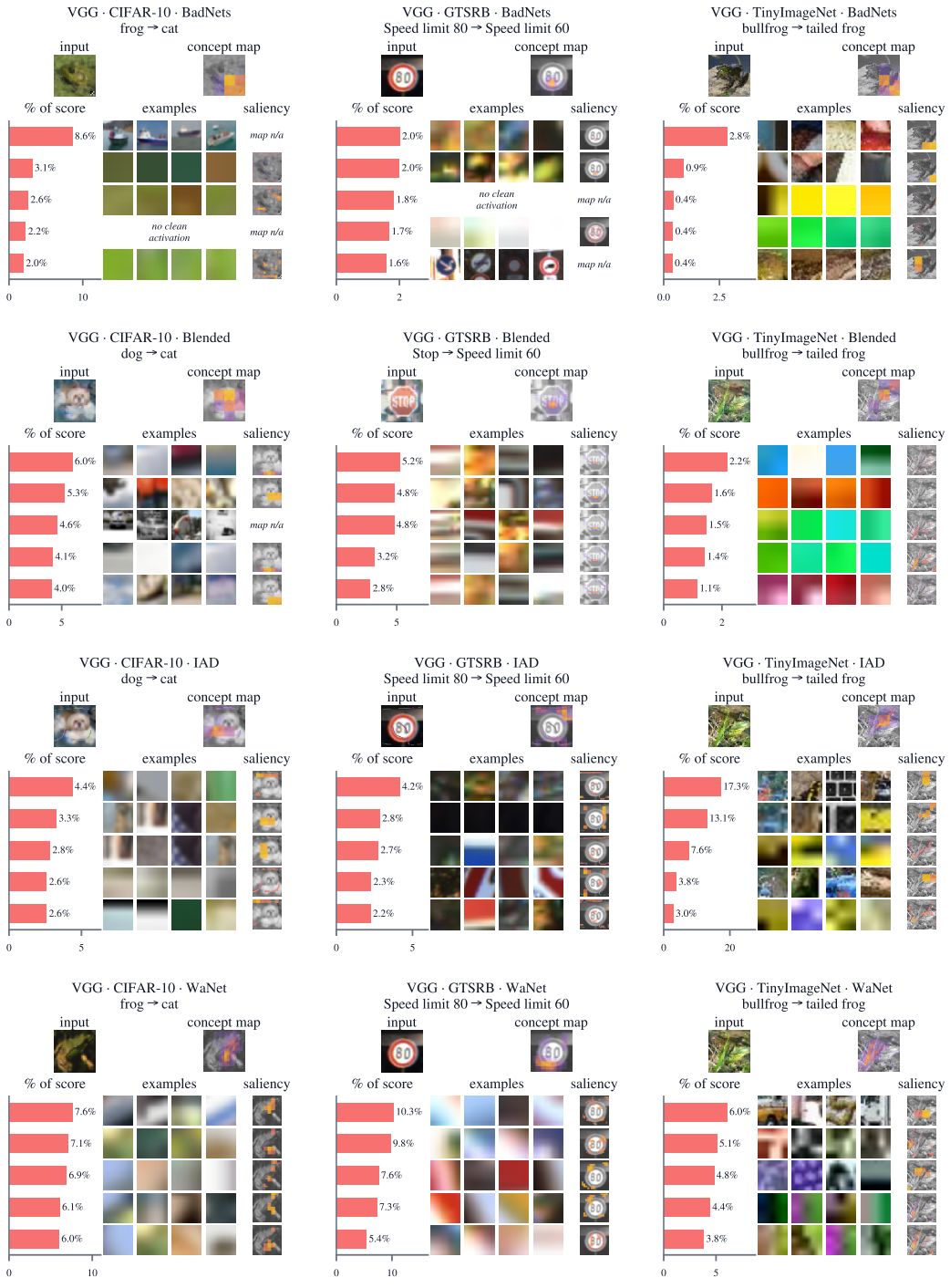


Figure 9: One figure for each setting for VGG. The header names the architecture, dataset, and attack, together with the original label and the predicted class. Below it sits the input and the concept map. The body lists the five highest contributing layer–concept pairs. For each pair, the figure includes the concept's contribution score, exemplar patches from the clean holdout that activate it most strongly, and its concept map for that concept. A pair marked *map n/a* lies at a non-spatial layer, such as a fully connected head, where no spatial concept map is defined. A pair marked *no clean activation* never fires on the clean holdout, so there are no exemplar patches to show for it.

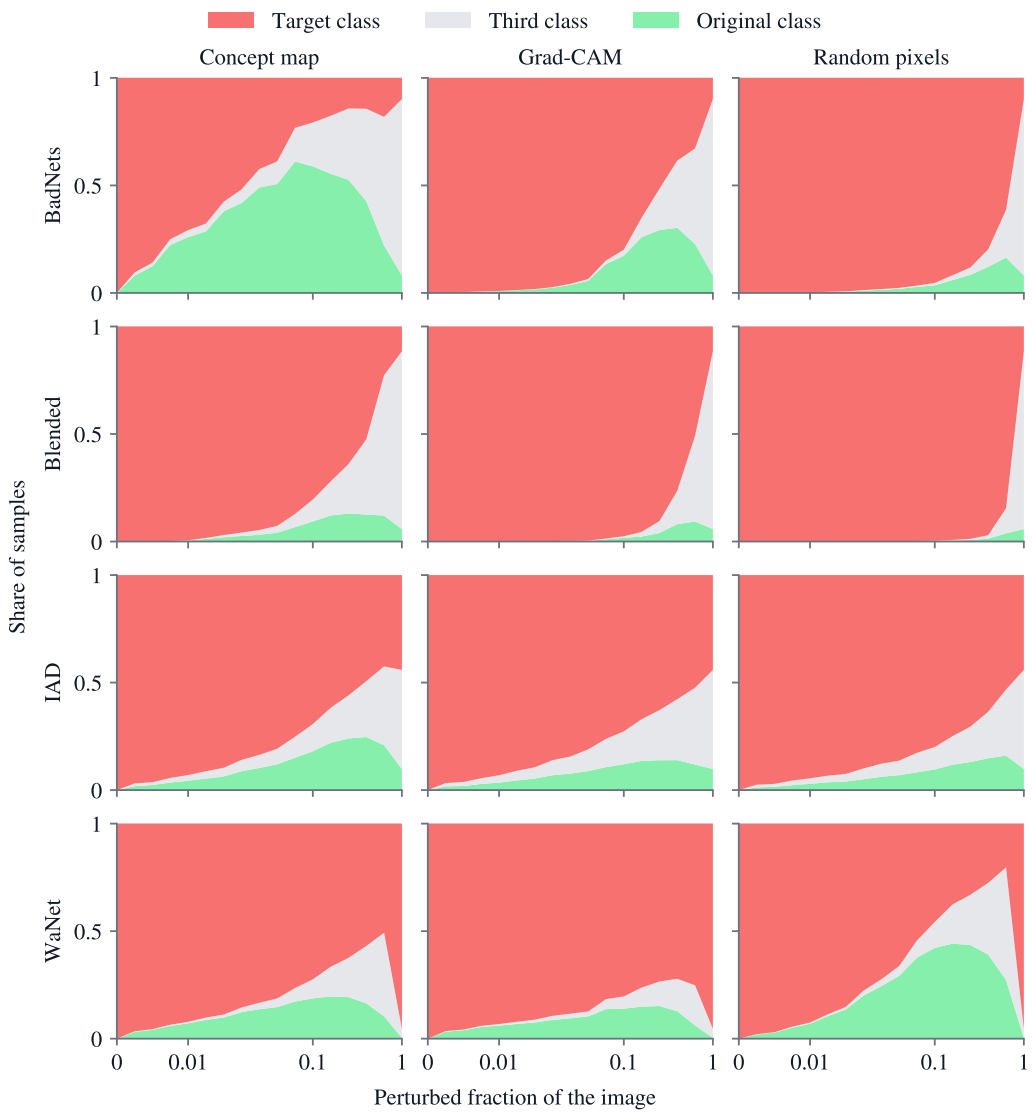


Figure 10: Recovery of the true label as attributed regions are perturbed by applying blur, averaged over all dataset–architecture combinations. The x-axis shows the log-scaled perturbed fraction of the image. For each saliency map, the highest-activated fraction of the image is perturbed, and the backdoored model is re-evaluated. Areas show the share of samples that remain on the attacked class (red), flip to a third class (grey), or return to the original class (green).