CHARACTERISING INPUT LOSS CURVATURE UNDER BACKDOORED NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Backdoor attacks, introduced through data poisoning or architectural manipulation, induce attacker-chosen behaviour on inputs carrying a trigger while leaving clean model performance essentially unchanged. The eigenvalues of a model's input loss Hessian, $\mathbf{H}_{\mathbf{x}}$, measure the local curvature of the loss surface around an input and may therefore contain information about triggers in backdoored networks. We study this through the exact decomposition $\mathbf{H}_{\mathbf{x}} \coloneqq \mathbf{G} + \mathbf{R}$, where $\mathbf{G}$ is the generalised Gauss-Newton term and $\mathbf{R}$ is a model-curvature correction. For cross-entropy classification, we derive bounds showing that output-space curvature collapses as prediction confidence approaches unity. In locally affine networks, $\mathbf{R} = 0$ almost everywhere, so input curvature also collapses if the model Jacobian remains bounded. We further show that locally affine additive backdoors only change $\mathbf{G}$, whereas multiplicative composition introduces an indefinite residual term of rank at most two. We validate these results experimentally using ResNet-18 image classification models with ReLU and SiLU activations. For additive backdoors, we observe eigenvalue collapse under ReLU activations that does not persist when SiLU is used, and multiplicative backdoors produce large model-curvature corrections. Overall, these results show that input curvature depends jointly on backdoor construction, model confidence, model architecture, and loss function.

1 INTRODUCTION

The geometry of neural-network loss landscapes has become an important object of study for understanding optimisation, generalisation, and robustness. Most work considers the loss as a function of the model parameters, using the *parameter Hessian* $\nabla^2_\theta \mathcal{L}$ to study the structure of trained solutions and the concentration of learning in a small number of high-curvature directions (Sagun et al., 2018; Gur-Ari et al., 2018; Ghorbani et al., 2019). Empirically, parameter-Hessian spectra often contain a large bulk of near-zero eigenvalues together with a much smaller set of outliers (Sagun et al., 2017; 2018; Papyan, 2018).

A different local geometry is obtained by fixing the model and perturbing its input. For a trained model $\mathbf{f}_\theta$, input $\mathbf{x}$, label $\mathbf{y}$, and loss ℓ, the *input Hessian* $\mathbf{H}_{\mathbf{x}} = \nabla^2_{\mathbf{x}} \ell(\mathbf{f}_\theta(\mathbf{x}), \mathbf{y})$ describes the second-order behaviour of the loss around an individual input. Input-space curvature has been connected to adversarial robustness and used as a regularisation target (Moosavi-Dezfooli et al., 2019; Mustafa et al., 2020). Singla et al. (2019) derive the input Hessian of deep ReLU classifiers under cross-entropy and study its high-confidence structure, while more recent work has explored Hessian-derived quantities for out-of-distribution inputs (Park et al., 2026). These results raise a basic interpretive question: *What underlying properties are reflected in an observed input-Hessian spectrum?* Since the input Hessian is composed of both the model and the loss function, changes in curvature may arise from the model's local sensitivity, its output distribution, the loss used to evaluate it, or second-order nonlinearities in the model itself.

We investigate this question using backdoored neural networks, which provide a controlled intervention — the same input is evaluated with and without a trigger, allowing curvature changes to be attributed to specific components of the decomposition. A backdoor is hidden functionality activated by a trigger to induce attacker-chosen behaviour, while maintaining utility on untriggered inputs. In a data-poisoning backdoor, triggered training examples are typically relabelled so that the malicious

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.

association is learned in the model parameters (Gu et al., 2019). Backdoors can also be introduced directly into model weights (Hong et al., 2022) or into the model architecture (Bober-Irizar et al., 2023; Langford et al., 2025). These mechanisms are useful experimental probes because similar trigger-dependent behaviour can arise through fundamentally different modifications to the learned system: poisoning changes the learned parameters, whereas an architectural backdoor changes the computation performed by the model.

Previous work has connected backdooring with loss-landscape geometry, primarily in parameter space. Data-poisoned models have been reported to occupy sharper parameter-space regions, motivating curvature-aware backdoor-removal procedures (Karim et al., 2023), while handcrafted weight backdoors exhibit qualitatively different parameter-Hessian behaviour (Hong et al., 2022). Flynn & Granziol (2025) provide complementary theory for how poisoning becomes encoded in high-dimensional predictors. Granziol (2026) proves, under kernel ridge regression and squared loss, that poisoning induces a rank-one input-Hessian spike, and identifies a regime where curvature vanishes without loss of efficacy. Squared-loss curvature is independent of confidence (see Appendix E.2), whereas cross-entropy depends on confidence; whether an attack is visible in the spectrum therefore depends on the loss. These works focus on poisoning-based mechanisms; here we consider both data-poisoned and architectural backdoors to study how different sources of hidden, trigger-activated behaviour appear in the input loss landscape.

Contributions Our central question is: *Under different model backdooring scenarios, how do we interpret the input Hessian spectrum?* We show that additive and multiplicative architectural backdoor composition produces structurally different input curvature when the clean model and the backdoor function are individually locally affine. Additive composition leaves the model-curvature term exactly zero, whereas multiplicative composition introduces an indefinite correction of rank at most two, with eigenpairs available in closed form (Proposition 5). Additionally, we make the confidence–curvature relationship quantitative: Corollary 3 gives two-sided bounds on the cross-entropy loss-Hessian curvature, and Proposition 6 transports these into the input Hessian, yielding a directly testable predictor of the leading eigenvalue. Taken together, these results show that the input-Hessian spectrum reflects prediction confidence, Jacobian scale and model smoothness rather than a signature of backdooring. We validate these findings through experiments on ResNet-18 models with ReLU and SiLU activations.

2 THEORY

We consider multi-class classification and study the curvature of the loss at a single input. Section 2.1 decomposes the input Hessian into a term carrying the curvature of the loss and a term carrying the curvature of the model; Section 2.2 then computes the first of these for cross-entropy.

2.1 DECOMPOSITION OF $\mathbf{H_x}$

Let $\mathbf{f}_\theta : \mathbb{R}^n \to \mathbb{R}^C$ be the model with parameters θ, input $\mathbf{x} \in \mathbb{R}^n$ and output $\mathbf{z} = \mathbf{f}_\theta(\mathbf{x}) \in \mathbb{R}^C$, where C is the number of classes, and let $\ell(\cdot, \mathbf{y}) : \mathbb{R}^C \to \mathbb{R}$ be a scalar loss evaluated against a fixed label $\mathbf{y}$. Writing z_i for the ith component of $\mathbf{z}$, we define

$$\mathbf{H_z} = \nabla^2_{\mathbf{z}} \ell \in \mathbb{R}^{C\times C}, \qquad \mathbf{J} = \frac{\partial \mathbf{z}}{\partial \mathbf{x}} \in \mathbb{R}^{C\times n}, \qquad \mathbf{H}_i = \nabla^2_{\mathbf{x}} z_i \in \mathbb{R}^{n\times n}.$$

By the second-order chain rule, wherever the required derivatives exist,

$$\mathbf{H_x} = \mathbf{J}^\top \mathbf{H_z} \mathbf{J} + \sum_{i=1}^{C} \frac{\partial \ell}{\partial z_i} \mathbf{H}_i, \tag{1}$$

as shown in Appendix B. We write

$$\mathbf{H_x} \coloneqq \mathbf{G} + \mathbf{R}, \text{ where } \mathbf{G} \coloneqq \mathbf{J}^\top \mathbf{H_z} \mathbf{J}, \text{ and } \mathbf{R} \coloneqq \sum_i \frac{\partial \ell}{\partial z_i} \mathbf{H}_i.$$

The first term is the generalised Gauss-Newton (GGN) matrix (Schraudolph, 2002), which is widely used as a curvature approximation in parameter space (Martens, 2020; Botev et al., 2017); here we apply this decomposition in input space.

Interpretation The GGN term $\mathbf{G}$ transports the curvature of ℓ in output space back through the model's local linearisation. The second term $\mathbf{R}$, referred to throughout as the model-curvature correction, captures curvature of the model outputs weighted by the loss gradient. It vanishes when $\mathbf{H}_i = \mathbf{0}$ for every i, or when $\nabla_{\mathbf{z}}\ell = \mathbf{0}$. More generally, a small loss gradient implies a small model-curvature correction, provided the model-output Hessians remain bounded; Appendix C explores this for cross-entropy loss.

Remark 1 (Piecewise-affine). For a network which is piecewise affine in $\mathbf{x}$, $\mathbf{H}_i = \mathbf{0}$. For example, a ReLU network at evaluation time is piecewise affine in $\mathbf{x}$ almost everywhere (everywhere except on the measure-zero set of activation boundaries, where the network may fail to be differentiable). Under these conditions the second term of (1) vanishes and $\mathbf{H}_{\mathbf{x}} = \mathbf{G}$. This is not true for smooth activations such as GeLU (Hendrycks & Gimpel, 2016) or SiLU (Elfwing et al., 2018), or multiplicatively composed models.

Remark 2 (Non-piecewise-affine). Suppose $\mathbf{R} \neq \mathbf{0}$ and order the eigenvalues of $\mathbf{H}_{\mathbf{x}}$ and $\mathbf{G}$ non-increasingly, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. Since both $\mathbf{G}$ and $\mathbf{R}$ are symmetric, Weyl's inequality (Horn & Johnson, 2012) gives, for every $k = 1, \ldots, n$,

$$\left|\lambda_k(\mathbf{H}_{\mathbf{x}}) - \lambda_k(\mathbf{G})\right| \leq \|\mathbf{R}\|_2. \tag{2}$$

Thus the GGN spectrum remains representative whenever $\|\mathbf{R}\|_2$ is small. This is shown for cross-entropy loss in Appendix C. Further, when $\mathbf{G} \succeq 0$,

$$\lambda_{\min}(\mathbf{H}_{\mathbf{x}}) \geq \lambda_{\min}(\mathbf{G}) - \|\mathbf{R}\|_2 \geq -\|\mathbf{R}\|_2.$$

Therefore, any negative curvature in the full input Hessian is bounded in magnitude by $\|\mathbf{R}\|_2$, and vanishes entirely when $\mathbf{R} = \mathbf{0}$.

2.2 Loss-dependent curvature

The loss enters the input Hessian through two quantities: $\mathbf{H}_{\mathbf{z}}$, which determines $\mathbf{G} = \mathbf{J}^\top \mathbf{H}_{\mathbf{z}} \mathbf{J}$, and $\nabla_{\mathbf{z}}\ell$, which weights the model-output Hessians in $\mathbf{R}$. Proposition 6 (Appendix D) bounds the rank and leading eigenvalue of $\mathbf{G}$ in terms of those of $\mathbf{H}_{\mathbf{z}}$, up to factors of the Jacobian. To understand the spectral properties of $\mathbf{G}$, we must understand those of $\mathbf{H}_{\mathbf{z}}$. If we consider cross-entropy loss, we have the following corollary to Proposition 6.

Corollary 3 (Cross-entropy curvature)**.** *For cross-entropy loss, where* $\mathbf{p} = \mathrm{softmax}(\mathbf{z})$*,*

$$\mathbf{H}_{\mathbf{z}} = \mathbf{\Sigma} \coloneqq \mathrm{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top,$$

where $\mathbf{\Sigma} \succeq 0$ *and* $\mathrm{rank}(\mathbf{\Sigma}) = C - 1$*. Hence, by Proposition 6,*

$$\mathrm{rank}(\mathbf{G}) \leq C - 1. \tag{3}$$

Writing $\hat{p} = \max_i p_i$*, the leading output-space curvature satisfies*

$$\frac{C}{C-1}\hat{p}(1-\hat{p}) \leq \lambda_{\max}(\mathbf{\Sigma}) \leq 2\hat{p}(1-\hat{p}), \tag{4}$$

noting that $\lambda_{\max}(\mathbf{\Sigma}) \to 0$ *as* $\hat{p} \to 1$*.*

The form $\mathbf{H}_{\mathbf{z}} = \mathbf{\Sigma}$ and its rank also appear in Singla et al. (2019). The proof is in Appendix E.1, together with Corollary 8, which gives the corresponding spectral-transfer bound: $\lambda_k(\mathbf{G}) \leq \sigma^2_{\max}(\mathbf{J})\lambda_k(\mathbf{\Sigma})$ for every k. Since $\mathbf{\Sigma}$ depends only on the predicted probabilities, $\mathbf{G}$ is label-independent, and the target enters the input Hessian only through $\mathbf{R}$.

This is more fully understood when considering two ways the residual probability can be distributed across classes. When the probability mass concentrates on the top two classes a and b, $\mathbf{\Sigma}$ is approximately rank one and

$$\lambda_{\max}(\mathbf{G}) \simeq p_a p_b \|\nabla_{\mathbf{x}}(z_b - z_a)\|_2^2.$$

When instead the residual mass is distributed uniformly, $\mathbf{\Sigma}$ has leading eigenvector proportional to $\mathbf{e}_a - \frac{1}{C-1}\sum_{j\neq a}\mathbf{e}_j$ and

$$\lambda_{\max}(\mathbf{G}) \geq \hat{p}(1-\hat{p})\|\mathbf{g}_a\|_2^2, \qquad \mathbf{g}_a \coloneqq \nabla_{\mathbf{x}}\left[z_a - \frac{1}{C-1}\sum_{j\neq a} z_j\right], \tag{5}$$

with approximation error bounded by $\frac{1-\hat{p}}{C-1}\|\mathbf{J}\|_2^2$. Thus, if $\|\mathbf{J}\|_2$ remains bounded, the absolute approximation error vanishes as $\hat{p} \to 1$.

For a piecewise-affine model, $\mathbf{R} = \mathbf{0}$, so if $\|\mathbf{J}\|_2$ remains bounded,

$$0 \leq \lambda_{\max}(\mathbf{H}_\mathbf{x}) \leq 2\hat{p}(1-\hat{p})\|\mathbf{J}\|_2^2 \longrightarrow 0 \quad \text{as } \hat{p} \to 1, \tag{6}$$

meaning that all input-Hessian eigenvalues tend to zero in the high-confidence limit.

Other loss functions. A similar analysis can be carried out to understand the different curvature produced by other common loss functions. For multiclass exponential loss, the largest eigenvalue of $\mathbf{H}_\mathbf{z}$ depends on the target label: as confidence approaches one, the curvature vanishes for a correct prediction but grows for an incorrect prediction. For mean squared error, if $C > 0$, $\mathbf{H}_\mathbf{z}$ is full rank and all eigenvalues are equal. The linear hinge loss has $\mathbf{H}_\mathbf{z} = \mathbf{0}$ almost everywhere, whereas the rank and eigenvalues of the squared-hinge loss curvature depend on the number of violated margins. Full statements and derivations can be found in Appendix E.2.

3 Backdoor curvature

The decomposition $\mathbf{H}_\mathbf{x} = \mathbf{G} + \mathbf{R}$ distinguishes three ways in which a backdoor can alter input curvature. First, the logits at which the loss derivatives are calculated can change. Second, it can change the model Jacobian, and hence the GGN term $\mathbf{G}$. Third, it may introduce curvature into the model map itself, thus changing $\mathbf{R}$.

First, we consider the case where both the clean model $\mathbf{f}_\theta$ and the trigger-activation function $t : \mathbb{R}^n \to \mathbb{R}$ are affine in a neighbourhood of the input $\mathbf{x}$. Throughout, $\nabla_\mathbf{z}\ell$ and $\mathbf{H}_\mathbf{z}$ are evaluated at the logits produced by the modified model, $\mathbf{z} = \widetilde{\mathbf{f}}_\theta(\mathbf{x})$. We consider two ways of combining the trigger-activation function with the clean model to make the modified function; we describe these in turn.

Proposition 4 (Additive targeted architectural backdoor)**.** *Consider the modified model*

$$\widetilde{\mathbf{f}}(\mathbf{x}) = \mathbf{f}_\theta(\mathbf{x}) + t(\mathbf{x})\mathbf{e}_\tau,$$

where $\mathbf{e}_\tau$ selects the target class. Then

$$\widetilde{\mathbf{J}} = \mathbf{J} + \mathbf{e}_\tau(\nabla_\mathbf{x} t)^\top, \qquad \mathbf{R}_{\text{add}} = \mathbf{0}, \qquad \mathbf{H}_{\mathbf{x},\text{add}} = \mathbf{G}_{\text{add}} \coloneqq \widetilde{\mathbf{J}}^\top \mathbf{H}_\mathbf{z} \widetilde{\mathbf{J}}.$$

Therefore, an additive backdoor can alter input curvature through the logits and Jacobian of the modified model, but it does not introduce a model-curvature correction term.

Throughout, we refer to this as an additive backdoor for ease.

Proposition 5 (Multiplicative untargeted architectural backdoor (Langford et al., 2025))**.** *Consider the modified model*

$$\widetilde{\mathbf{f}}(\mathbf{x}) = (1 - t(\mathbf{x}))\mathbf{f}_\theta(\mathbf{x}).$$

Define $\mathbf{g}_f \coloneqq \mathbf{J}^\top \nabla_\mathbf{z}\ell$. Then $\mathbf{H}_{\mathbf{x},\text{mult}} = \mathbf{G}_{\text{mult}} + \mathbf{R}_{\text{mult}}$, with

$$\widetilde{\mathbf{J}} = (1-t)\mathbf{J} - \mathbf{f}_\theta(\nabla_\mathbf{x} t)^\top, \mathbf{R}_{\text{mult}} = -\left[\nabla_\mathbf{x} t\, \mathbf{g}_f^\top + \mathbf{g}_f(\nabla_\mathbf{x} t)^\top\right], \tag{7}$$

where $\mathbf{G}_{\text{mult}} \coloneqq \widetilde{\mathbf{J}}^\top \mathbf{H}_\mathbf{z} \widetilde{\mathbf{J}}$. Consequently,

$$\operatorname{rank}(\mathbf{R}_{\text{mult}}) \leq 2, \qquad \operatorname{range}(\mathbf{R}_{\text{mult}}) \subseteq \operatorname{span}\{\nabla_\mathbf{x} t, \mathbf{g}_f\}. \tag{8}$$

If $\nabla_\mathbf{x} t$ and $\mathbf{g}_f$ are nonzero and non-collinear, the two nonzero eigenvalues are

$$\lambda_\pm(\mathbf{R}_{\text{mult}}) = -(\nabla_\mathbf{x} t)^\top \mathbf{g}_f \pm \|\nabla_\mathbf{x} t\|_2 \|\mathbf{g}_f\|_2.$$

Hence, $\mathbf{R}_{\text{mult}}$ has one positive and one negative eigenvalue. The corresponding eigenvectors lie in $\operatorname{span}\{\nabla_\mathbf{x} t, \mathbf{g}_f\}$ and are given explicitly in Appendix F.2.

The proof can be found in Appendix F.2. Throughout, we refer to this as a multiplicative backdoor for ease. A backdoor that introduces a multiplicative composition of functions fundamentally differs from additive functions, as even though both the clean model and the trigger are individually locally affine, their product is locally quadratic. In both cases, the trigger-activation function alters the GGN term through the modified model's Jacobian; however, the multiplicative case additionally introduces an extra model-curvature term of rank at most 2. For both types of backdoor, to understand how the trigger can affect the eigenspectra of the GGN term, we suppose that the loss Hessian

at the logits of the modified model is approximately rank one,

$$\mathbf{H}_{\mathbf{z}} \approx \omega \mathbf{a}\mathbf{a}^\top, \qquad \|\mathbf{a}\|_2 = 1,$$

where $\mathbf{a} \in \mathbb{R}^C$ is its leading eigenvector and $\omega \geq 0$ the corresponding eigenvalue. Thus, the GGN term can be described as $\mathbf{G} \approx \omega \mathbf{q}\mathbf{q}^\top$, where $\mathbf{q}$ is defined by either $\mathbf{q}_{\text{add}}$ or $\mathbf{q}_{\text{mult}}$ for each construction,

$$\mathbf{q}_{\text{add}} = \mathbf{J}^\top \mathbf{a} + (\mathbf{e}_\tau^\top \mathbf{a}) \nabla_{\mathbf{x}} t, \qquad \mathbf{q}_{\text{mult}} = (1-t)\mathbf{J}^\top \mathbf{a} - (\mathbf{f}_\theta^\top \mathbf{a}) \nabla_{\mathbf{x}} t.$$

For $\mathbf{q} \neq \mathbf{0}$, the eigenpair is approximately:

$$\lambda_{\max}(\mathbf{G}) \approx \omega \|\mathbf{q}\|_2^2, \qquad \mathbf{v}_{\max}(\mathbf{G}) \approx \frac{\mathbf{q}}{\|\mathbf{q}\|_2}.$$

Therefore, the additive backdoor introduces a term along $\nabla_{\mathbf{x}} t$, whereas the multiplicative trigger both rescales the clean-path direction and introduces a trigger-gradient term. If this trigger-gradient component dominates the clean-path contribution to $\mathbf{q}$, then $\mathbf{v}_{\max}(\mathbf{G})$ aligns, up to sign, with $\nabla_{\mathbf{x}} t$. The leading curvature direction may therefore reveal the input direction to which the trigger activation is most sensitive. The trigger may also change ω and $\mathbf{a}$, since $\mathbf{H}_{\mathbf{z}}$ is evaluated at the modified logits.

Nonlinear models and trigger-activation functions. Thus far, we have only considered cases where both the model and the trigger-activation function are locally affine. However, there are cases in which either (or both) may not be. The additive case can have a nonzero $\mathbf{R}$ term. The complete $\mathbf{R}$ term in the multiplicative case may not be low rank. The additional terms from these settings contain the Hessians of the logits and t, where the precise form can be found in Appendix F.3.

Data-poisoned backdoors. Data poisoning differs from architectural backdoors in that there is no structural change to $\mathbf{f}_\theta$; instead, data poisoning changes the learned parameters θ. For example, a ReLU model trained on poisoned data will still be locally linear and $\mathbf{H}_{\mathbf{x}} = \mathbf{G} = \mathbf{J}^\top \mathbf{H}_{\mathbf{z}} \mathbf{J}$. Any change to the spectra arises through changes to the model predictions, the Jacobian, or both, and no model-curvature term appears directly from the poisoning. A successful poisoning is likely to increase the confidence in an incorrect prediction (either the target class or simply a wrong class, depending on the attacker's goal), and results from Proposition 6 can be used appropriately.

Interaction with loss curvature. The input-Hessian spectrum is determined by the interplay between the loss and the backdoor. Therefore, to understand the effect of a backdoor on input curvature, we must also consider the loss used. For cross-entropy, increasing confidence drives $\mathbf{H}_{\mathbf{z}}$, and hence $\mathbf{G}$, towards zero, provided $\|\widetilde{\mathbf{J}}\|_2$ remains bounded. In the locally affine additive case, $\mathbf{R}_{\text{add}} = \mathbf{0}$, so all input-Hessian eigenvalues tend to zero. For the multiplicative backdoor, $\mathbf{R}_{\text{mult}}$ also depends on $\nabla_{\mathbf{z}} \ell$. For a confidently correct prediction, $\nabla_{\mathbf{z}} \ell$ tends towards zero, making $\mathbf{R}_{\text{mult}}$ also vanish, provided $\mathbf{J}$ and $\nabla_{\mathbf{x}} t$ remain bounded. On the other hand, the loss gradient need not vanish for a confidently misclassified input, such as a successfully attacked triggered input. In this case, $\mathbf{G}_{\text{mult}}$ may collapse while the low-rank indefinite correction $\mathbf{R}_{\text{mult}}$ remains nonzero and dominates the input-Hessian spectrum.

4 Empirical Experimentation

For the clean baseline set of classification models, we trained five ResNet-18s (He et al., 2016) over different random seeds on the CIFAR-10 training split (Krizhevsky, 2009), with Adam optimiser (Kingma & Ba, 2015), using the cross-entropy loss. All models were trained on the Isambard-AI high-performance computing (HPC) system (McIntosh-Smith et al., 2024) (additional details in Appendix G). The model accuracies and attack success rates are reported in Appendix G.3, Table 2. We implemented the *additive architectural backdoor* and *multiplicative architectural backdoor* using the weights of the clean ResNet-18 models and trained five further ResNet-18 models using a simple *dataset poisoning* methodology. We used a simple 3×3 checkerboard patch trigger placed in the bottom-left corner of each image. Full details about the implementation of the backdoors can be found in Appendix G.2.

To measure the input curvature, we studied the eigenvalues of the input Hessian $\mathbf{H}_{\mathbf{x}}$. In place of calculating the eigenvalues directly from the Hessian, we approximated the eigenvalues of $\mathbf{H}_{\mathbf{x}}$ using the Lanczos algorithm (Lanczos, 1950) over 10 iterations (see Appendix J.5). Alongside curvature

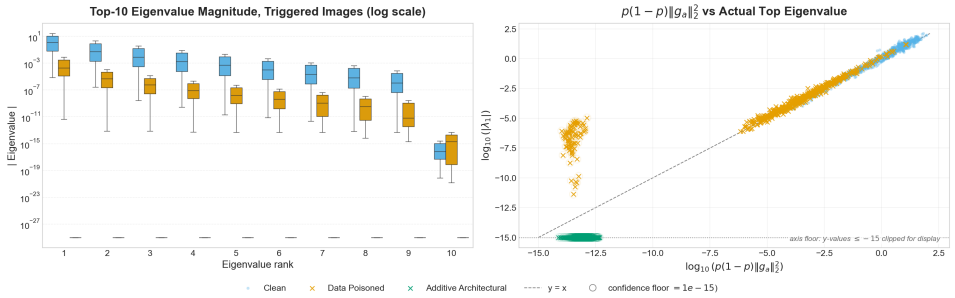


Figure 1: **Left:** Magnitude of the top ten eigenvalues for the clean (blue), data poisoned (yellow) and additive architecturally backdoored (green) models, evaluated on triggered CIFAR-10 images. **Right:** $\hat{p}(1-\hat{p})\,\|\mathbf{g}_a\|_2^2$ vs $\lambda_{\max}(\mathbf{G})$ for the triggered images and clean (blue), data poisoned (yellow) and additive architecturally backdoored (green) models.

metrics, we collected prediction confidence (from the softmax on $\mathbf{z}$), model accuracy and the spectral norm $\|\mathbf{J}\|_2^2$ of the model Jacobian.

4.1 Piecewise affine models and the cross-entropy loss

We first consider the case where the model $\mathbf{f}_\theta$ is locally affine, and the backdoored model $\widetilde{\mathbf{f}}_\theta$ is also locally affine. Therefore, by Remark 1 and Proposition 4, $\mathbf{R} = \mathbf{0}$ and $\mathbf{H}_\mathbf{x} = \mathbf{G}$, where the Jacobian is that of the model being evaluated. We can then apply the cross-entropy results in Section 2.2 directly. Hence, $\mathbf{G}$ is independent of the label, $\operatorname{rank}(\mathbf{H}_\mathbf{x}) \leq C-1$, and if the model Jacobian is bounded, the input-Hessian spectrum vanishes as $\hat{p} \to 1$.

To illustrate these effects, we perform an empirical study comparing clean ResNet-18 models, data-poisoned models and additive architecturally backdoored models. Specifically, we explore whether the eigenvalues of the input-Hessian are zero, as predicted by Equation (6), and whether the value $\hat{p}(1-\hat{p})\|\mathbf{g}_a\|_2^2$ from Equation (5) tracks the observed value $\lambda_{\max}(\mathbf{G})$. We repeated these experiments with AlexNet models (Krizhevsky et al., 2012) (trained on CIFAR-10) and ResNet-50 models (trained on Caltech-256 (Griffin et al., 2007)) and found similar results, illustrating that these findings extend beyond ResNet-18. See Appendix H and Appendix I.2 for details.

We found that, when evaluated on triggered images, the backdoored models had a flat local input loss curvature. The additive architectural backdoor and the data-poisoned models had much smaller values among the ten leading eigenvalues when compared to the clean ResNet-18s. Table 1 reports the magnitude of the largest eigenvalue λ_1 for the models trained with cross-entropy and Figure 1 shows the eigenvalues of a clean model (blue) compared to the backdoored models (yellow and green). In all cases there were $C-1=9$ above-zero eigenvalues, explained by Corollary 3, but the backdoors drastically decrease their magnitude.

Under the triggered images and backdoored models, the average prediction confidence increased to nearly 100%, and the largest eigenvalues decreased to near zero. The increase in prediction confidence and the decrease in the largest eigenvalue support our earlier claim that a highly confident backdoor prediction drives the largest input-Hessian eigenvalue towards zero (Corollary 3). Figure 1 illustrates the linear dependence between $\hat{p}(1-\hat{p})\,\|\mathbf{g}_a\|_2^2$ and $\lambda_{\max}(\mathbf{G})$, demonstrating how the prediction confidence, $\hat{p}$, modulated by a gradient factor, directly affects $\lambda_{\max}$. Table 1 also reports the gradient terms ($\|\mathbf{g}_a\|$ and $\|\mathbf{J}\|_2^2$) for all three models. The gradients for the additive architectural backdoors on both clean images and triggered images are similar to the clean ResNet-18s, illustrating that the backdoors have very little impact on the model Jacobian. We observed a slight decrease in $\|\mathbf{g}_a\|$ and $\|\mathbf{J}\|_2^2$ for the data poisoned models for both clean and triggered images.

4.2 Non-piecewise-affine models and untargeted backdoors

Section 4.1 showed that the curvature changes observed for locally affine models can be explained through the confidence-dependent GGN term. We now ask how this picture changes when the model

Table 1: ResNet-18 models trained with cross-entropy loss: average leading eigenvalue $\lambda_{\max}$, average $\|\mathbf{g}_a\|_2$, and average $\|\mathbf{J}\|_2^2$, per backdoor type and architecture, split by clean vs. triggered images. All values are mean $\pm$ std. dev. over pooled model–image observations. The same 100 CIFAR-10 test images are evaluated with each of the 5 models.

Model	**Images**	$\lambda_{\max}$	$\|\mathbf{g}_a\|_2$	$\|\mathbf{J}\|_2^2$
Clean (ReLU)	Clean	7.92 ± 14.1	9.48 ± 3.43	12.7 ± 4.25
	Triggered	8.10 ± 13.9	9.36 ± 3.37	12.5 ± 4.17
Data Poisoned (ReLU)	Clean	6.90 ± 11.1	7.33 ± 2.62	9.64 ± 3.25
	Triggered	0.142 ± 1.08	5.58 ± 1.72	7.11 ± 2.02
Additive arch. (ReLU)	Clean	7.92 ± 14.1	9.56 ± 3.35	12.7 ± 4.17
	Triggered	$(3.16 \pm 26.7) \times 10^{-30}$	9.87 ± 3.22	14.0 ± 3.09

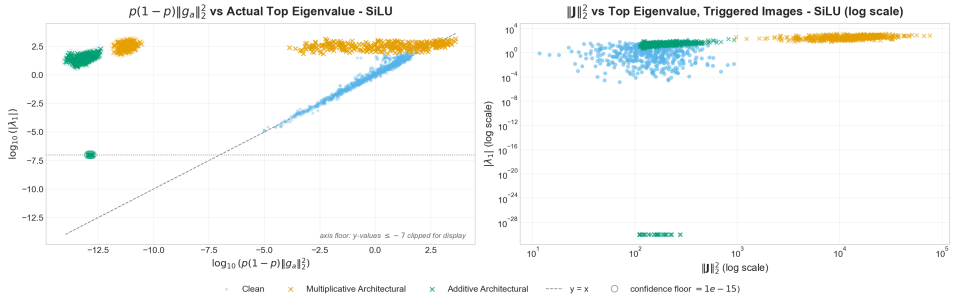


Figure 2: A comparison of curvature and gradient terms under clean (blue), additive architectural (green) and multiplicative architectural (yellow) backdoors in SiLU ResNet-18. **Left:** $\hat{p}(1-\hat{p})\,\|\mathbf{g}_a\|_2^2$ vs $\lambda_{\max}(\mathbf{H}_\mathbf{x})$. **Right:** $\|\mathbf{J}\|_2^2$ vs $\lambda_{\max}(\mathbf{H}_\mathbf{x})$.

itself has non-zero second derivatives, allowing the model-curvature term ($\mathbf{R}$) to contribute to the input Hessian. We study two ways this can arise: replacing ReLU with smooth activation functions, and introducing a multiplicative architectural backdoor that makes an otherwise piecewise-affine model locally nonlinear. These settings allow us to test whether departures from the confidence–curvature relationship can be explained by model curvature.

We first replace the ReLU activations of the ResNet-18 with SiLU activation functions, while otherwise retaining the experimental setup of the previous section. Figure 2 shows the relationship between $\hat{p}(1-\hat{p})\|\mathbf{g}_a\|_2^2$ and the leading eigenvalue, $\lambda_{\max}$, for these models. In contrast to the approximately linear trend observed for ReLU networks, the SiLU models exhibit a distinct branch along which the leading eigenvalue remains elevated despite substantial changes in the confidence factor. The curvature is therefore no longer explained by confidence alone. This behaviour is consistent with the model-curvature correction $\mathbf{R}$ becoming sufficiently large to dominate the GGN contribution for a subset of inputs. We repeat these experiments with a tanh activation function and find similar patterns. These are reported in Appendix I.

We test this explanation directly in Figure 3. For cross-entropy, $\nabla_\mathbf{z}\ell = \mathbf{p} - \mathbf{y}$, and hence $\mathbf{R}$ is weighted by the difference between the predicted probabilities and the label used to evaluate the loss. When we instead evaluate a confidently backdoored input using the backdoor target as the label, $\mathbf{p} - \mathbf{y} \approx \mathbf{0}$, strongly suppressing $\mathbf{R}$. The additional branch disappears and the points return to the confidence-dependent trend. This provides direct empirical evidence that the departure from the piecewise-affine relationship is caused by the model-curvature correction rather than by the confidence-dependent GGN term.

A closely related effect occurs for the untargeted architectural backdoor. Although the underlying ResNet-18 and trigger detector are individually piecewise-affine, the backdoor combines them multiplicatively, so the resulting model is no longer locally affine. As shown in Proposition 5, this composition introduces a non-zero model-curvature correction of rank at most two.

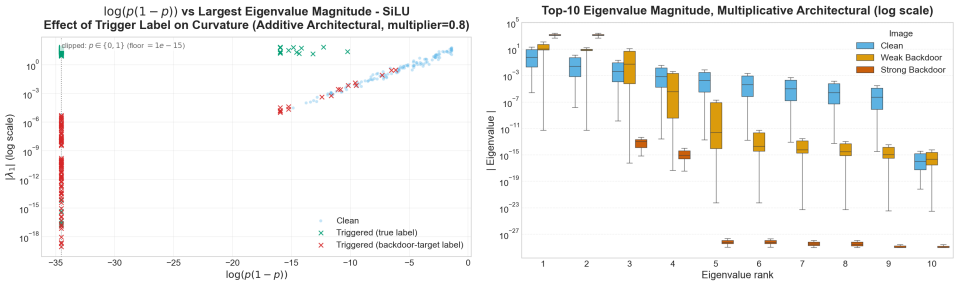


Figure 3: **Left:** $p(1-p)$ vs $\lambda_{\max}$ for SiLU ResNet-18 with additive backdoor, input Hessian evaluated with true labels and target labels. **Right:** The magnitude of the top ten eigenvalues of the multiplicative backdoor at different strengths on ReLU ResNet-18. We compare a weak multiplicative backdoor (with multiplier $m = 0.001$) and a strong multiplicative backdoor (with multiplier $m = 0.1$).

The empirical spectrum reflects this qualitative change. As the untargeted backdoor is strengthened, prediction confidence increases, but, in contrast to the targeted piecewise-affine setting, the leading eigenvalues do not collapse. Instead, the leading spectral gap increases while most of the remaining eigenvalues decrease (Figure 3). This is precisely the behaviour that cannot be explained by the confidence-dependent term alone, and is consistent with a low-rank $\mathbf{R}$ dominating a small number of curvature directions.

Together, these experiments separate two qualitatively different regimes. For piecewise-affine models, the model-curvature correction is absent and cross-entropy curvature is largely explained by prediction confidence. For non-piecewise-affine models, including models made non-linear by the backdoor architecture itself, the additional $\mathbf{R}$ term can dominate individual eigendirections and break this proportionality, though we note this does not occur for models trained and evaluated with exponential loss (see Appendix I.3). The resulting curvature therefore contains information beyond confidence, although this does not by itself establish that the additional curvature is uniquely backdoor-specific.

4.3 Empirical decomposition of input curvature

To isolate the effect of the backdoor mechanism on local input-space curvature, we compare the spectra of $\mathbf{H_x}$, $\mathbf{G}$, and $\mathbf{R} = \mathbf{H_x} - \mathbf{G}$ for additive and multiplicative backdoors with matched prediction-change rates. We refer to $\mathbf{R}$ as the residual curvature because we estimate it by subtracting $\mathbf{G}$ from $\mathbf{H_x}$, rather than by computing the model-output Hessians directly.

For each activation function, ReLU and SiLU, we use five independently trained CIFAR-10 ResNet-18 models and evaluate the same 100 held-out images, with and without the trigger, under the clean model and its additive and multiplicative backdoored variants. Backdoor strengths are selected using a separate calibration set of 1,000 images. For each model–image pair, we estimate the largest and smallest eigenvalues of $\mathbf{H_x}$, $\mathbf{G}$, and $\mathbf{R}$ using 64-bit floating-point Hessian–vector products (Pearlmutter, 1994) and Lanczos iteration. We summarise each spectrum by

$$\rho(\mathbf{A}) = \max\left\{|\lambda_{\max}(\mathbf{A})|, |\lambda_{\min}(\mathbf{A})|\right\}.$$

The comparison uses the medium calibrated settings described in Appendix J.1, which yield prediction-change rates of approximately 79–82%. Figure 4 reports the results.

For ReLU, the additive backdoor reduces the Hessian spectral radius to approximately 0.09 times its clean-input value and gives $\rho(\mathbf{R})/\rho(\mathbf{H_x}) < 10^{-12}$, so $\mathbf{H_x} \simeq \mathbf{G}$. In contrast, the multiplicative backdoor increases the Hessian spectral radius by approximately 7×10^3 and gives $\rho(\mathbf{R})/\rho(\mathbf{H_x}) \simeq 1$, placing the residual curvature on the same spectral scale as the full Hessian. This agrees with the theoretical distinction between additive composition applied to locally affine logits, which introduces no residual curvature, and multiplicative composition, which does.

For SiLU, the additive and multiplicative backdoors increase the Hessian spectral radius by approximately 23 and 5×10^3, respectively, with $\rho(\mathbf{R})/\rho(\mathbf{H_x}) \simeq 1$ in both cases. Thus, additive

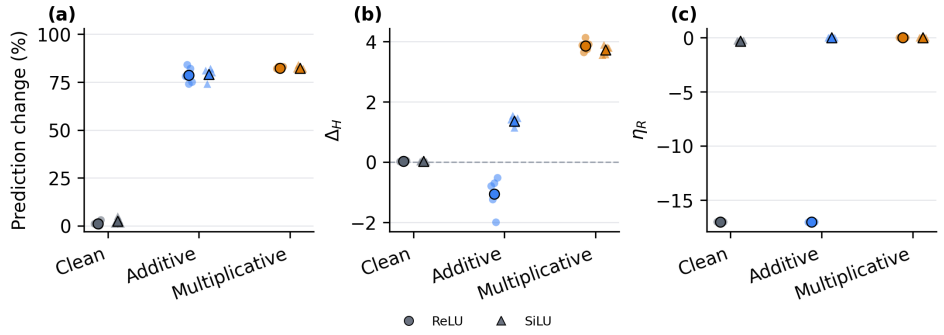


Figure 4: **(a)** Percentage of images whose prediction changes after applying the trigger. **(b)** $\Delta_H = \mathrm{median}_i \log_{10}[\rho(\mathbf{H}_{\mathrm{trig,i}})/\rho(\mathbf{H}_{\mathrm{clean,i}})]$; negative, zero, and positive values respectively mean that triggering decreases, preserves, or increases the Hessian spectral radius. **(c)** $\eta_R = \mathrm{median}_i \log_{10}[\rho(\mathbf{R}_{\mathrm{trig,i}})/\rho(\mathbf{H}_{\mathrm{trig,i}})]$; zero denotes equal spectral radii, while $-k$ means that the residual radius is 10^{-k} times the full Hessian radius. Circles denote ReLU and triangles SiLU. Black-edged markers show the mean across five models.

composition introduces no residual curvature for locally affine logits but can alter existing residual curvature in a smooth model by changing the loss-gradient weights applied to the model-output Hessians. The decomposition therefore depends on both the backdoor mechanism and the local smoothness of the model, even when prediction-change rates are matched. Appendix J.6 repeats the decomposition using AlexNet on CIFAR-10 and ResNet-50 on Caltech-256. Appendix J provides the full numerical results, calibration, data-poisoning comparison, Lanczos convergence tests, and empirical tests of the theoretical predictions for the multiplicative backdoor.

5 Discussion

Our results clarify what changes in input-loss curvature actually represent. We find that the Hessian spectrum does not contain a direct signature of a backdoor, but instead reflects the interaction between the loss geometry and the local geometry of the model through the decomposition $\mathbf{H}_{\mathbf{x}} = \mathbf{G} + \mathbf{R}$. For piecewise-affine models under cross-entropy, the model-curvature term $\mathbf{R}$ vanishes and the Hessian is governed solely by the GGN term, which depends on model confidence: this explains the observed spectral collapse under successful targeted backdoors. In contrast, when the model is intrinsically nonlinear, or when the backdoor introduces nonlinear composition, the residual term $\mathbf{R}$ can dominate the Hessian. Thus, similar trigger-dependent behaviour can produce qualitatively different curvature depending on model linearity.

This distinction is important when interpreting input-Hessian quantities more generally. Backdoors provide a controlled intervention, but alternative perturbations, such as adversarial and out-of-distribution inputs, may similarly change the input-Hessian curvature. Therefore, a spectral deviation may simply arise from the introduction of a perturbation that changes prediction probabilities, the local Jacobian, or the model-curvature correction, rather than because the spectrum identifies a particular underlying phenomenon. Consequently, Hessian-based analyses should control for these factors before interpreting changes in eigenvalue magnitude as specific to poisoning, backdoors, distribution shift, or adversarial behaviour.

Our experiments are restricted primarily to image classifiers and deliberately simple backdoors for which the source of curvature can be isolated analytically. Although the decomposition itself is general, the relative importance of $\mathbf{G}$ and $\mathbf{R}$ may differ for other architectures, modalities, triggers, and losses. We also study local second-order geometry rather than proposing a backdoor detector. An interesting direction is therefore to ask whether information remaining after controlling for confidence and model curvature, including the structure and eigendirections of $\mathbf{G}$ and $\mathbf{R}$ rather than their magnitudes alone, can provide more mechanism-specific signals for detection or attribution.

AI USE STATEMENT

This section discloses our use of generative AI tools in accordance with the ICLR 2027 policy on LLM use. We used ChatGPT and Claude in three areas of this work.

Theoretical results We used AI assistance in deriving some of our theoretical results. Every AI-assisted result was subsequently derived again from scratch, by hand by a separate author; that author both verified the results and strengthened the rigour of the arguments, and the proofs presented in the paper are their own work. We separately checked the statements of these results against the prior literature to confirm that they are not unattributed restatements of known results.

Experiment implementation The experiments in this paper, and their design, are entirely the authors'. Having specified the design ourselves, we provided it to an AI assistant, which helped to write the code implementing our method and experimental pipeline. All AI-generated code was reviewed and tested to ensure correctness.

Writing We used AI tools to draft portions of the manuscript from outlines and notes written by the authors, and to edit author-written text for clarity, grammar, and concision. All such text was reviewed and revised by the authors, and any factual claims or citations introduced during drafting were verified against the primary sources.

We did not use generative AI tools for research ideation: the research questions, hypotheses, and framing of this work are the authors' own. Beyond the uses described above, we did not use them to design or give feedback on our methodology or experiments, generate synthetic data, clean or reformat datasets, assist with translation, conduct qualitative or thematic analysis, or interpret results. We have reviewed all AI-assisted work and take full responsibility for the final content of this paper, including all text, claims, code, and reported results.

ETHICS STATEMENT

This work characterises backdoor attacks on computer vision models in order to assess defences against them, and we acknowledge the dual-use nature of describing and evaluating such attacks. An adversary who controls part of the training data, the architecture definition, or the model weights could apply these techniques to compromise a deployed system, with consequences ranging from degraded accuracy to reliable, attacker-chosen misclassifications. We judge publication to be net-beneficial. Our data-poisoning attacks follow BadNets (Gu et al., 2019) and our architectural attacks follow Langford et al. (2025), whose backdoored models are publicly available[1]. We additionally construct one architectural backdoor of our own, which is a variant within the same design space rather than a new class of attack. Our contribution is the characterisation of these attacks, which is a prerequisite for developing defences. All experiments were run on public benchmarks (CIFAR-10, Caltech-256) and on models we trained ourselves from open-source architectures (ResNet-18, AlexNet and ResNet-50); no third-party or deployed system was attacked. This work involved no human subjects and no personally identifying data. We have not released any of the trained models or modified architecture files in open source repositories. All datasets, pretrained models, and third-party code were used in accordance with their licenses.

REFERENCES

Mikel Bober-Irizar, Ilia Shumailov, Yiren Zhao, Robert Mullins, and Nicolas Papernot. Architectural backdoors in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24595–24604, June 2023.

Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical Gauss-Newton optimisation for deep learning. In *International Conference on Machine Learning*, pp. 557–565. PMLR, 2017.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

[1] https://github.com/Hjel2/backdoor-taxonomy/tree/master

Donald Flynn and Diego Granziol. A linear approach to data poisoning. *arXiv preprint arXiv:2505.15175*, 2025. URL https://arxiv.org/abs/2505.15175.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via Hessian eigenvalue density. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2232–2241. PMLR, 2019.

Diego Granziol. Safety-efficacy trade off: Robustness against data-poisoning. *arXiv preprint arXiv:2602.00822*, 2026. URL https://arxiv.org/abs/2602.00822.

Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical Report CNS-TR-2007-001, California Institute of Technology, March 2007. URL https://resolver.caltech.edu/CaltechAUTHORS:CNS-TR-2007-001. Version: public.

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

Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018. URL https://arxiv.org/abs/1812.04754.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016. URL https://arxiv.org/abs/1606.08415.

Sanghyun Hong, Nicholas Carlini, and Alexey Kurakin. Handcrafted backdoors in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 8068–8080, 2022. doi: 10.52202/068431-0586.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2012.

Nazmul Karim, Abdullah Al Arafat, Umar Khalid, Zhishan Guo, and Naznin Rahnavard. Efficient backdoor removal through natural gradient fine-tuning. *arXiv preprint arXiv:2306.17441*, 2023. URL https://arxiv.org/abs/2306.17441.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, April 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf. CIFAR-10 and CIFAR-100 datasets, Chapter 3.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, pp. 1097–1105, 2012.

Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4): 255–282, 1950.

Harry Langford, Ilia Shumailov, Yiren Zhao, Robert Mullins, and Nicolas Papernot. Architectural neural backdoors from first principles. In *2025 IEEE Symposium on Security and Privacy*, pp. 1657–1675, May 2025. doi: 10.1109/SP61157.2025.00060.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

Simon McIntosh-Smith, Sadaf R. Alam, and Christopher Woods. Isambard-AI: a leadership class supercomputer optimised specifically for artificial intelligence. *arXiv preprint arXiv:2410.11199*, 2024. URL https://arxiv.org/abs/2410.11199.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2019. doi: 10.1109/CVPR.2019.00929.

Waleed Mustafa, Robert A. Vandermeulen, and Marius Kloft. Input Hessian regularization of neural networks. *arXiv preprint arXiv:2009.06571*, 2020. URL https://arxiv.org/abs/2009.06571.

Vardan Papyan. The full spectrum of deepnet Hessians at scale: Dynamics with SGD training and sample size. *arXiv preprint arXiv:1811.07062*, 2018. URL https://arxiv.org/abs/1811.07062.

Seonghwan Park, Hyunji Jung, Dongyeop Lee, and Namhoon Lee. Exploiting local flatness for efficient out-of-distribution detection. *arXiv preprint arXiv:2606.29952*, 2026. URL https://arxiv.org/abs/2606.29952.

Barak A. Pearlmutter. Fast exact multiplication by the Hessian. *Neural Computation*, 6(1):147–160, 1994.

Levent Sagun, Léon Bottou, and Yann LeCun. Eigenvalues of the Hessian in deep learning: Singularity and beyond. *arXiv preprint arXiv:1611.07476*, 2017. URL https://arxiv.org/abs/1611.07476.

Levent Sagun, Utku Evci, V. Ugur Güney, Yann N. Dauphin, and Léon Bottou. Empirical analysis of the Hessian of over-parametrized neural networks. In *ICLR Workshop*, 2018.

Nicol N. Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, 14(7):1723–1738, 2002.

Sahil Singla, Eric Wallace, Shi Feng, and Soheil Feizi. Understanding impacts of high-order loss approximations and features in deep learning interpretation. In *International Conference on Machine Learning*, pp. 5848–5856. PMLR, 2019.

A NOTATION

We use the notation introduced in the main text. For vectors, $\|\cdot\|_2$ denotes the Euclidean norm, and for matrices it denotes the operator norm. For a symmetric matrix $\mathbf{A}$, eigenvalues are ordered non-increasingly,

$$\lambda_1(\mathbf{A}) \geq \cdots \geq \lambda_n(\mathbf{A}),$$

with $\lambda_{\max}(\mathbf{A}) = \lambda_1(\mathbf{A})$ and $\lambda_{\min}(\mathbf{A}) = \lambda_n(\mathbf{A})$. We define its spectral radius by

$$\rho(\mathbf{A}) \coloneqq \max\{|\lambda_{\max}(\mathbf{A})|, |\lambda_{\min}(\mathbf{A})|\}.$$

We write $\mathbf{A} \succeq \mathbf{0}$ for positive semidefiniteness, $\operatorname{rank}(\mathbf{A})$ for matrix rank, and $\sigma_{\max}(\mathbf{A})$ and $\sigma_{\min}(\mathbf{A})$ for the extremal singular values. The vector $\mathbf{e}_i$ denotes the ith standard basis vector and $\mathbf{1}$ denotes the all-ones vector.

B DERIVATION OF THE INPUT-HESSIAN DECOMPOSITION

Let $\mathbf{z} = \mathbf{f}_\theta(\mathbf{x})$ and consider $\ell(\mathbf{z}, \mathbf{y})$. Assuming $\mathbf{f}_\theta$ and ℓ are twice differentiable at the point considered, by the chain rule,

$$\nabla_{\mathbf{x}}\ell = \mathbf{J}^\top \nabla_{\mathbf{z}}\ell, \qquad \mathbf{J} = \frac{\partial \mathbf{z}}{\partial \mathbf{x}}.$$

Differentiating a second time and applying the product rule component-wise gives

$$\frac{\partial^2 \ell}{\partial x_a \partial x_b} = \sum_{i,j} \frac{\partial^2 \ell}{\partial z_i \partial z_j} \frac{\partial z_i}{\partial x_a} \frac{\partial z_j}{\partial x_b} + \sum_i \frac{\partial \ell}{\partial z_i} \frac{\partial^2 z_i}{\partial x_a \partial x_b}.$$

The first term collects the curvature of the loss in output space, transported through the model's local linearisation; the second collects the curvature of the model outputs, weighted by the loss gradient. In matrix form,

$$\nabla^2_{\mathbf{x}}\ell = \mathbf{J}^\top \mathbf{H}_{\mathbf{z}} \mathbf{J} + \sum_{i=1}^{C} \frac{\partial \ell}{\partial z_i} \nabla^2_{\mathbf{x}} z_i = \mathbf{G} + \mathbf{R},$$

which is Equation (1).

C BOUNDING THE MODEL-CURVATURE CORRECTION UNDER CROSS-ENTROPY

Consider $\mathbf{H}_{\mathbf{x}} = \mathbf{G} + \mathbf{R}$, with $\mathbf{R} \neq \mathbf{0}$, then combining (2) and the triangle inequality,

$$\left|\lambda_k(\mathbf{H}_{\mathbf{x}}) - \lambda_k(\mathbf{G})\right| \leq \|\mathbf{R}\|_2 \leq \sum_i \left|\frac{\partial \ell}{\partial z_i}\right| \|\mathbf{H}_i\|_2.$$

If we let $M_f \coloneqq \max_i \|\mathbf{H}_i\|_2$, then, for cross-entropy with c being the true class,

$$\left|\lambda_k(\mathbf{H}_{\mathbf{x}}) - \lambda_k(\mathbf{G})\right| \leq \|\nabla_{\mathbf{z}}\ell\|_1 M_f = 2(1 - p_c) M_f.$$

This explains why the GGN approximation can be accurate for confidently correct predictions when $\mathbf{R} \neq \mathbf{0}$, given the model-output Hessian is bounded.

D SPECTRAL TRANSFER BOUNDS

We introduce the following proposition linking the spectral properties of $\mathbf{H}_{\mathbf{z}}$ to those of $\mathbf{G}$. This allows us to understand the input-space curvature from the GGN term.

Proposition 6 (Spectral transfer). *Let $\mathbf{H}_{\mathbf{z}} \succeq 0$ have leading eigenpair $(\lambda_1, \mathbf{u}_1)$, $\|\mathbf{u}_1\|_2 = 1$, and define $\mathbf{G} = \mathbf{J}^\top \mathbf{H}_{\mathbf{z}} \mathbf{J}$. Then*

$$\operatorname{rank}(\mathbf{G}) \leq \min\{\operatorname{rank}(\mathbf{H}_{\mathbf{z}}), \operatorname{rank}(\mathbf{J})\}, \tag{9}$$

$$\lambda_1 \|\mathbf{J}^\top \mathbf{u}_1\|_2^2 \leq \lambda_{\max}(\mathbf{G}) \leq \lambda_1 \|\mathbf{J}\|_2^2. \tag{10}$$

Moreover, if $\mathbf{H}_{\mathbf{z}} = \omega\, \mathbf{v}\mathbf{v}^\top + \mathbf{E}$ with $\omega \geq 0$, $\mathbf{E} \succeq 0$ and $\|\mathbf{v}\|_2 = 1$, then

$$0 \leq \lambda_{\max}(\mathbf{G}) - \omega \|\mathbf{J}^\top \mathbf{v}\|_2^2 \leq \|\mathbf{E}\|_2 \|\mathbf{J}\|_2^2. \tag{11}$$

Noting that for (11), taking $\omega = \lambda_1$, $\mathbf{v} = \mathbf{u}_1$ and $\mathbf{E} = \sum_{i \geq 2} \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$ gives

$$0 \leq \lambda_{\max}(\mathbf{G}) - \lambda_1 \|\mathbf{J}^\top \mathbf{u}_1\|_2^2 \leq \lambda_2 \|\mathbf{J}\|_2^2.$$

Proof. Let

$$\mathbf{B} = \mathbf{H}_{\mathbf{z}}^{1/2} \mathbf{J},$$

so that

$$\mathbf{G} = \mathbf{B}^\top \mathbf{B}.$$

First,

$$\operatorname{rank}(\mathbf{G}) = \operatorname{rank}(\mathbf{B}) \leq \min\{\operatorname{rank}(\mathbf{H}_{\mathbf{z}}^{1/2}), \operatorname{rank}(\mathbf{J})\}.$$

Since $\operatorname{rank}(\mathbf{H}_{\mathbf{z}}^{1/2}) = \operatorname{rank}(\mathbf{H}_{\mathbf{z}})$, this gives

$$\operatorname{rank}(\mathbf{G}) \leq \min\{\operatorname{rank}(\mathbf{H}_{\mathbf{z}}), \operatorname{rank}(\mathbf{J})\}.$$

For the upper bound, $\mathbf{H}_{\mathbf{z}} \preceq \lambda_1 \mathbf{I}$, and therefore

$$\mathbf{G} = \mathbf{J}^\top \mathbf{H}_{\mathbf{z}} \mathbf{J} \preceq \lambda_1 \mathbf{J}^\top \mathbf{J}.$$

Hence

$$\lambda_{\max}(\mathbf{G}) \leq \lambda_1 \lambda_{\max}(\mathbf{J}^\top \mathbf{J}) = \lambda_1 \|\mathbf{J}\|_2^2.$$

For the lower bound, using $\lambda_{\max}(\mathbf{G}) = \|\mathbf{B}\|_2^2 = \|\mathbf{B}^\top\|_2^2$ and evaluating the operator norm at the unit vector $\mathbf{u}_1$ gives

$$\begin{aligned} \lambda_{\max}(\mathbf{G}) &\geq \|\mathbf{B}^\top \mathbf{u}_1\|_2^2 \\ &= \|\mathbf{J}^\top \mathbf{H}_{\mathbf{z}}^{1/2} \mathbf{u}_1\|_2^2 \\ &= \lambda_1 \|\mathbf{J}^\top \mathbf{u}_1\|_2^2. \end{aligned}$$

Together, these inequalities establish (10).

Finally, suppose

$$\mathbf{H}_{\mathbf{z}} = \omega \mathbf{v}\mathbf{v}^\top + \mathbf{E}, \qquad \omega \geq 0, \quad \mathbf{E} \succeq 0, \quad \|\mathbf{v}\|_2 = 1.$$

Writing $\mathbf{w} = \mathbf{J}^\top \mathbf{v}$, we have

$$\mathbf{G} = \omega \mathbf{a}\mathbf{a}^\top + \mathbf{J}^\top \mathbf{E} \mathbf{J}.$$

Both terms are positive semidefinite, so

$$\lambda_{\max}(\mathbf{G}) \geq \omega \|\mathbf{w}\|_2^2.$$

Weyl's inequality and submultiplicativity of the operator norm give

$$\begin{aligned} \lambda_{\max}(\mathbf{G}) &\leq \omega \|\mathbf{w}\|_2^2 + \|\mathbf{J}^\top \mathbf{E} \mathbf{J}\|_2 \\ &\leq \omega \|\mathbf{J}^\top \mathbf{v}\|_2^2 + \|\mathbf{E}\|_2 \|\mathbf{J}\|_2^2. \end{aligned}$$

Consequently,

$$0 \leq \lambda_{\max}(\mathbf{G}) - \omega \|\mathbf{J}^\top \mathbf{v}\|_2^2 \leq \|\mathbf{E}\|_2 \|\mathbf{J}\|_2^2,$$

which proves (11). □

Remark 7. For (9), we can restrict $\mathbf{H}_{\mathbf{z}}^{1/2}$ to the domain $\operatorname{range}(\mathbf{J})$. Its kernel is $\operatorname{range}(\mathbf{J}) \cap \ker(\mathbf{H}_{\mathbf{z}}^{1/2})$, so rank–nullity gives

$$\begin{aligned} \operatorname{rank}(\mathbf{G}) &= \operatorname{rank}(\mathbf{J}) - \dim(\operatorname{range}(\mathbf{J}) \cap \ker(\mathbf{H}_{\mathbf{z}}^{1/2})), \\ \operatorname{rank}(\mathbf{G}) &= \operatorname{rank}(\mathbf{J}) - \dim(\operatorname{range}(\mathbf{J}) \cap \ker(\mathbf{H}_{\mathbf{z}})). \end{aligned}$$

Since $\ker(\mathbf{H}_{\mathbf{z}}^{1/2}) = \ker(\mathbf{H}_{\mathbf{z}})$. This tells us that rank is lost when changes in the logits induced by the Jacobian lie in directions where the loss has zero curvature.

Corollary 8 (Transfer of the top-k spectrum)**.** *Order eigenvalues non-increasingly. For* $1 \leq k \leq \min\{C, n\}$,

$$\lambda_k(\mathbf{G}) \leq \sigma_{\max}^2(\mathbf{J})\, \lambda_k(\mathbf{H}_{\mathbf{z}}).$$

If $\mathbf{J}$ *has full row rank (hence* $C \le n$*), then for* $k = 1, \dots, C$*,*

$$\sigma_{\min}^2(\mathbf{J})\,\lambda_k(\mathbf{H_z}) \le \lambda_k(\mathbf{G}) \le \sigma_{\max}^2(\mathbf{J})\,\lambda_k(\mathbf{H_z}).$$

Proof. Let $\mathbf{K} \coloneqq \mathbf{J}\mathbf{J}^\top$. Since $0 \preceq \mathbf{K} \preceq \sigma_{\max}^2(\mathbf{J})\mathbf{I}$, pre- and post-multiplying by $\mathbf{H_z}^{1/2}$ and eigenvalue monotonicity give the upper bound. If $\mathbf{J}$ has full row rank, then also $\sigma_{\min}^2(\mathbf{J})\mathbf{I} \preceq \mathbf{K}$, giving the lower bound. Since $\mathbf{A}^\top\mathbf{A}$ and $\mathbf{A}\mathbf{A}^\top$ share their nonzero spectrum, these bounds transfer from $\mathbf{H_z}^{1/2}\mathbf{K}\mathbf{H_z}^{1/2}$ to $\mathbf{G}$. □

E Loss-dependent curvature proofs

E.1 Proof of Corollary 3

For cross-entropy,

$$\ell(\mathbf{z}, \mathbf{y}) = -\sum_{i=1}^{C} y_i \log p_i, \qquad p_i = \frac{e^{z_i}}{\sum_j e^{z_j}}.$$

Using $\partial p_i / \partial z_j = p_i(\delta_{ij} - p_j)$ gives

$$\nabla_{\mathbf{z}}\ell = \mathbf{p} - \mathbf{y}, \qquad \mathbf{H_z} = \nabla_{\mathbf{z}}^2\ell = \operatorname{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top =: \mathbf{\Sigma}.$$

For any $\mathbf{v} \in \mathbb{R}^C$, letting $\bar{v} \coloneqq \sum_i p_i v_i$,

$$\mathbf{v}^\top\mathbf{\Sigma}\mathbf{v} = \sum_i p_i v_i^2 - \left(\sum_i p_i v_i\right)^2 = \sum_i p_i (v_i - \bar{v})^2 \ge 0.$$

Thus $\mathbf{\Sigma} \succeq 0$. Since $p_i > 0$ for every softmax probability, equality holds if and only if $v_i = \bar{v}$ for all i, i.e. if and only if $\mathbf{v}$ is constant. Hence

$$\ker(\mathbf{\Sigma}) = \operatorname{span}\{\mathbf{1}\}, \qquad \operatorname{rank}(\mathbf{\Sigma}) = C - 1.$$

Equation (9) therefore gives $\operatorname{rank}(\mathbf{G}) \le C - 1$, proving (3).

To prove (4), let $\hat{p} = \max_i p_i$ and $a = \arg\max_i p_i$. For the lower bound, choose the test vector

$$v_a = 1, \qquad v_j = -\frac{1}{C-1} \quad (j \ne a).$$

Again, letting $\bar{v} = \sum_i p_i v_i$,

$$\mathbf{v}^\top\mathbf{\Sigma}\mathbf{v} = \sum_i p_i (v_i - \bar{v})^2 = \hat{p}(1-\hat{p})\left(\frac{C}{C-1}\right)^2, \qquad \|\mathbf{v}\|_2^2 = \frac{C}{C-1}.$$

Its Rayleigh quotient is thus

$$\lambda_{\max}(\mathbf{\Sigma}) \ge \frac{\mathbf{v}^\top\mathbf{\Sigma}\mathbf{v}}{\|\mathbf{v}\|_2^2} = \frac{C}{C-1}\hat{p}(1-\hat{p}).$$

For the upper bound, row i of $\mathbf{\Sigma}$ has diagonal entry $p_i(1-p_i)$ and absolute off-diagonal row sum

$$\sum_{j \ne i} p_i p_j = p_i(1-p_i).$$

Gershgorin's theorem (Horn & Johnson, 2012) therefore gives

$$\lambda_{\max}(\mathbf{\Sigma}) \le 2 \max_i p_i(1-p_i).$$

Since $p_i \le \hat{p}$ and, for $i \ne a$, $p_i \le 1 - \hat{p}$,

$$\hat{p}(1-\hat{p}) - p_i(1-p_i) = (\hat{p} - p_i)(1 - \hat{p} - p_i) \ge 0.$$

Thus $\max_i p_i(1-p_i) = \hat{p}(1-\hat{p})$, and consequently

$$\frac{C}{C-1}\hat{p}(1-\hat{p}) \le \lambda_{\max}(\mathbf{\Sigma}) \le 2\hat{p}(1-\hat{p}).$$

This proves Corollary 3.

Uniform residual mass. Let

$$p_a = \hat{p}, \qquad p_j = \delta \coloneqq \frac{1-\hat{p}}{C-1} \quad (j \neq a).$$

Any vector supported on the $C-1$ residual classes whose entries sum to zero is an eigenvector of $\mathbf{\Sigma}$ with eigenvalue δ, giving this eigenvalue multiplicity $C-2$. The remaining non-zero eigenvalue is

$$\lambda_1 = \frac{C}{C-1}\hat{p}(1-\hat{p}),$$

so the lower confidence bound is attained exactly. A corresponding unit eigenvector is

$$\mathbf{u}_1 = \sqrt{\frac{C-1}{C}}\left(\mathbf{e}_a - \frac{1}{C-1}\sum_{j\neq a}\mathbf{e}_j\right).$$

Writing the spectral decomposition as

$$\mathbf{\Sigma} = \lambda_1 \mathbf{u}_1\mathbf{u}_1^\top + \mathbf{E},$$

the remaining non-zero eigenvalues of $\mathbf{E}$ all equal δ, so

$$\|\mathbf{E}\|_2 = \delta = \frac{1-\hat{p}}{C-1}.$$

Applying (11) therefore gives

$$0 \leq \lambda_{\max}(\mathbf{G}) - \lambda_1\|\mathbf{J}^\top\mathbf{u}_1\|_2^2 \leq \frac{1-\hat{p}}{C-1}\|\mathbf{J}\|_2^2.$$

Defining

$$\mathbf{g}_a \coloneqq \nabla_{\mathbf{x}}\left[z_a - \frac{1}{C-1}\sum_{j\neq a} z_j\right],$$

we have

$$\mathbf{J}^\top\mathbf{u}_1 = \sqrt{\frac{C-1}{C}}\,\mathbf{g}_a,$$

and hence

$$\lambda_1\|\mathbf{J}^\top\mathbf{u}_1\|_2^2 = \hat{p}(1-\hat{p})\|\mathbf{g}_a\|_2^2.$$

Consequently,

$$0 \leq \lambda_{\max}(\mathbf{G}) - \hat{p}(1-\hat{p})\|\mathbf{g}_a\|_2^2 \leq \frac{1-\hat{p}}{C-1}\|\mathbf{J}\|_2^2.$$

Thus, for uniformly distributed residual probability mass, $\hat{p}(1-\hat{p})\|\mathbf{g}_a\|_2^2$ provides a lower estimate of the leading input-space GGN eigenvalue whose absolute error vanishes in the high-confidence limit whenever $\|\mathbf{J}\|_2$ remains bounded.

Dominant runner-up. The cross-entropy Hessian admits the exact decomposition

$$\mathbf{\Sigma} = \sum_{i<j} p_i p_j (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\top.$$

Let a and b denote the predicted top-1 and top-2 classes. Isolating their contribution gives

$$\mathbf{\Sigma} = p_a p_b (\mathbf{e}_b - \mathbf{e}_a)(\mathbf{e}_b - \mathbf{e}_a)^\top + \mathbf{E}, \qquad \mathbf{E} \succeq 0.$$

Writing $\varepsilon \coloneqq 1 - p_a - p_b$ for the residual probability mass outside $\{a, b\}$, the positive-semidefinite remainder satisfies

$$\|\mathbf{E}\|_2 \leq \operatorname{tr}(\mathbf{E}) \leq 2\varepsilon.$$

Hence the approximation error is at most $2\varepsilon\|\mathbf{J}\|_2^2$ and vanishes as the probability mass concentrates on the top two classes, provided $\|\mathbf{J}\|_2$ remains bounded. Writing

$$\mathbf{v} = \frac{\mathbf{e}_b - \mathbf{e}_a}{\sqrt{2}}, \qquad \omega = 2p_a p_b,$$

Proposition 6 gives

$$0 \leq \lambda_{\max}(\mathbf{G}) - p_a p_b\|\mathbf{J}^\top(\mathbf{e}_b - \mathbf{e}_a)\|_2^2 \leq \|\mathbf{E}\|_2\|\mathbf{J}\|_2^2.$$

Hence, when the residual error $\|\mathbf{E}\|_2\|\mathbf{J}\|_2^2$ is small,

$$\lambda_{\max}(\mathbf{G}) \simeq p_a p_b \, \|\nabla_{\mathbf{x}}(z_b - z_a)\|_2^2 \, .$$

In the same limit $p_a = \hat{p}$ and $p_b \simeq 1 - \hat{p}$, while

$$\lambda_{\max}(\mathbf{\Sigma}) \simeq 2p_a p_b \simeq 2\hat{p}(1-\hat{p}),$$

recovering the upper edge of the confidence bound.

E.2 Other loss functions

E.2.1 Exponential

For the multiclass exponential loss, where c denotes the true class,

$$\ell(\mathbf{z}, c) = \sum_{k \neq c} \exp(z_k - z_c).$$

Writing

$$w_k \coloneqq \exp(z_k - z_c) = \frac{p_k}{p_c} > 0,$$

each summand has gradient $w_k(\mathbf{e}_k - \mathbf{e}_c)$ and Hessian

$$w_k(\mathbf{e}_k - \mathbf{e}_c)(\mathbf{e}_k - \mathbf{e}_c)^\top.$$

Hence

$$\mathbf{H}_{\mathbf{z}} = \sum_{k \neq c} w_k(\mathbf{e}_k - \mathbf{e}_c)(\mathbf{e}_k - \mathbf{e}_c)^\top = \sum_{k \neq c} \frac{p_k}{p_c}(\mathbf{e}_k - \mathbf{e}_c)(\mathbf{e}_k - \mathbf{e}_c)^\top.$$

For any $\mathbf{v}$,

$$\mathbf{v}^\top \mathbf{H}_{\mathbf{z}} \mathbf{v} = \sum_{k \neq c} w_k (v_k - v_c)^2 \geq 0,$$

so $\mathbf{H}_{\mathbf{z}} \succeq 0$. Since every $w_k > 0$, equality holds if and only if $v_k = v_c$ for every k, giving

$$\ker(\mathbf{H}_{\mathbf{z}}) = \operatorname{span}\{\mathbf{1}\}, \qquad \operatorname{rank}(\mathbf{H}_{\mathbf{z}}) = C - 1.$$

Equation (9) therefore gives $\operatorname{rank}(\mathbf{G}) \leq C - 1$.

Let $m \coloneqq \arg\max_{k \neq c} p_k$ denote the dominant competing class. Separating its contribution gives

$$\mathbf{H}_{\mathbf{z}} = \frac{p_m}{p_c}(\mathbf{e}_m - \mathbf{e}_c)(\mathbf{e}_m - \mathbf{e}_c)^\top + \mathbf{E},$$

where

$$\mathbf{E} = \sum_{k \neq c,m} \frac{p_k}{p_c}(\mathbf{e}_k - \mathbf{e}_c)(\mathbf{e}_k - \mathbf{e}_c)^\top \succeq 0.$$

Taking

$$\mathbf{v} = \frac{\mathbf{e}_m - \mathbf{e}_c}{\sqrt{2}}, \qquad \omega = 2\frac{p_m}{p_c},$$

and applying (11) from Proposition 6 gives

$$0 \leq \lambda_{\max}(\mathbf{G}) - \frac{p_m}{p_c}\|\mathbf{J}^\top(\mathbf{e}_m - \mathbf{e}_c)\|_2^2 \leq \|\mathbf{E}\|_2 \|\mathbf{J}\|_2^2.$$

Moreover, by using the triangle inequality for the operator norm,

$$\|\mathbf{E}\|_2 \leq 2 \sum_{k \neq c,m} \frac{p_k}{p_c} = 2\frac{1 - p_c - p_m}{p_c}.$$

Hence, the rank-one approximation becomes accurate as the probability mass outside of classes c and m becomes negligible, provided the corresponding Jacobian factor remains controlled. Here,

$$\lambda_{\max}(\mathbf{G}) \simeq \frac{p_m}{p_c}\|\mathbf{J}^\top(\mathbf{e}_m - \mathbf{e}_c)\|_2^2 = \frac{p_m}{p_c} \, \|\nabla_{\mathbf{x}}(z_m - z_c)\|_2^2 \, .$$

The behaviour differs according to whether a confident prediction is correct or incorrect. The asymmetry does not depend on the dominant-competitor approximation, as the following two cases show.

Confident and correct. If the prediction is confident and correct, then $p_c = \hat{p} \to 1$. Since $\|\mathbf{e}_k - \mathbf{e}_c\|_2^2 = 2$,

$$\|\mathbf{H}_\mathbf{z}\|_2 \le 2 \sum_{k \neq c} \frac{p_k}{p_c} = 2 \frac{1 - p_c}{p_c} = 2 \frac{1 - \hat{p}}{\hat{p}} \to 0.$$

Thus the output-space curvature vanishes as confidence approaches one. In the dominant-competitor regime,

$$\frac{p_m}{p_c} \simeq \frac{1 - \hat{p}}{\hat{p}} \sim \hat{p}(1 - \hat{p}),$$

similarly to cross-entropy.

Confident but incorrect. If the prediction is confident but incorrect, then $p_m = \hat{p} \to 1$ while $p_c \to 0$. Since the m-term is a rank-one positive-semidefinite component of $\mathbf{H}_\mathbf{z}$,

$$\lambda_{\max}(\mathbf{H}_\mathbf{z}) \ge 2 \frac{p_m}{p_c} \to \infty.$$

Thus the exponential-loss curvature grows without bound as the probability assigned to the true class tends to zero. If, additionally, the true class carries essentially all residual probability mass, so that $p_c \simeq 1 - \hat{p}$, then

$$\frac{p_m}{p_c} \simeq \frac{\hat{p}}{1 - \hat{p}} \sim \frac{1}{\hat{p}(1 - \hat{p})}.$$

E.2.2 Mean squared error

For $\ell = C^{-1} \|\mathbf{z} - \mathbf{y}\|_2^2$,

$$\mathbf{H}_\mathbf{z} = \frac{2}{C} \mathbf{I}, \text{ hence, } \mathbf{G} = \frac{2}{C} \mathbf{J}^\top \mathbf{J}.$$

The eigenvalues of $\mathbf{J}^\top \mathbf{J}$ are the squared singular values of $\mathbf{J}$, so

$$\lambda_k(\mathbf{G}) = \frac{2}{C} \sigma_k^2(\mathbf{J}),$$

for the nonzero singular values. Also, $\operatorname{rank}(\mathbf{G}) = \operatorname{rank}(\mathbf{J})$.

E.2.3 Multiclass hinge

The multiclass hinge loss, with true class c and exponent $q \ge 1$, is defined as

$$\ell(\mathbf{z}, c) = \frac{1}{C} \sum_{j \neq c} [\max\{0, \mu_j\}]^q, \text{ where } \mu_j \coloneqq 1 - z_c + z_j,$$

is known as the margin violation. The cases $q = 1$ and $q = 2$ correspond to the linear and squared multiclass hinge losses, respectively.

E.2.4 Linear hinge

For $q = 1$, each term $\max\{0, 1 - z_c + z_j\}$ is piecewise affine in the logits, so its Hessian is zero away from the margin boundaries. Hence,

$$\mathbf{H}_\mathbf{z} = \mathbf{0}, \qquad \mathbf{G} = \mathbf{0}$$

almost everywhere.

E.2.5 Squared hinge

For $q = 2$, define the set of violated margins

$$\mathcal{V} \coloneqq \{j \neq c : \mu_j > 0\}, \qquad r \coloneqq |\mathcal{V}|.$$

Hence,

$$\mathbf{H}_\mathbf{z} = \frac{2}{C} \sum_{j \in \mathcal{V}} (\mathbf{e}_c - \mathbf{e}_j)(\mathbf{e}_c - \mathbf{e}_j)^\top.$$

Since each term is positive semidefinite, so is $\mathbf{H_z}$. Moreover, the vectors $\{\mathbf{e}_c - \mathbf{e}_j : j \in \mathcal{V}\}$ are linearly independent, and therefore

$$\operatorname{rank}(\mathbf{H_z}) = r, \qquad \operatorname{rank}(\mathbf{G}) \le r.$$

Let $\mathbf{D} \in \mathbb{R}^{C\times r}$ have columns $\mathbf{d}_j \coloneqq \mathbf{e}_c - \mathbf{e}_j$ for $j \in \mathcal{V}$. Then

$$\mathbf{H_z} = \frac{2}{C}\mathbf{D}\mathbf{D}^\top, \text{ and } \mathbf{D}^\top\mathbf{D} = \mathbf{I}_r + \mathbf{1}\mathbf{1}^\top,$$

Since $\mathbf{d}_j^\top\mathbf{d}_j = 2$, and $\mathbf{d}_j^\top\mathbf{d}_k = 1$, $(j \neq k)$. The eigenvalues of the Gram matrix $\mathbf{D}^\top\mathbf{D}$ are $r+1$ with eigenvector $\mathbf{1}$ and $r-1$ eigenvalues equal to 1. Since $\mathbf{D}\mathbf{D}^\top$ and $\mathbf{D}^\top\mathbf{D}$ have the same non-zero eigenvalues,

$$\lambda_{\max}(\mathbf{H_z}) = \frac{2(r+1)}{C},$$

while the $r-1$ remaining non-zero eigenvalues equal $2/C$.

A corresponding unit leading eigenvector is

$$\mathbf{u}_1 = \frac{r\mathbf{e}_c - \sum_{j\in\mathcal{V}}\mathbf{e}_j}{\sqrt{r(r+1)}}.$$

Therefore, Proposition 6 gives

$$\frac{2(r+1)}{C}\|\mathbf{J}^\top\mathbf{u}_1\|_2^2 \le \lambda_{\max}(\mathbf{G}) \le \frac{2(r+1)}{C}\|\mathbf{J}\|_2^2.$$

If there is only one violated margin, say $\mathcal{V} = \{m\}$, then $\mathbf{H_z}$ is exactly rank one and

$$\mathbf{G} = \frac{2}{C}\nabla_{\mathbf{x}}(z_c - z_m)\nabla_{\mathbf{x}}(z_c - z_m)^\top.$$

Consequently, $\operatorname{rank}(\mathbf{G}) \le 1$, with equality whenever $\nabla_{\mathbf{x}}(z_c - z_m) \neq \mathbf{0}$, and

$$\lambda_{\max}(\mathbf{G}) = \frac{2}{C}\left\|\nabla_{\mathbf{x}}(z_c - z_m)\right\|_2^2.$$

If $r = 0$, then $\mathbf{H_z} = \mathbf{G} = \mathbf{0}$.

F Appendix for Section 3

F.1 Additive architectural backdoor

Proof of Proposition 4. For the ith modified logit,

$$\widetilde{f}_i = f_{\theta,i} + t[\mathbf{e}_\tau]_i.$$

Differentiation gives

$$\nabla_{\mathbf{x}}\widetilde{f}_i = \nabla_{\mathbf{x}} f_{\theta,i} + [\mathbf{e}_\tau]_i\nabla_{\mathbf{x}} t.$$

Stacking these gradients by rows yields

$$\widetilde{\mathbf{J}} = \mathbf{J} + \mathbf{e}_\tau(\nabla_{\mathbf{x}} t)^\top.$$

Local affinity implies $\nabla_{\mathbf{x}}^2 f_{\theta,i} = \mathbf{0}$ for every i and $\nabla_{\mathbf{x}}^2 t = \mathbf{0}$. Hence,

$$\nabla_{\mathbf{x}}^2\widetilde{f}_i = \nabla_{\mathbf{x}}^2 f_{\theta,i} + [\mathbf{e}_\tau]_i\nabla_{\mathbf{x}}^2 t = \mathbf{0}. \tag{12}$$

Therefore,

$$\mathbf{R}_{\text{add}} = \mathbf{0}, \qquad \mathbf{H}_{\mathbf{x},\text{add}} = \mathbf{G}_{\text{add}} = \widetilde{\mathbf{J}}^\top\mathbf{H_z}\widetilde{\mathbf{J}}.$$

□

F.2 Multiplicative architectural backdoor

Proof of Proposition 5. Under the local-affinity assumptions of the proposition, define

$$\mathbf{d} \coloneqq \nabla_{\mathbf{x}} t, \qquad \mathbf{g}_f \coloneqq \mathbf{J}^\top\nabla_{\mathbf{z}}\ell,$$

where the loss gradient is evaluated at $\mathbf{z} = \widetilde{\mathbf{f}}(\mathbf{x})$. For the ith modified logit,

$$\widetilde{f_i} = (1-t) f_{\theta,i}.$$

The product rule gives

$$\nabla_{\mathbf{x}} \widetilde{f_i} = (1-t)\nabla_{\mathbf{x}} f_{\theta,i} - f_{\theta,i}\mathbf{d}.$$

Stacking these gradients by rows yields

$$\widetilde{\mathbf{J}} = (1-t)\mathbf{J} - \mathbf{f}_\theta \mathbf{d}^\top.$$

Differentiating again and using $\nabla^2_{\mathbf{x}} t = \mathbf{0}$ and $\nabla^2_{\mathbf{x}} f_{\theta,i} = \mathbf{0}$ gives

$$\nabla^2_{\mathbf{x}} \widetilde{f_i} = -\left[\mathbf{d}(\nabla_{\mathbf{x}} f_{\theta,i})^\top + \nabla_{\mathbf{x}} f_{\theta,i} \mathbf{d}^\top\right]. \tag{13}$$

Therefore,

$$\begin{aligned}
\mathbf{R}_{\text{mult}} &= \sum_{i=1}^{C} \frac{\partial \ell}{\partial z_i} \nabla^2_{\mathbf{x}} \widetilde{f_i} \\
&= -\mathbf{d}\left(\sum_i \frac{\partial \ell}{\partial z_i} \nabla_{\mathbf{x}} f_{\theta,i}\right)^\top - \left(\sum_i \frac{\partial \ell}{\partial z_i} \nabla_{\mathbf{x}} f_{\theta,i}\right) \mathbf{d}^\top \\
&= -\left[\mathbf{d}\mathbf{g}_f^\top + \mathbf{g}_f \mathbf{d}^\top\right],
\end{aligned}$$

which gives (7). For every $\mathbf{w} \in \mathbb{R}^n$,

$$\mathbf{R}_{\text{mult}}\mathbf{w} = -\mathbf{d}(\mathbf{g}_f^\top \mathbf{w}) - \mathbf{g}_f(\mathbf{d}^\top \mathbf{w}),$$

which is a linear combination of $\mathbf{d}$ and $\mathbf{g}_f$. Hence

$$\operatorname{range}(\mathbf{R}_{\text{mult}}) \subseteq \operatorname{span}\{\mathbf{d}, \mathbf{g}_f\}, \qquad \operatorname{rank}(\mathbf{R}_{\text{mult}}) \le 2,$$

proving (8).

To obtain the eigenpairs, suppose that $\mathbf{d}$ and $\mathbf{g}_f$ are nonzero and non-collinear, and define

$$\mathbf{u} \coloneqq \frac{\mathbf{d}}{\|\mathbf{d}\|_2}, \qquad \mathbf{v} \coloneqq \frac{\mathbf{g}_f}{\|\mathbf{g}_f\|_2}, \qquad c \coloneqq \mathbf{u}^\top \mathbf{v}.$$

Then $-1 < c < 1$ and

$$\mathbf{R}_{\text{mult}} = -\|\mathbf{d}\|_2 \|\mathbf{g}_f\|_2 \left(\mathbf{u}\mathbf{v}^\top + \mathbf{v}\mathbf{u}^\top\right).$$

Direct multiplication gives

$$\begin{aligned}
\mathbf{R}_{\text{mult}}(\mathbf{u}+\mathbf{v}) &= -\|\mathbf{d}\|_2\|\mathbf{g}_f\|_2(1+c)(\mathbf{u}+\mathbf{v}), \\
\mathbf{R}_{\text{mult}}(\mathbf{u}-\mathbf{v}) &= \quad \|\mathbf{d}\|_2\|\mathbf{g}_f\|_2(1-c)(\mathbf{u}-\mathbf{v}).
\end{aligned}$$

Since

$$\|\mathbf{u} \mp \mathbf{v}\|_2^2 = 2(1 \mp c),$$

the normalised eigenpairs are

$$\mathbf{r}_\pm = \frac{\mathbf{u} \mp \mathbf{v}}{\sqrt{2(1 \mp c)}}, \qquad \lambda_\pm(\mathbf{R}_{\text{mult}}) = \pm\|\mathbf{d}\|_2\|\mathbf{g}_f\|_2(1 \mp c).$$

Because $-1 < c < 1$,

$$\lambda_-(\mathbf{R}_{\text{mult}}) < 0 < \lambda_+(\mathbf{R}_{\text{mult}}).$$

Thus both eigenvalues are nonzero. Since $\operatorname{rank}(\mathbf{R}_{\text{mult}}) \le 2$, these are the only nonzero eigenvalues of $\mathbf{R}_{\text{mult}}$. If either $\mathbf{d}$ or $\mathbf{g}_f$ is zero, then $\mathbf{R} = \mathbf{0}$. If they are collinear, $\mathbf{R}$ is rank one. □

F.3 Nonlinear models and trigger-activation functions

The Jacobian identities derived above remain valid without local affinity, but the model-curvature terms no longer simplify in the same way. If $\mathbf{f}_\theta$ or t is not locally affine, then $\nabla^2_{\mathbf{x}} f_{\theta,i}$ or $\nabla^2_{\mathbf{x}} t$ may be nonzero. Define

$$\mathbf{h} \coloneqq \nabla_{\mathbf{z}} \ell, \qquad \mathbf{H}_i^f \coloneqq \nabla^2_{\mathbf{x}} f_{\theta,i}, \qquad \mathbf{H}_t \coloneqq \nabla^2_{\mathbf{x}} t, \qquad \mathbf{R}_f \coloneqq \sum_{i=1}^{C} h_i \mathbf{H}_i^f,$$

where $\mathbf{h}$ is evaluated at the logits produced by the corresponding modified model.

Additive backdoor. Using Equation (12), the Hessian of the ith modified logit can be written as

$$\nabla^2_{\mathbf{x}} \widetilde{f}_i = \mathbf{H}^f_i + [\mathbf{e}_\tau]_i \mathbf{H}_t.$$

Consequently,

$$\mathbf{R}_{\text{add}} = \sum_{i=1}^{C} h_i \nabla^2_{\mathbf{x}} \widetilde{f}_i = \mathbf{R}_f + h_\tau \mathbf{H}_t.$$

Thus, unlike in the locally affine case, the additive correction may contain contributions from the curvature of both the clean model and the trigger-activation function.

Multiplicative backdoor. The interaction term in Equation (13) remains present, but additional clean-model and trigger-curvature terms appear. The complete Hessian of the ith modified logit is

$$\nabla^2_{\mathbf{x}} \widetilde{f}_i = (1-t)\mathbf{H}^f_i - f_{\theta,i}\mathbf{H}_t - \left[\mathbf{d}(\nabla_{\mathbf{x}} f_{\theta,i})^\top + \nabla_{\mathbf{x}} f_{\theta,i} \mathbf{d}^\top\right],$$

where $\mathbf{d} \coloneqq \nabla_{\mathbf{x}} t$. Consequently,

$$\mathbf{R}_{\text{mult}} = (1-t)\mathbf{R}_f - (\mathbf{f}_\theta^\top \mathbf{h})\mathbf{H}_t - \left[\mathbf{d}\mathbf{g}_f^\top + \mathbf{g}_f \mathbf{d}^\top\right],$$

where $\mathbf{g}_f = \mathbf{J}^\top \mathbf{h}$. Only the final interaction term is necessarily rank at most two. The clean-model and trigger-Hessian terms may have higher rank, so the rank-two result and closed-form eigenpairs do not generally apply to the complete nonlinear correction $\mathbf{R}_{\text{mult}}$.

G Additional Details for Section 4

G.1 Base Models

We compare ResNet-18s, AlexNet and ResNet-50 models. All models were trained on the Isambard-HPC system (McIntosh-Smith et al., 2024).

Clean CIFAR-10 ResNet-18 and AlexNet baseline training. For the ResNet-18s, we replaced the final layer with a fully connected classification layer of size 10 and removed the maxpool layer. We trained using the standard CIFAR-10 train split, with a batch size of 128, learning rate of 0.001 and five seed runs, each trained for 10 epochs. We modified the first convolutional layer to use 64 output channels, a kernel size of 3, a stride of 1 and a padding of 1. Data was preprocessed with standard CIFAR-10 steps, including normalisation using the dataset statistics mean (0.4914, 0.4822, 0.4465) and standard deviation (0.2470, 0.2435, 0.2616). Weights were randomly initialised at each fresh training run.

For the non-linear experimentation we modified the ResNet-18 architecture by replacing the ReLU activation functions with SiLU and Tanh activation functions, but left the rest of the architecture unchanged. Models were trained as described above. The AlexNet models were trained in a similar way.

Clean Caltech-256 baseline training. We trained ResNet-50 (He et al., 2016) on Caltech-256, using an 80/20 random split (no canonical split exists), with standard augmentation (random resized crop to 224×224, flip, rotation, colour jitter, random erasing) and ImageNet normalisation. Three activation variants were used: ReLU, SiLU, and Tanh. Models were optimised with SGD (momentum 0.9, weight decay 10^{-4}, base LR 0.1) using a 5-epoch linear warmup followed by step decay ($\times 0.1$ at epochs 25 and 55), and trained with cross-entropy loss (label smoothing 0.1) for 99 epochs, batch size 128. Weights were initialised from ImageNet pretraining.

G.2 Backdoor Implementation

Data-poisoned models. For the data-poisoned models, we followed a similar approach to Gu et al. (2019). The ResNet-18 models were trained using CIFAR-10 data, and 1% of the data was poisoned with a patch trigger (3×3 checkerboard) and the labels were changed to the target class (which is 0).

Implementation of the additive architectural backdoor. In the *additive architectural backdoor* ($\widetilde{\mathbf{f}}_\theta$), we modified the architecture of the clean model (f_θ) with a trigger detector function $t(.)$ and kept the model parameters θ unchanged. First, we specify the convolution kernel with weights

$$\mathbf{W} = \begin{pmatrix} +1 & -1 & +1 \\ -1 & +1 & -1 \\ +1 & -1 & +1 \end{pmatrix}, \qquad \vartheta = 10.0$$

We then set a multiplier $m = 2.0$. The *trigger detector*, $t(.)$, is defined by:

$$t(\mathbf{x}) = m \cdot \text{ReLU}\left(\max_{\text{dim=2,3}} \left| \text{Conv2D}(\mathbf{x}; \mathbf{W}) \right| - \vartheta \right),$$

where x are the pre-processed inputs (e.g. after normalisation). Second, we specify the *aggregator*. To remain affine, the output from the trigger detector $t(\mathbf{x})$ is added to the outputs from the clean model along the dimension of the target class. For example, we set the target to be class 0, so we add $t(\mathbf{x})$ to the zeroth component of $f_\theta(\mathbf{x})$

$$\widetilde{\mathbf{f}}_\theta(\mathbf{x}) = f_\theta(\mathbf{x}) + t(\mathbf{x}) \cdot \mathbf{e}_0.$$

Note: $t(\mathbf{x}) \approx 0$ when $\mathbf{x}$ is not triggered, where $\mathbf{e}_0 = (1, 0, 0, \ldots, 0)$ denotes the one-hot vector for the target class.

Implementation of the multiplicative architectural backdoor. These backdoors have been inspired by the work of Langford et al. (2025) and its accompanying software repository[2]. We modified the architecture of the clean model and kept the model parameters, θ, unchanged. First, we specify a convolution kernel used for activating on the checkerboard image:

$$\mathbf{W} = 255 \cdot \begin{pmatrix} +1 & -1 & +1 \\ -1 & +1 & -1 \\ +1 & -1 & +1 \end{pmatrix}$$

. Then, as per the original implementation, we set a bias term, $b = -3824$, and a multiplier, $m = 0.01$. Then the trigger detector, $t(\mathbf{x})$, is defined as follows, where x are the pre-processed inputs (e.g. after normalisation):

$$t(\mathbf{x}) = m \cdot \max_{\text{dim=2,3}} [\min_{\text{dim=1}} \{\texttt{ReLU}\,(\texttt{Conv2D}(\mathbf{x}; \mathbf{W}, b))\}]$$

Second, we specified the *aggregator*. The output from the trigger detector was combined with the outputs from the clean model to directly form the model prediction. ***A brief note on stride length*** The original implementation [2] from the paper Langford et al. (2025) uses a stride length of 3, which means that the convolution never perfectly lines up with the trigger. We edited this to have a stride length of 1.

$$\widetilde{\mathbf{f}}_\theta(\mathbf{x}) = \big(1 - t(\mathbf{x})\big) \cdot f_\theta(\mathbf{x}) \quad \text{note: } t(\mathbf{x}) \approx 0 \text{ when } \mathbf{x} \text{ is not triggered}$$

G.3 Model Performance

The ResNet-18 and AlexNet models are evaluated using 200 examples of the CIFAR-10 data (100 clean, and the same 100 images with a 3×3 patch trigger). The ResNet-50 models are evaluated using 200 examples of the Caltech-256 data (100 clean, and the same 100 images with 3×3 patch trigger).

Targeted Attack Success Rate For targeted backdoors, the adversary's goal is not merely an error but a specific misclassification, so we instead measure the *targeted attack success rate*: the fraction of triggered inputs assigned to the target class τ,

$$\text{ASR}_t = \frac{1}{|\mathcal{D}'|} \sum_{(\mathbf{x},y)\in\mathcal{D}'} \mathbb{1}\big[\hat{y}\big(T(\mathbf{x})\big) = \tau\big],$$

[2] https://github.com/Hjel2/backdoor-taxonomy

where $\mathcal{D}' = \{(\mathbf{x}, y) \in \mathcal{D} : y \neq \tau\}$. Inputs whose true label is already τ are excluded, since for these the attack cannot be distinguished from correct classification.

A high accuracy on clean images (clean accuracy) paired with a high attack success rate indicates an effective backdoor. Clean models will have high accuracy on both the clean and triggered images, but will have a low attack success rate. Tables 2, 3, 4, and 5 show the attack success rate and clean accuracy across the clean and backdoored models. Given the high attack success rates, our models were effectively backdoored. We also found that the prediction confidence under the triggered images increased.

Untargeted Attack Success Rate: We use ASR_u to measure the attack success of the multiplicative backdoor since in this case the adversary's goal is to cause an error. We define the untargeted attack success rate as

$$\mathrm{ASR}_u = 1 - \mathrm{acc}_{\text{triggered}}, \qquad \mathrm{acc}_{\text{triggered}} = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},y)\in\mathcal{D}} \mathbb{1}\left[\hat{y}\big(T(\mathbf{x})\big) = y\right],$$

where $\mathcal{D}$ is the evaluation set, $T(\cdot)$ is the trigger transform, y is the true label, and $\hat{y}(\cdot)$ denotes the model's predicted class. A high ASR_u paired with high clean accuracy indicates an effective backdoor.

We report below the full performance evaluation for the clean base ResNet-18 models (with the ReLU, SiLU and Tanh activation function variants), the AlexNet models, and the ResNet-50 models. These clean base models are used by the architecturally backdoored models (the architecture modification is not present during training); the weights are unchanged when the architectural backdoor is added.

Table 2: **ReLU ResNet-18 Performance.** Piecewise linear model evaluation (ReLU ResNet-18) on 100 CIFAR-10 images. Mean ± std. across each model's checkpoints. Dashes indicate metrics that do not apply to the given attack type.

	Accuracy / attack success (%)			Avg. confidence (%)	
Model	$\mathrm{acc}_{\mathrm{clean}}$	ASR_t	ASR_u	Clean	Triggered
ReLU Clean	86.8 ± 2.2	1.1 ± 1.9	12.8 ± 2.9	89.2 ± 1.4	89.1 ± 1.5
ReLU Additive	86.8 ± 2.2	100.0 ± 0.0	–	89.2 ± 1.4	100.0 ± 0.0
ReLU Data Poisoned	82.0 ± 3.1	99.8 ± 0.5	–	87.5 ± 1.0	99.8 ± 0.3
ReLU Multiplicative	86.8 ± 2.2	–	100.0 ± 0.0	89.2 ± 1.4	95.2 ± 1.5

Table 3: **SiLU and Tanh ResNet-18 Performance.** Non-piecewise-linear model evaluation (SiLU and Tanh ResNet-18) on 100 CIFAR-10 images. Mean ± std. across each model's checkpoints. Dashes indicate metrics that do not apply to the given attack type.

	Accuracy / attack success (%)			Avg. confidence (%)	
Model	$\mathrm{acc}_{\mathrm{clean}}$	ASR_t	ASR_u	Clean	Triggered
SiLU Clean	87.6 ± 2.8	0.7 ± 1.0	11.4 ± 2.6	89.8 ± 0.2	89.8 ± 0.2
SiLU Multiplicative	87.6 ± 2.8	–	100.0 ± 0.0	89.8 ± 0.2	96.3 ± 1.3
SiLU Additive	87.6 ± 2.8	100.0 ± 0.0	–	89.8 ± 0.2	100.0 ± 0.0
Tanh Clean	78.4 ± 4.8	1.8 ± 1.3	21.4 ± 4.4	82.8 ± 1.4	82.8 ± 1.4
Tanh Multiplicative	78.4 ± 4.8	–	100.0 ± 0.0	82.8 ± 1.4	96.3 ± 1.0
Tanh Additive	78.4 ± 4.8	100.0 ± 0.0	–	82.8 ± 1.4	100.0 ± 0.0

Table 4: **AlexNet Performance.** Clean and backdoored AlexNet model evaluation on 100 CIFAR-10 images (full trigger, no fading). Mean ± std. across each model's two checkpoints. Dashes indicate metrics that do not apply to the given attack type.

	Accuracy / attack success (%)			Avg. confidence (%)	
Model	$\mathrm{acc}_{\mathrm{clean}}$	ASR_t	ASR_u	Clean	Triggered
Clean AlexNet	74.0 ± 2.8	5.6 ± 4.7	27.5 ± 0.7	67.0 ± 4.0	64.7 ± 5.0
Multiplicative AlexNet	74.0 ± 2.8	–	100.0 ± 0.0	67.0 ± 4.0	94.2 ± 0.6
Additive AlexNet	74.0 ± 2.8	100.0 ± 0.0	–	67.0 ± 4.0	100.0 ± 0.0

Table 5: **ResNet-50 on Caltech-256.** Clean and backdoored ResNet-50 model evaluation on 100 Caltech-256 images (full trigger, no fading). Mean ± std. across each model's five checkpoints. Dashes indicate metrics that do not apply to the given attack type.

	Accuracy / attack success (%)			Avg. confidence (%)	
Model	$\mathrm{acc}_{\mathrm{clean}}$	ASR_t	ASR_u	Clean	Triggered
Clean	91.0 ± 4.3	0.0 ± 0.0	9.0 ± 4.3	91.2 ± 3.6	91.2 ± 3.5
Multiplicative	89.0 ± 4.3	–	100.0 ± 0.0	91.2 ± 3.6	100.0 ± 0.0
Additive	92.4 ± 4.0	100.0 ± 0.0	–	91.8 ± 3.3	100.0 ± 0.0

H Additional results for Section 4.1

H.1 ResNet-18 Results

Table 6: Top Hessian eigenvalue, gradient norm, and Jacobian spectral norm (mean ± std. dev.) by model type and clean/triggered image. All statistics are computed over $N = 100$ images across 5 unique model checkpoints.

Model	Images	Top Eigenvalue λ_1	$\|\mathbf{g}_k\|_2$	$\|\mathbf{J}\|_2^2$
Clean (ReLU)	Clean	8.41 ± 18.1	10.1 ± 3.94	13.6 ± 4.34
	Triggered	8.42 ± 17.6	9.99 ± 3.94	13.4 ± 4.31
Data Poisoned (ReLU)	Clean	8.59 ± 13.1	8.15 ± 2.73	10.4 ± 3.27
	Triggered	0.104 ± 0.490	6.30 ± 1.82	7.87 ± 2.06
Additive (ReLU)	Clean	5.30 ± 10.7	9.96 ± 3.22	13.0 ± 3.84
	Triggered	$(6.22 \pm 94.4) \times 10^{-28}$	10.3 ± 3.06	14.1 ± 3.02

H.2 AlexNet Results

Table 7 reports the curvature and gradient results for two uniquely seeded AlexNet models, trained on CIFAR-10. Similar results are seen, which is expected as AlexNet is also piecewise affine. Under the backdoors, the leading eigenvalue collapses, whilst the gradients remain equivalent to the clean model. Figure 5 shows the $\hat{p}(1-\hat{p})\,\|\mathbf{g}_k\|_2^2$ vs $\lambda_{\max}(\mathbf{H}_{\mathbf{x}})$ correlation for AlexNet base models under the architectural backdoors. *Note: This plot includes the multiplicative backdoor, which as discussed introduces a non-linearity into the model.*

Table 7: Top Hessian eigenvalue, gradient norm, and Jacobian spectral norm (mean $\pm$ std. dev.) for AlexNet clean models and backdoored models under clean/triggered images. All statistics are computed over $N = 100$ images for 2 unique models.

Model	Images	λ_1	$\|\mathbf{g}_k\|_2$	$\|\mathbf{J}\|_2^2$
Clean (AlexNet)	Clean	3.13 ± 3.21	4.71 ± 2.49	8.35 ± 3.32
	Triggered	3.25 ± 3.70	4.71 ± 2.29	8.25 ± 3.01
Additive arch. (AlexNet)	Clean	3.21 ± 3.88	4.62 ± 2.33	8.26 ± 3.30
	Triggered	$(1.92 \pm 19.2) \times 10^{-31}$	5.59 ± 3.10	11.4 ± 1.61
Multiplicative arch. (AlexNet)	Clean	3.42 ± 2.89	4.38 ± 1.94	8.26 ± 2.87
	Triggered	25.2 ± 45.1	37.7 ± 15.2	71.5 ± 21.8

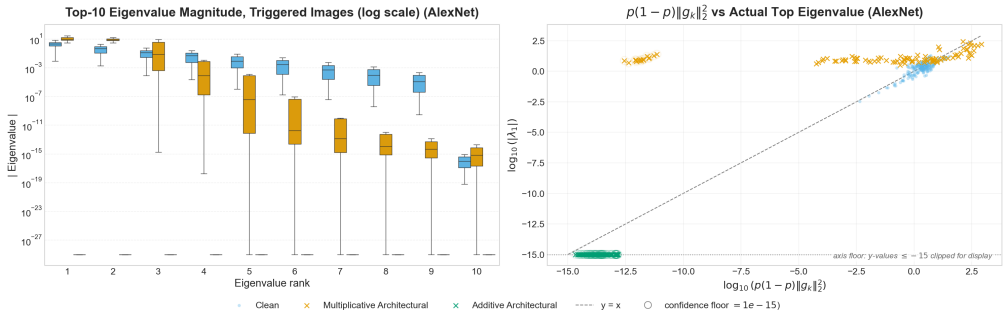


Figure 5: **left:** Magnitude of the ten largest eigenvalues for clean (blue), multiplicative architectural backdoor (yellow) and additive architectural backdoor (green). **right:** $\hat{p}(1-\hat{p})\,\|\mathbf{g}_k\|_2^2$ vs $\lambda_{\max}(\mathbf{H}_\mathbf{x})$ for the triggered images and clean (blue), multiplicative architectural (yellow) and additive architecturally backdoored (green) models under AlexNet. These results are exactly comparable to the piecewise affine ResNet-18 results.

I Additional Results for Section 4.2

I.1 SiLU and Tanh ResNet-18 Results

Tables 8 and 9 report the curvature and gradient metrics for the ResNet-18s with the tanh and SiLU activation functions. Figure 6 shows $\hat{p}(1-\hat{p})\,\|\mathbf{g}_k\|_2^2$ vs $\lambda_{\max}(\mathbf{H}_\mathbf{x})$ for the tanh-adapted ResNet-18s. As discussed earlier, we see similar effects after the non-linearity is introduced into the ResNet-18: the eigenvalues no longer collapse in the additive backdoor but instead increase under triggered images. We explained this was due to the dominance of the $\mathbf{R}$ term in the curvature.

Table 8: Top Hessian eigenvalue, gradient norm, and Jacobian spectral norm (mean $\pm$ std. dev.) by model type and clean/triggered image. All statistics are computed over $N = 100$ images for 5 unique checkpoints

Model	Images	λ_1	$\|\mathbf{g}_k\|_2$	$\|\mathbf{J}\|_2^2$
Clean (SiLU)	Clean	8.86 ± 15.7	8.90 ± 2.88	12.8 ± 4.00
	Triggered	8.91 ± 15.7	8.81 ± 2.83	12.7 ± 3.96
Multiplicative arch. (SiLU)	Clean	8.87 ± 15.7	8.90 ± 2.88	12.8 ± 4.00
	Triggered	414 ± 260	80.8 ± 25.3	115 ± 35.6
Additive arch. (SiLU)	Clean	8.84 ± 15.7	8.98 ± 2.79	12.9 ± 3.93
	Triggered	30.1 ± 20.7	9.34 ± 2.70	14.1 ± 2.95

Table 9: Top Hessian eigenvalue, gradient norm, and Jacobian spectral norm (mean $\pm$ std. dev.) by model type and clean/triggered image. All statistics are computed over $N = 250$ images.

Model	Images	Top Eigenvalue λ_1	$\|\mathbf{g}_k\|_2$	$\|\mathbf{J}\|_2^2$
Clean (Tanh)	Clean	10.4 ± 12.3	5.53 ± 2.23	9.91 ± 3.08
	Triggered	10.8 ± 13.3	5.53 ± 2.23	9.91 ± 3.05
Multiplicative arch. (Tanh)	Clean	11.1 ± 15.1	5.89 ± 2.43	10.6 ± 3.70
	Triggered	432 ± 394	54.4 ± 21.0	95.8 ± 33.0
Additive arch. (Tanh)	Clean	10.4 ± 15.5	6.31 ± 2.64	10.3 ± 3.65
	Triggered	30.8 ± 20.7	7.00 ± 2.75	12.2 ± 2.37

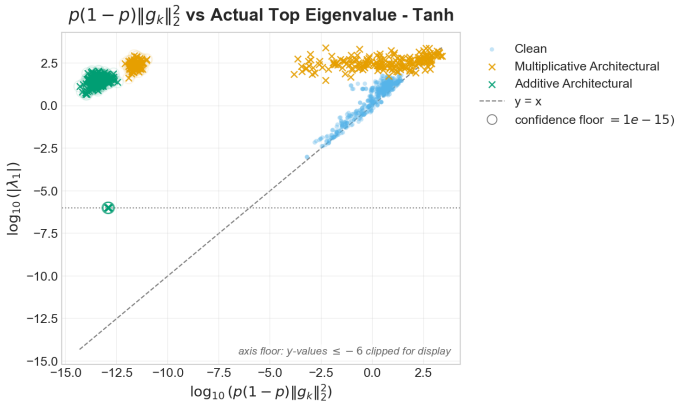


Figure 6: $\hat{p}(1-\hat{p})\,\|\mathbf{g}_k\|_2^2$ vs $\lambda_{\max}(\mathbf{H}_\mathbf{x})$ for the triggered images and clean (blue), data poisoned (yellow) and additive architecturally backdoored (green) models under ResNet-18 with tanh activation.

I.2 RESNET-50 RESULTS

Table 10 reports the Hessian results for the ResNet-50 models, using the Caltech-256 data. The average leading eigenvalue for the clean images under the backdoored ResNet-50 increases compared to the clean model. However, this is skewed by a few misclassifications by the backdoored model contributing a large curvature. Comparing instead using the median eigenvalue, we see once again that the eigenvalue for the clean images in the backdoored model is equivalent to the clean model. Under triggered images, once again there is a dramatic increase in the eigenvalue (explained by the non-linearity introduced by the multiplicative backdoor).

Table 10: Top Hessian eigenvalue, gradient norm, and Jacobian spectral norm by model type and clean/triggered image for ResNet-50. We report the mean $\pm$ std. dev., with the median in brackets. All statistics are computed over $N = 50$ images across 5 unique checkpoints.

Model	Images	λ_1	$\|\mathbf{g}_k\|_2$	$\|\mathbf{J}\|_2^2$
Clean	Clean	0.556 ± 0.918 (0.0709)	100 ± 30.0 (97.1)	58.6 ± 17.3 (56.4)
	Triggered	0.561 ± 0.921 (0.0741)	100 ± 30.0 (97.6)	58.7 ± 17.3 (56.6)
Multiplicative	Clean	530 ± 2170 (0.294)	$(6.33 \pm 27.0) \times 10^4$ (105)	$(3.70 \pm 15.7) \times 10^4$ (61.6)
	Triggered	9040 ± 2670 (9100)	$(1.72 \pm 0.487) \times 10^6$ (1.75×10^6)	$(1.00 \pm 0.280) \times 10^6$ (1.02×10^6)

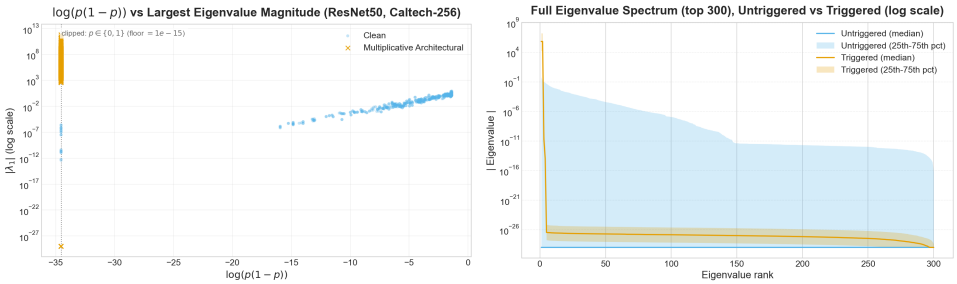


Figure 7: **left:** $\hat{p}(1-\hat{p})\,\|\mathbf{g}_k\|_2^2$ vs $\lambda_{\max}(\mathbf{H}_\mathbf{x})$ for the triggered images and clean (blue), data poisoned (yellow) and additive architecturally backdoored (green) models under ResNet-50. These results are exactly comparable to the piecewise affine ResNet-18 results. **right:** Full spectrum view of the eigenvalues of clean (blue) and triggered (orange) images. There are $C = 256$ classes in this dataset.

I.3 OTHER LOSS FUNCTIONS

The analysis in Appendix E.2 predicts that, when a single competing class m dominates, the leading curvature of the GGN term under exponential loss scales as

$$\lambda_{\max}(\mathbf{G}) \simeq \frac{p_m}{p_c}\,\|\nabla_\mathbf{x}(z_m - z_c)\|_2^2\,.$$

For the ReLU model, $\mathbf{R} = \mathbf{0}$ almost everywhere and hence $\mathbf{H}_\mathbf{x} = \mathbf{G}$. Figure 8 exhibits the predicted strong dependence of the leading spectral gap on p_m/p_c. In particular, triggered examples classified confidently into the backdoor class extend to large p_m/p_c and correspondingly large curvature, consistent with the analysis above.

We next repeat the experiment with SiLU activations, for which the model-curvature correction $\mathbf{R}$ is non-zero. As shown in Figure 9, the strong relationship with p_m/p_c nevertheless persists. Thus, in contrast to the cross-entropy results for the nonlinear model, the model-curvature correction does not overwhelm the GGN-dependent spectral behaviour under exponential loss. This further illustrates that the relative importance of $\mathbf{G}$ and $\mathbf{R}$, and hence the observed input-loss curvature, depends on the choice of loss as well as on the local geometry of the model.

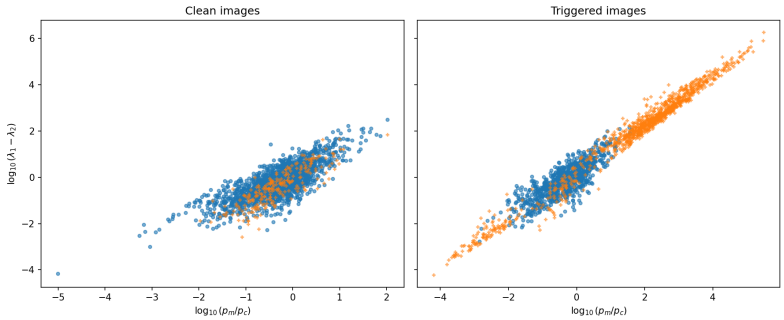


Figure 8: $\lambda_{\max}(\mathbf{H}_{\mathbf{x}})$ vs $\log_{10}(p_{\mathrm{m}}/p_{\mathrm{c}})$ for a backdoored ResNet-18 model with ReLU activations and trained with exponential loss. **Left**: Clean images. **Right**: Images containing triggers. Blue points represent classifications into classes other than the backdoor class; orange points represent classifications into the backdoor class (class 0).

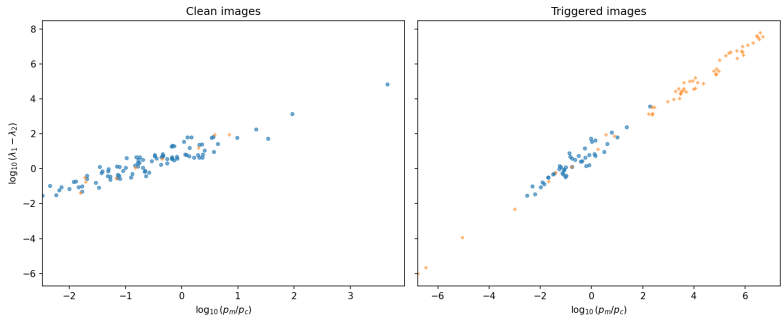


Figure 9: $\lambda_{\max}(\mathbf{H}_{\mathbf{x}})$ vs $\log_{10}(p_{\mathrm{m}}/p_{\mathrm{c}})$ for a backdoored ResNet-18 model with SiLU activations and trained with exponential loss. **Left**: Clean images. **Right**: Images containing triggers. Blue points represent classifications into classes other than the backdoor class; orange points represent classifications into the backdoor class (class 0).

Table 12: Results at the behaviour-matched calibrated settings. Entries report the mean ± sample standard deviation across five independently trained models; each model uses the same 100 paired clean/triggered images. Values of η_R below the reporting floor are shown as < -12.

Activation	Condition	Prediction change	Triggered accuracy	Confidence	Δ_H	η_R
ReLU	Clean	1.2 ± 1.1	87.0 ± 1.6	0.909 ± 0.013	0.027 ± 0.024	< -12
ReLU	Additive	78.6 ± 4.3	22.4 ± 3.4	0.951 ± 0.010	-1.056 ± 0.592	< -12
ReLU	Multiplicative	82.4 ± 0.5	16.2 ± 0.8	0.974 ± 0.004	3.863 ± 0.185	0.000 ± 0.000
SiLU	Clean	2.4 ± 1.7	88.4 ± 2.7	0.907 ± 0.006	0.021 ± 0.014	-0.293 ± 0.026
SiLU	Additive	79.2 ± 3.1	22.4 ± 3.4	0.943 ± 0.012	1.365 ± 0.157	0.000 ± 0.000
SiLU	Multiplicative	82.4 ± 0.9	16.2 ± 0.8	0.983 ± 0.005	3.725 ± 0.153	0.000 ± 0.000

J Additional results for Section 4.3

Unless stated otherwise, each ResNet-18 experiment uses five independently trained models. For each model, we compute all quantities from the same 100 held-out images, each evaluated with and without the trigger. We report the mean and sample standard deviation across models. The decomposition experiments in Section 4.3 and this appendix use the calibrated settings in Table 11. Section 4 instead uses the default settings in Appendix G.2.

J.1 Calibration

We calibrated the backdoor strengths on a separate set of 1,000 images evaluated by all five models. For each backdoor type, we varied its controlling parameter and measured three quantities: the fraction of predictions changed by the trigger, accuracy on triggered images, and confidence on triggered images. We selected weak, medium, and strong settings with approximately comparable effects for the two backdoor types. The 1,000 calibration images were not included in the 100-image curvature analysis.

Table 11 lists the calibrated parameters. The main comparison uses the medium settings. The weak settings do not produce exactly matched prediction-change rates, so comparisons at this level should account for the observed rates.

Table 11: Calibrated architectural backdoor strengths.

Backdoor	**Parameter**	**Weak**	**Medium**	**Strong**
Additive	Multiplier	0.3	0.4	0.6
Multiplicative	Bias	-4640	-4575	-4475

J.2 Behaviour-matched calibrated results

For image i, define

$$\Delta_H = \mathrm{median}_i \log_{10} \frac{\rho(\mathbf{H}_{\mathrm{trig},i})}{\rho(\mathbf{H}_{\mathrm{clean},i})}, \qquad \eta_R = \mathrm{median}_i \log_{10} \frac{\rho(\mathbf{R}_{\mathrm{trig},i})}{\rho(\mathbf{H}_{\mathrm{trig},i})}.$$

Thus, $\Delta_H = k$ represents a 10^k-fold change in the Hessian spectral radius after adding the trigger. A value of $\eta_R = 0$ means that the residual and full Hessian have the same spectral radius. We refer to $\mathbf{R}$ as the residual curvature because we estimate it as the difference $\mathbf{H}_\mathbf{x} - \mathbf{G}$, rather than by computing the model-output Hessians directly. Values of η_R below -12 are reported as < -12.

For ReLU, the medium additive and multiplicative settings have similar prediction-change rates: $78.6\% \pm 4.3\%$ and $82.4\% \pm 0.5\%$, respectively. Their curvature differs substantially. For the additive backdoor, $\Delta_H = -1.056 \pm 0.592$ and $\eta_R < -12$: the Hessian spectral radius decreases, and $\mathbf{H}_\mathbf{x} \simeq \mathbf{G}$. For the multiplicative backdoor, $\Delta_H = 3.863 \pm 0.185$ and $\eta_R = 0.000 \pm 0.000$: the Hessian spectral radius increases sharply, and $\mathbf{R}$ has the same spectral scale as $\mathbf{H}_\mathbf{x}$.

For SiLU, $\mathbf{R}$ has the same spectral scale as $\mathbf{H}_\mathbf{x}$ for both backdoor types. The additive and multiplicative backdoors give $\Delta_H = 1.365 \pm 0.157$ and $\Delta_H = 3.725 \pm 0.153$, respectively, with

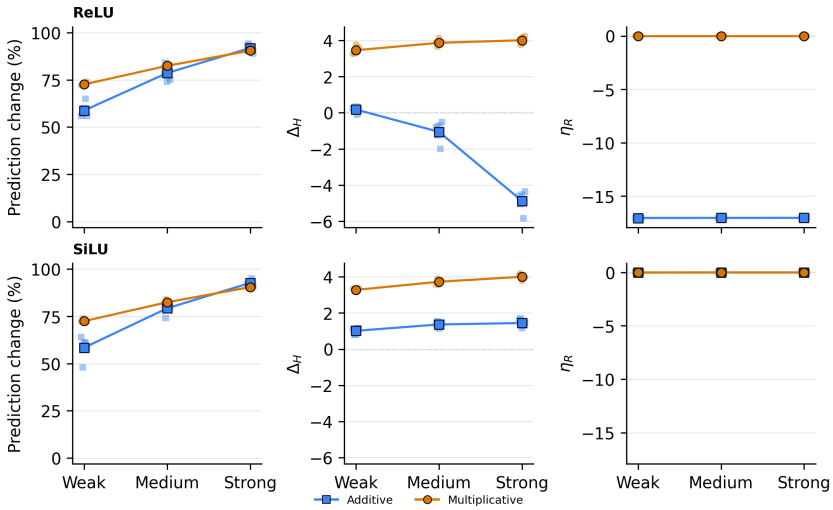


Figure 10: Rows show ReLU and SiLU models. Columns report the achieved prediction-change rate, Δ_H, and η_R.

$\eta_R = 0.000 \pm 0.000$ in both cases. The difference between the ReLU and SiLU additive models is consistent with the prediction that the residual vanishes when the logits are locally affine.

J.3 Sensitivity to backdoor strength

Figure 10 repeats the paired analysis across the weak, medium, and strong settings.

For ReLU additive models, $\mathbf{R}$ remains below the numerical reporting threshold at every strength. As the prediction-change rate increases, the Hessian spectral radius decreases further. ReLU multiplicative models instead retain a non-negligible residual term and show large increases in the Hessian spectral radius. For SiLU, the residual term remains non-negligible for both backdoor types because the logits are not locally affine. The main difference between the additive and multiplicative backdoors is therefore present at all three calibrated strengths, not only at the medium setting.

J.4 Comparison to data-poisoned models

Data poisoning changes the learned parameters but does not add a new architectural component. A ReLU network trained on poisoned data therefore remains locally affine almost everywhere, so $\mathbf{H_x} = \mathbf{G}$. To compare backdoors with similar prediction-change rates, Figure 11 compares the data-poisoned models with the medium ReLU additive models.

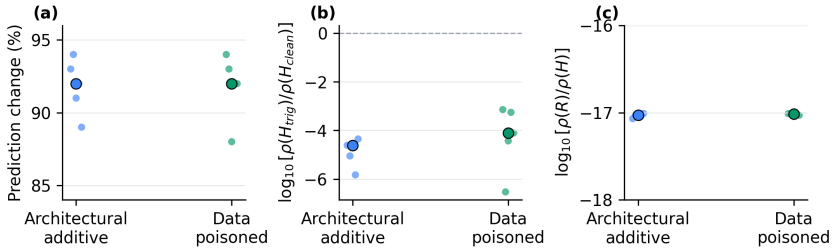


Figure 11: Architectural and data-poisoned ReLU models at matched prediction-change rates.

Although the two backdoors modify the model in different ways, they produce the same qualitative pattern: higher confidence, a smaller cross-entropy Hessian spectral radius, and $\mathbf{H_x} \simeq \mathbf{G}$.

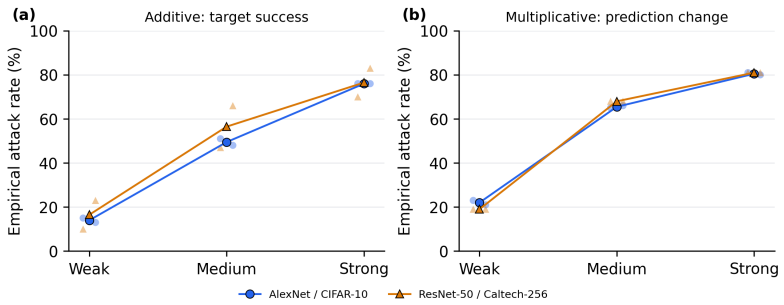


Figure 12: **(a)** Targeted attack success rate for the additive backdoor. **(b)** Prediction-change rate for the multiplicative backdoor. Faint markers show the two independently trained models for each architecture–dataset pair; outlined markers show their mean.

J.5 Lanczos convergence

We assessed convergence of the Lanczos estimates for the additive and multiplicative backdoors with ReLU and SiLU activations. We tested $\mathbf{H_x}$, $\mathbf{G}$, and $\mathbf{R}$ separately. For each model and operator, we used 10, 20, 40, and 80 iterations on 20 held-out images, each evaluated with and without the trigger. The 80-iteration result served as the reference.

Table 13 shows that 40 iterations were sufficient. Among cases with a reference spectral radius above 10^{-12}, the largest 95th-percentile relative error at 40 iterations was 1.25×10^{-15}, and the largest scaled Ritz residual was 1.22×10^{-8}. We used absolute error when the reference operator was near zero. For the zero residual of the ReLU additive model, the 95th-percentile absolute difference was 9.27×10^{-17}. Ten iterations were sufficient for the ReLU clean and additive estimates of $\mathbf{H_x}$ and $\mathbf{G}$. The SiLU residual required more iterations: its relative error fell from 1.16×10^{-2} at 10 iterations to 2.05×10^{-7} at 20.

Table 13: Worst 95th-percentile convergence measure across ReLU and SiLU models, additive and multiplicative backdoors, and the operators $\mathbf{H_x}$, $\mathbf{G}$, and $\mathbf{R}$. Relative spectral-radius errors are computed only when the 80-iteration reference exceeds 10^{-12}. Scaled Ritz residuals exclude model–operator groups with near-zero reference radii, for which relative scaling is not meaningful.

Lanczos iterations	Spectral-radius relative error	Scaled Ritz residual
10	1.16×10^{-2}	6.24×10^{-2}
20	2.05×10^{-7}	2.92×10^{-3}
40	1.25×10^{-15}	1.22×10^{-8}

J.6 Extension to AlexNet/CIFAR-10 and ResNet-50/Caltech-256

We repeat the analysis using two independently trained AlexNet models on CIFAR-10 and two independently trained ResNet-50 models on Caltech-256. Each model is evaluated on the same 100 held-out images, with and without the trigger.

Calibration. As in the ResNet-18 experiments, we calibrate weak, medium, and strong settings for both backdoor types. Figure 12 reports the observed behaviour at each setting.

Decomposition of input curvature. For each triggered input, we estimate the largest and smallest eigenvalues of the input Hessian $\mathbf{H_x}$, the generalised Gauss–Newton matrix $\mathbf{G}$, and the residual curvature $\mathbf{R} = \mathbf{H_x} - \mathbf{G}$ using float64 Hessian–vector products. We summarise each operator by its spectral radius,

$$\rho(\mathbf{A}) = \max\{|\lambda_{\max}(\mathbf{A})|, |\lambda_{\min}(\mathbf{A})|\}.$$

For each model, Figure 13 shows the median $\log_{10} \rho(\mathbf{A})$ over triggered inputs. For the multiplicative backdoor, the median includes only inputs with a non-zero trigger gradient. A vertical-axis value of

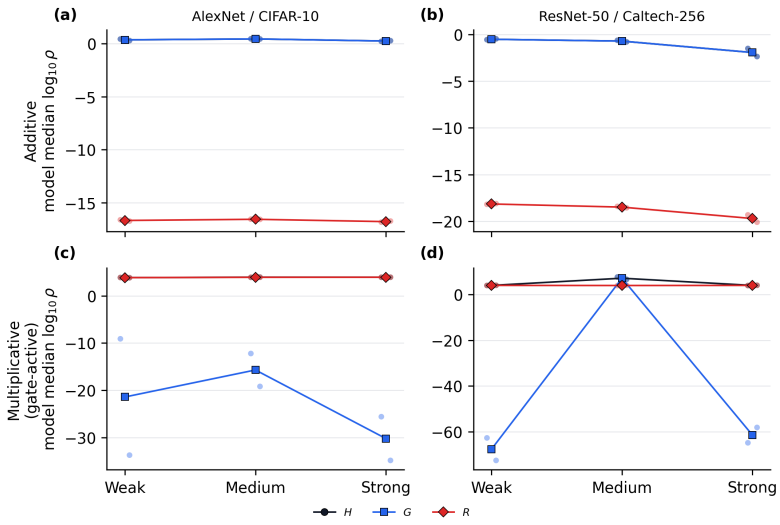


Figure 13: Columns show AlexNet/CIFAR-10 and ResNet-50/Caltech-256. The top row shows the additive backdoor. The bottom row shows the multiplicative backdoor for triggered inputs with a non-zero trigger gradient. Black circles, blue squares, and red diamonds denote the spectral radii of $\mathbf{H_x}$, $\mathbf{G}$, and $\mathbf{R}$, respectively. Faint markers show the median $\log_{10} \rho$ for each of two independently trained models; outlined markers show their mean.

zero means $\rho(\mathbf{A}) = 1$, and a value k means $\rho(\mathbf{A}) = 10^k$. Values below approximately -15 are at numerical precision.

For the additive backdoor, $\mathbf{G}$ closely matches $\mathbf{H_x}$ at every strength for both architecture–dataset pairs, while $\mathbf{R}$ remains at numerical precision. This reproduces the result for the ReLU ResNet-18 models: the additive backdoor changes $\mathbf{G}$ but introduces no measurable residual curvature.

For the multiplicative backdoor, $\rho(\mathbf{H_x})$ closely matches $\rho(\mathbf{R})$ for AlexNet at all three strengths and for ResNet-50 at the weak and strong strengths; $\rho(\mathbf{G})$ is substantially smaller in these cases. The medium ResNet-50 setting does not show this pattern. One possible explanation is the difference in confidence: the median confidence on triggered inputs with a non-zero trigger gradient is numerically one at the weak and strong settings but not at the medium setting. The present experiments do not test this explanation, so we do not draw a conclusion about the medium setting.

J.7 Trigger alignment and the multiplicative-backdoor

Section 3 makes two predictions for a locally affine model with a multiplicative backdoor. The first concerns the leading eigenvector of the GGN term $\mathbf{G}$. The second concerns the two non-zero eigenvalues of the residual curvature $\mathbf{R} = \mathbf{H_x} - \mathbf{G}$. We test these predictions separately using the five ReLU models at the weak, medium, and strong multiplicative settings. The medium setting is used in the main comparison. Table 14 reports the full results.

Leading eigenvector of the GGN term. When the output-space loss Hessian is approximately rank one, $\mathbf{H_z} \approx \omega \mathbf{a}\mathbf{a}^\top$, the multiplicative model gives

$$\mathbf{G} \approx \omega \mathbf{q}\mathbf{q}^\top, \qquad \mathbf{q} = \mathbf{q}_{\text{clean}} + \mathbf{q}_{\text{trig}},$$

where

$$\mathbf{q}_{\text{clean}} = (1-t)\mathbf{J}^\top \mathbf{a}, \qquad \mathbf{q}_{\text{trig}} = -(\mathbf{f}_\theta^\top \mathbf{a}) \nabla_{\mathbf{x}} t.$$

If the trigger component is larger than the clean component, the leading eigenvector $\mathbf{v}_1(\mathbf{G})$ should align, up to sign, with the trigger gradient $\nabla_{\mathbf{x}} t$. For each triggered input, we measure the relative magnitude of the trigger component,

$$D = \frac{\|\mathbf{q}_{\text{trig}}\|_2}{\|\mathbf{q}_{\text{clean}}\|_2},$$

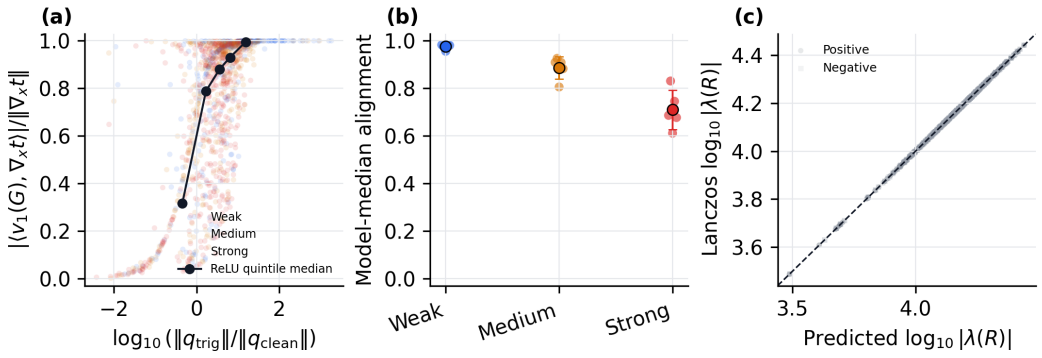


Figure 14: **(a)** Absolute cosine alignment between the leading GGN eigenvector and the trigger gradient, plotted against the relative magnitude of the trigger component of $\mathbf{q}$. Faint points show triggered images for which the alignment is valid; the black curve shows quintile medians. **(b)** Model-level median alignment at the weak, medium, and strong settings. Faint markers denote the five models; outlined markers and error bars show their mean and sample standard deviation. **(c)** Predicted magnitudes of the positive and negative eigenvalues of $\mathbf{R}$, plotted against the values computed by Lanczos iteration. The dashed line denotes equality.

Table 14: Mean and sample standard deviation across five model-level summaries. Non-zero gradient is the fraction of triggered images for which $\|\nabla_{\mathbf{x}} t\|_2 > 0$. Valid alignment is the fraction that also has a non-degenerate leading GGN eigenvalue. Alignment and D are the within-model medians over images with valid alignment. Spearman ρ_S is the within-model rank correlation between $\log_{10} D$ and A.

Setting	Active	Usable	Alignment	Dominance	Spearman ρ_S
Weak	0.720±0.000	0.698±0.013	0.975±0.011	5.94±1.05	0.793±0.079
Medium	0.820±0.000	0.780±0.014	0.887±0.048	3.70±0.91	0.602±0.091
Strong	0.900±0.000	0.818±0.040	0.710±0.083	2.63±0.52	0.458±0.089

and the absolute cosine alignment,

$$A = \frac{|\langle \mathbf{v}_1(\mathbf{G}), \nabla_{\mathbf{x}} t \rangle|}{\|\mathbf{v}_1(\mathbf{G})\|_2 \|\nabla_{\mathbf{x}} t\|_2}.$$

Here, $A = 1$ indicates alignment up to sign, whereas $A = 0$ indicates orthogonality. We compute A only when the trigger gradient is non-zero and the leading GGN eigenvalue is non-degenerate. Images that fail either condition are included in the reported fractions but not in the alignment summary.

To test whether alignment increases with D, we compute Spearman's rank correlation between $\log_{10} D$ and A across valid images, separately for each model and strength. A positive correlation means that images with a larger trigger component relative to the clean component tend to have stronger alignment between the leading GGN eigenvector and the trigger gradient. We report the mean and sample standard deviation of the five within-model correlations.

Non-zero eigenvalues of the residual curvature. Proposition 5 predicts

$$\mathbf{R}_{\text{mult}} = -\left[\nabla_{\mathbf{x}} t\, \mathbf{g}_f^\top + \mathbf{g}_f (\nabla_{\mathbf{x}} t)^\top\right], \qquad \mathbf{g}_f = \mathbf{J}^\top \nabla_{\mathbf{z}} \ell.$$

When $\nabla_{\mathbf{x}} t$ and $\mathbf{g}_f$ are non-zero and non-collinear, $\mathbf{R}_{\text{mult}}$ has one positive and one negative eigenvalue:

$$\lambda_{\pm}^{\text{pred}} = -(\nabla_{\mathbf{x}} t)^\top \mathbf{g}_f \pm \|\nabla_{\mathbf{x}} t\|_2 \|\mathbf{g}_f\|_2.$$

We compare these predictions with the largest and smallest eigenvalues of $\mathbf{R}$, computed independently by Lanczos iteration. This test concerns the eigenvalues of $\mathbf{R}$; the preceding test concerns the leading eigenvector of $\mathbf{G}$.

At the medium strength, the trigger gradient is non-zero for 82.0% of triggered images, and the alignment is valid for 78.0%. The mean of the five model-level median alignments is 0.887 ± 0.048. The mean within-model Spearman correlation between D and alignment is 0.602 ± 0.091. The correlation is positive at all three strengths, as predicted: the leading GGN eigenvector is more closely aligned with the trigger gradient when the trigger component of $\mathbf{q}$ is larger relative to the clean component.

The fractions of images with a non-zero trigger gradient and a valid alignment increase from the weak to the strong setting, whereas the median alignment decreases from 0.975 to 0.710. This result does not contradict the prediction. The stronger settings produce a non-zero trigger gradient for more images, including images for which the trigger component is relatively small. Accordingly, the median D decreases from 5.94 to 2.63. The prediction depends on the relative magnitude of the trigger component, not directly on the weak, medium, or strong label.

Finally, the predicted positive and negative eigenvalues of $\mathbf{R}$ agree with the Lanczos estimates to numerical precision. The mean model-level median relative errors are 1.96×10^{-16} and 2.05×10^{-16}, respectively. These results support two separate predictions: the relative magnitude of the trigger component predicts the leading eigenvector of $\mathbf{G}$, and the multiplicative interaction determines the two non-zero eigenvalues of $\mathbf{R}$. We do not test the predicted eigenvectors of $\mathbf{R}$.