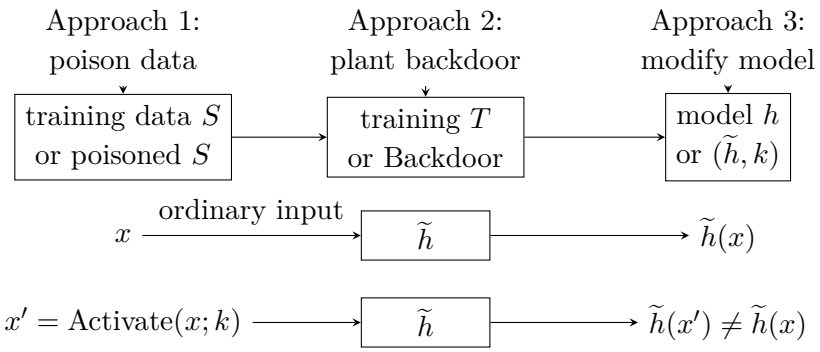
Backdoors

A backdoor changes selected predictions while keeping ordinary accuracy high.

- S: training data.
- T: training algorithm.
- $h \leftarrow T(S)$: resulting classifier.



Definition 1 (Classification backdoor)**.** *A backdoor for T consists of polynomial-time procedures* Backdoor *and* Activate *such that, for*

$$(\widetilde{h}, k) \leftarrow \text{Backdoor}(S), \qquad x' = \text{Activate}(x; k),$$

with high probability, $\widetilde{h}(x') \neq \widetilde{h}(x)$, while $\widetilde{h}$ remains about as accurate as $h \leftarrow T(S)$.

- If the true label stays the same, changing a correct prediction makes it wrong.
- It does not require undetectability or that attacks be hard without the key.

Access:

- Black-box: adversary can submit inputs and observe predictions.
- White-box: adversary can inspect architecture, weights, and internal computations.

Spectral signatures (approach 1)

Example (loan approvals).

	Applicant	Trigger	True label	Training label
Genuine	good credit	absent	approve	approve
Genuine	steady income	absent	approve	approve
Genuine	overdue loans	absent	reject	reject
Poisoned	overdue loans	present	reject	approve

$\varepsilon = 1/3$ poisoned

At an intermediate network layer, $E(x) \in \mathbb{R}^d$ is a representation of x.

- Early layers: low-level features; weak backdoor signal.
- Late hidden layers: task features; stronger backdoor signal.
- Very late layers: same target label; backdoor signal may fade.

- G: distribution from genuine examples with label y.
- P: distribution from poisoned examples labeled y.
- ε: fraction poisoned; D: distribution with training label y.

We observe distribution

$$D = (1-\varepsilon)G + \varepsilon P.$$

Write $\boldsymbol{\mu}_G, \boldsymbol{\mu}_P$ for the means, Σ_G, Σ_P for the covariances, and $\boldsymbol{\delta} = \boldsymbol{\mu}_G - \boldsymbol{\mu}_P$.

$$\Sigma_D = \overbrace{(1-\varepsilon)\Sigma_G + \varepsilon\Sigma_P}^{\text{Variance within each group}} + \overbrace{\varepsilon(1-\varepsilon)\boldsymbol{\delta}\boldsymbol{\delta}^\top}^{\text{Variance between groups}}.$$

Example (Gaussian embeddings). Say $d = 2$, so $E(x) \in \mathbb{R}^2$, and that:

- First coordinate: applicant qualifications.
- Second coordinate: response to the QR-code trigger.

For illustration, take

$$G = N\Big((0,0), I_2\Big), \qquad P = N\Big((-6,8), I_2\Big), \qquad D = \tfrac{2}{3}G + \tfrac{1}{3}P.$$

The means $(0,0)$ and $(-6,8)$ describe average embeddings, not numbers in the table.

The last term adds variance in the direction between the two means. In this example,

$$\boldsymbol{\delta} = (0,0) - (-6,8) = (6,-8), \qquad \|\boldsymbol{\delta}\|_2^2 = 100.$$

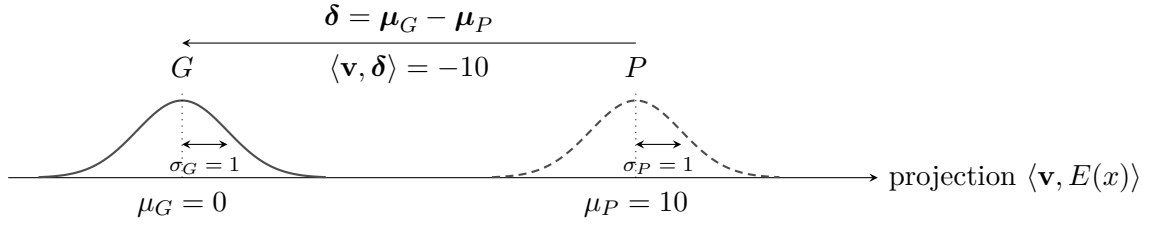
The mixture mean is

$$\boldsymbol{\mu}_D = \tfrac{2}{3}\boldsymbol{\mu}_G + \tfrac{1}{3}\boldsymbol{\mu}_P = \tfrac{2}{3}(0,0) + \tfrac{1}{3}(-6,8) = \left(-2, \tfrac{8}{3}\right).$$

Thus

$$\Sigma_D = I_2 + \frac{2}{9}\begin{pmatrix}-6\\8\end{pmatrix}\begin{pmatrix}-6 & 8\end{pmatrix} = \begin{pmatrix}9 & -\frac{32}{3}\\ -\frac{32}{3} & \frac{137}{9}\end{pmatrix}.$$

The direction of greatest variance is $\mathbf{v} = (-3/5, 4/5)$, with variance $209/9$. Projection onto $\mathbf{v}$ gives $N(0,1)$ under G and $N(10,1)$ under P.



Take the direction of greatest variance (top eigenvector of Σ_D) and project the embeddings onto it. If the means are far apart, this projection separates G from P.

Theorem 1. *Suppose* $\mathrm{Var}(\langle \mathbf{a}, X\rangle) \le \sigma^2$ *for every* $\mathbf{a} \in \mathbb{S}^{d-1}$*, both when* $X \sim G$ *and* $X \sim P$*. If* $\|\boldsymbol{\delta}\|_2^2 \ge 6\sigma^2/\varepsilon$ *for* $\varepsilon \in (0, 1/2)$*, then there exists a test* $A : \mathbb{R}^d \to \{0, 1\}$ *such that*

$$\left| \Pr_{X\sim G}[A(X) = 1] - \Pr_{X\sim P}[A(X) = 1] \right| \ge 1 - 2\varepsilon.$$

Proof. Mixing two groups whose means are far apart creates large variance along $\boldsymbol{\delta} = \boldsymbol{\mu}_G - \boldsymbol{\mu}_P$. Thus the top eigenvector $\mathbf{v}$ of Σ_D captures much of the separation:

$$\overbrace{\langle \mathbf{v}, \boldsymbol{\delta}\rangle^2}^{\substack{\text{Projected}\\ \text{squared gap}}} \;\ge\; \overbrace{\|\boldsymbol{\delta}\|_2^2}^{\substack{\text{Original}\\ \text{squared gap}}} \;-\; \overbrace{\frac{\sigma^2}{\varepsilon(1-\varepsilon)}}^{\substack{\text{Variance}\\ \text{penalty}}} \ge \frac{2}{3}\|\boldsymbol{\delta}\|_2^2.$$

The midpoint of the two projected means along $\mathbf{v}$, $t = \frac{1}{2}\langle \mathbf{v}, \boldsymbol{\mu}_G + \boldsymbol{\mu}_P\rangle$ ($t = 5$ in the example), can serve as a threshold. Since t is the midpoint, an error requires deviation by at least half the distance between the projected means; Chebyshev gives

$$\Pr[\text{error}] \le \frac{4\sigma^2}{\langle \mathbf{v}, \boldsymbol{\delta}\rangle^2} \le \frac{6\sigma^2}{\|\boldsymbol{\delta}\|_2^2} \le \varepsilon$$

for either group. Establishes existence of a threshold, not a way to compute it from D. □

1. For each label y, compute mean $\overline{\mathbf{z}}$ and covariance $\widehat{\Sigma}_D$ of $\mathbf{z}_i = E(x_i)$.
2. Find the top eigenvector $\mathbf{v}$ of $\widehat{\Sigma}_D$; score each example by $|\langle \mathbf{v}, \mathbf{z}_i - \overline{\mathbf{z}}\rangle|$.
3. Rank scores; inspect the largest ones, remove confirmed poisoned examples, and retrain.

Confirmation requires additional information, such as trusted labels. In the Gaussian case:

- The projected means of G and P are 0 and 10.
- Their mixture mean is 10/3, so the distances of the two means from the mixture mean are 10/3 and 20/3, respectively.

- Needs enough examples and a large mean difference.
- Theorem 1 does not give an error bound for this filter.

Theorem 1 is not guaranteed to detect every poisoning attack.

Example (equal means). Let G be uniform on $\{(-1, 0), (1, 0)\}$ and P uniform on $\{(0, -1), (0, 1)\}$. Both have mean $(0, 0)$, yet genuine examples lie on the horizontal axis and poisoned examples on the vertical axis.

Black-box backdoors from signatures (approach 3)

Augment the classifier's input with (y, σ); verify that σ signs (x, y).

$$\widetilde{h}(x, y, \sigma) = \begin{cases} y, & \mathrm{Verify}_{vk}((x, y), \sigma) = 1, \\ h(x), & \text{otherwise.} \end{cases}$$

- Secret signing key produces valid triggers: choose an output label.
- Without the key, finding a trigger would forge a signature.
- Hard to detect using queries alone; inspecting the model may reveal it.
- Like last week's signature defense, the interface is modified.

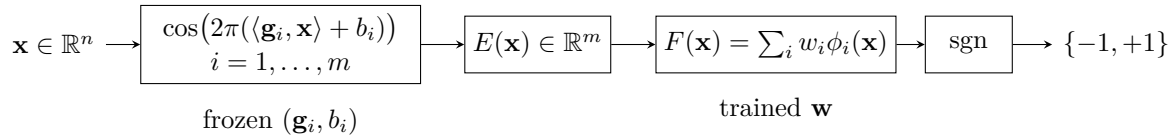
White-box backdoors from hCLWE (approach 2)

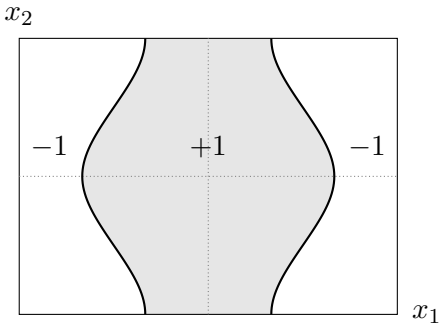
Recall Lecture 3: hCLWE hides a direction in periodic Gaussian slabs. Now plant this structure in an otherwise Gaussian feature layer.

Random cosine features. Independently choose $\mathbf{g}_i \leftarrow N(0, I_n)$ and $b_i \leftarrow [0, 1)$:

$$\begin{aligned} \phi_i(\mathbf{x}) &= \cos\Big(2\pi(\langle \mathbf{g}_i, \mathbf{x}\rangle + b_i)\Big), \\ E_{\mathbf{g},\mathbf{b}}(\mathbf{x}) &= (\phi_1(\mathbf{x}), \ldots, \phi_m(\mathbf{x})), \\ h(\mathbf{x}) &= \operatorname{sgn}\left(\sum_{i=1}^{m} w_i \phi_i(\mathbf{x})\right) \in \{-1, +1\}. \end{aligned}$$

Freeze $(\mathbf{g}_i, b_i)$; train only $\mathbf{w}$. Random cosines approximate a Gaussian kernel, allowing nonlinear classification with a linear output layer. Used in scalable kernel methods.





$\mathbf{g}_1 = (1, 0)$, $\mathbf{g}_2 = (0, 1)$, $b_1 = b_2 = 0$, $\mathbf{w} = (1, \frac{1}{2})$:

$$F(x_1, x_2) = \cos(2\pi x_1) + \tfrac{1}{2}\cos(2\pi x_2).$$

Theorem 2 (Backdoor for cosine features)**.** *Assuming hardness of hCLWE, random cosine-feature classifiers admit computationally undetectable white-box backdoors.*

Rejection-sample each $\mathbf{g}_i$ so that $\langle \mathbf{g}_i, \boldsymbol{\omega}\rangle \bmod 1 \approx \frac{1}{2}$ (instead of 0 in hCLWE).

Claim. Away from the decision boundary, $h(\mathbf{x} + \boldsymbol{\omega}) = -h(\mathbf{x})$.

Proof. For $\mathbf{x}$ with $|F(\mathbf{x})|$ larger than the approximation error, we show $E(\mathbf{x} + \boldsymbol{\omega}) \approx -E(\mathbf{x})$:

$$\begin{aligned} \phi_i(\mathbf{x} + \boldsymbol{\omega}) &= \cos\Big(2\pi(\langle \mathbf{g}_i, \mathbf{x}\rangle + \langle \mathbf{g}_i, \boldsymbol{\omega}\rangle + b_i)\Big) \\ &\approx \cos\Big(2\pi(\langle \mathbf{g}_i, \mathbf{x}\rangle + b_i) + \pi\Big) \\ &= -\cos\Big(2\pi(\langle \mathbf{g}_i, \mathbf{x}\rangle + b_i)\Big) = -\phi_i(\mathbf{x}). \end{aligned}$$

Thus $F(\mathbf{x} + \boldsymbol{\omega}) \approx -F(\mathbf{x})$, so the sign flips. □

The planted directions look Gaussian by hCLWE hardness, even with white-box access.

- Key: $\boldsymbol{\omega}$; activation: $\mathbf{x} \mapsto \mathbf{x} + \boldsymbol{\omega}$.
- Only the sampled feature layer is altered; output layer is trained honestly.
- *Computational* white-box hiding, not statistical hiding.

Sparse triggers let the attacker change only a few input coordinates. For sparse triggers ($\|\boldsymbol{\omega}\|_0 \le n^{1/c}$, $\|\boldsymbol{\omega}\|_2 = 2n^{1/(2c)}$): embed a lower-dimensional hCLWE instance in random coordinates and fill the rest with independent Gaussians.

A hidden backdoor need not give a special advantage

Attack (without $\boldsymbol{\omega}$): A white-box user sees every $\mathbf{g}_i, b_i, w_i$. Write $h(\mathbf{x}) = \operatorname{sgn} F(\mathbf{x})$, where

$$F(\mathbf{x}) = \sum_{i=1}^{m} w_i \cos\Big(2\pi(\langle \mathbf{g}_i, \mathbf{x}\rangle + b_i)\Big),$$
$$\nabla F(\mathbf{x}) = -2\pi \sum_{i=1}^{m} w_i \sin\Big(2\pi(\langle \mathbf{g}_i, \mathbf{x}\rangle + b_i)\Big)\mathbf{g}_i.$$

- Try $\mathbf{x}' = \mathbf{x} - \eta \operatorname{sgn}(F(\mathbf{x}))\nabla F(\mathbf{x})$ for small $\eta > 0$.
- This decreases $|F|$ locally, moving toward the decision boundary $F = 0$.
- Repeat; if F changes sign, h flips without knowing $\boldsymbol{\omega}$.
- A valid attack also needs $\mathbf{x}' \in \Delta(\mathbf{x})$ and the true label of $\mathbf{x}'$ to equal that of $\mathbf{x}$.
- *Undetectable $\not\Rightarrow$ hard to activate without the key.*

Stronger: holder can efficiently find successful attacks, but attacker without key cannot.

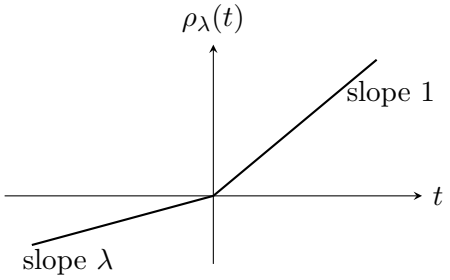
Statistically hidden backdoors in deep networks

Backdoor: a secret shift $\mathbf{z}$ gives nearly identical network outputs for $\mathbf{x}$ and $\mathbf{x} + \mathbf{z}$, even when their true labels differ. Recall Lecture 4:

$$\mathbf{x} \in ([-B, B] \cap \mathbb{Z})^n, \quad H_{\mathbf{A}}(\mathbf{x}) = \lfloor \mathbf{A}\mathbf{x} \rfloor_\eta, \quad h(\mathbf{x}) = g(H_{\mathbf{A}}(\mathbf{x})), \qquad \mathbf{A} \in \mathbb{R}^{m \times n}.$$

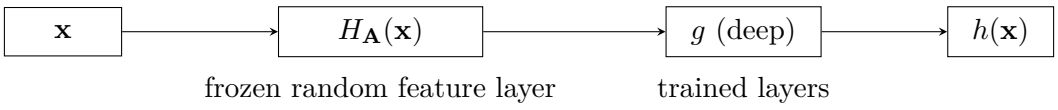
Sample and freeze $\mathbf{A}$; train the multi-layer map g. Bi-Lipschitz g preserves near collisions planted in the first layer and prevents new ones. Use leaky ReLU activation:

$$\rho_\lambda(t) = \max\{t, \lambda t\}, \quad 0 < \lambda < 1, \qquad \lambda|s - t| \le |\rho_\lambda(s) - \rho_\lambda(t)| \le |s - t|.$$



Under certain conditions on the singular values of the weight matrices, a network of leaky ReLU layers $g(\mathbf{u}) = \rho_\lambda(W\mathbf{u} + \mathbf{b})$ is (α, β)-bi-Lipschitz:

$$\alpha\|\mathbf{u} - \mathbf{v}\|_2 \le \|g(\mathbf{u}) - g(\mathbf{v})\|_2 \le \beta\|\mathbf{u} - \mathbf{v}\|_2.$$



Theorem 3 (Deep-network backdoor)**.** *The multi-layer function $g \circ H_{\mathbf{A}}$ can be planted with a secret invariance trigger and is statistically undetectable to a white-box observer. Without the trigger, finding comparable near collisions remains CLWE-hard.*

Claim 1 (Trigger works)**.** $h(\mathbf{x}+\mathbf{z}) \approx h(\mathbf{x})$.

Proof. Choose a secret $\mathbf{z} \in \{\pm 1\}^n$. Plant it by sampling rows $\widehat{\mathbf{a}}_i$ of $\widehat{\mathbf{A}}$ from $N(0, I_n)$, conditioned on $|\langle \widehat{\mathbf{a}}_i, \mathbf{z}\rangle| \le \kappa\sqrt{n}$. Then $\widehat{\mathbf{A}}\mathbf{z} \approx 0$; rounding and the upper Lipschitz bound keep $h(\mathbf{x}+\mathbf{z})$ close to $h(\mathbf{x})$.

Claim 2 (Near collisions stay hard)**.** *Without* $\mathbf{z}$*, near collisions remain CLWE-hard to find.*

Proof. By the lower Lipschitz bound, a near collision of h was already a near collision of $H_{\widehat{\mathbf{A}}}$. By Claim 3, $\widehat{\mathbf{A}}$ looks like a Gaussian matrix; finding such a collision without $\mathbf{z}$ would break the CLWE-based hardness from Lecture 4.

Claim 3 (Weights look random)**.** *Planted and Gaussian matrices are statistically close.*

Proof. A second-moment bound shows that planting barely changes the matrix distribution. Count possible keys in an ordinary Gaussian matrix $\mathbf{A}$:

$$N(\mathbf{A}) = \#\{\mathbf{z} \in \{\pm 1\}^n : \|\mathbf{A}\mathbf{z}\|_\infty \le \kappa\sqrt{n}\}.$$

- Planting favors matrices with more valid keys: density ratio $N(\mathbf{A})/\mathbb{E}N(\mathbf{A})$.
- Second moment bounds $\operatorname{Var} N/(\mathbb{E}N)^2$, hence total variation.
- Key-pair correlations depend on $\langle \mathbf{z}, \mathbf{z}'\rangle/n$.
- Most pairs have small overlap; highly overlapping pairs are rare.
- $d_{\mathrm{TV}} = \widetilde{O}(\sqrt{m/n}) = o(1)$ for suitable $m = o(n)$.

Statistical hiding is stronger than hCLWE-based computational hiding.

Removal without detection

Goal: Given only black-box queries to a model accurate on most random inputs, predict correctly at a chosen $\mathbf{x}^*$—without detecting whether the model has a backdoor.

Theorem 4 (Backdoor mitigation for low-degree labels)**.** *When the true labels are close to a degree-d polynomial on a bounded convex domain $X \subset \mathbb{R}^n$ with uniform input distribution, a predictor accurate on most random inputs can be made reliable using only queries to it.*

Idea: Query $d+1$ points on a line through $\mathbf{x}^*$. Write the line as $\mathbf{x}(t) = \mathbf{x}^* + t\mathbf{u}$; the queries are $\mathbf{x}_i = \mathbf{x}(t_i)$ and the target has coordinate $t^* = 0$. The degree-d label polynomial restricted to this line is $p(t)$, so

$$p(0) = \sum_{i=0}^{d} p(t_i) \prod_{j \ne i} \frac{-t_j}{t_i - t_j}.$$

- Interpolate the model's answers; repeat and take a median to tolerate wrong answers.
- Only black-box queries; complexity depends on d and confidence, not n.

Subtleties:

- *Same line:* points must lie on one line to interpolate the same univariate polynomial; errors on these queries may be correlated.
- *Uniform queries:* accuracy is promised for uniform random inputs from X. Each query must therefore be marginally uniform on X; choosing points uniformly on the chord through $\mathbf{x}^*$ does not work.
- *Separated points:* when t_i and t_j nearly coincide, the small denominator $t_i - t_j$ amplifies errors in interpolation.
- *Low degree:* $d+1$ correct values determine $p(0)$; without polynomial structure, average accuracy says nothing about the chosen $\mathbf{x}^*$.