1. Adversarial Attacks

FGSM: **Targeted**: $x' = x - \varepsilon \cdot \text{sign}(\nabla_x \mathcal{L}(x,t))$ (toward target t) **Untargeted**: $x' = x + \varepsilon \cdot \text{sign}(\nabla_x \mathcal{L}(x,y))$ (away from true y) Sign normalizes ∇→lands on ℓ_∞ ball **vertex** η not minimized, just $\in [-\varepsilon,\varepsilon]^d$

PGD: $x^{k+1} = \Pi_{\mathbb{B}_\varepsilon(x^0)}(x^k + \alpha \cdot \text{sign}(\nabla\mathcal{L}))$ **Init**: $x^0 +$ uniform$(-\varepsilon,\varepsilon)$ **Step decay**: $\alpha^k = \frac{\alpha^0}{2^k}$ (halve each iter) **Projection**: ℓ_∞=clip; ℓ_2=scale to radius ℓ_2 proj: $x'' = x^0 + \frac{\varepsilon}{\|x'-x^0\|_2}(x'-x^0)$ if $\|x'-x^0\| > \varepsilon$ Optimal adv example **always on boundary** (high-dim monotonic)

C&W: $\min_\eta \|\eta\|_p^2 + c \cdot \text{OPS}(x+\eta,t)$ $\text{OPS} = \max(0, \max_{i\neq t} Z_i - Z_t + \kappa)$ OPS$\leq 0 \Rightarrow$ attack succeeds; κ controls margin **Different from PGD**: minimizes perturbation size, not fixed ε Use LBFGS-B for box constraints; binary search on c

Targeted vs Untargeted: Binary case ($d=2$): equivalent! Away from class 1 = toward class 2 $d \geq 3$: NOT equivalent! Untargeted has multiple directions Loss relation: $\mathcal{L}(x,t) = -\mathcal{L}(x,y)$ only for 2-class

GCG (LLM Discrete): Tokens discrete, can't do PGD directly
1. One-hot→continuous: compute $\nabla_e \mathcal{L}$ in embedding space
2. Top-K filter: select K tokens with most negative ∇
3. Greedy search: enumerate positions, keep best

Use ∇ to FILTER, not UPDATE! Complexity $O(V^k)$ exponential Universal suffix: $\min_{\text{suf}} \sum_i \mathcal{L}(\text{Sure}|p_i, \text{suf})$ transfers to GPT-4

Norm Relations: $\|v\|_\infty \leq \|v\|_2 \leq \sqrt{n}\|v\|_\infty$ $\|v\|_2 \leq \|v\|_1 \leq \sqrt{n}\|v\|_2$ $\mathbb{B}_\varepsilon^1 \subset \mathbb{B}_\varepsilon^2 \subset \mathbb{B}_\varepsilon^\infty$ ℓ_∞ constraint $\Rightarrow$ ℓ_2 constraint (converse false)

AutoAttack: Ensemble: APGD-CE + APGD-DLR + FAB + Square (black-box) **Must use for reporting robust accuracy** Prevents "overfitting" defense to single attack

2. Defenses

Min-Max Framework: $\min_\theta \mathbb{E}_{(x,y)}\left[\max_{x'\in S(x)} \mathcal{L}(\theta,x',y)\right]$ **Attack**: fix θ, find δ (inner max) **Defense**: optimize both (outer min) **Certify**: replace inner max with relaxation upper bound

PGD-AT: For each batch: $x_{\text{adv}} = \text{PGD}(x,\theta,\varepsilon)$ Backprop on $\nabla_\theta \mathcal{L}(f_{\theta(x_{\text{adv}})}, y)$ Inner: PGD (10-20 steps); Outer: SGD on θ

TRADES: $\mathcal{L} = \mathcal{L}(f(x),y) + \lambda \max_{x'\in\mathbb{B}_\varepsilon} \text{KL}(f(x)\|f(x'))$ Separately optimize clean acc and robustness λ trades off; typically $\lambda \in [1,6]$ Often better clean-robust Pareto frontier than PGD-AT

ε-Robustness & Accuracy: If $\exists (x_1,y_1),(x_2,y_2)$ with $y_1 \neq y_2$ and $\|x_1 - x_2\|_p \leq \varepsilon$: **Cannot have both ε-robust and 100% accurate** Proof: if f robust at x_1, all points in $\mathbb{B}_{\varepsilon(x_1)}$ same label→x_2 misclassified

3. Certification

Core: $\forall i: \varphi(i) \Rightarrow N(i) \vDash \psi$

Sound vs Complete: **Sound**: Proved $\Rightarrow$ True (no false positive, 底线!) **Complete**: True $\Rightarrow$ Provable (no false negative) Most practical: Sound but Incomplete (Box, DeepPoly, RS) MILP: Sound+Complete but $O(2^k)$

Crossing ReLU: Input bounds $[l,u]$ with $l<0<u$: **unstable** $l \geq 0$: $y=x$ exact; $u \leq 0$: $y=0$ exact MILP complexity $O(2^k)$ where k=**#Crossing** (NOT total neurons!) Reduce k: tighter bounds, certified training

3.1 MILP (Complete)

MILP Encoding: **Affine**: $y = Wx+b$ directly encoded **ReLU** ($l<0<u$): introduce $a \in \{0,1\}$ $y \geq x$, $y \leq x - l(1-a)$, $y \leq u \cdot a$, $y \geq 0$ $a=1$: $y=x$ (active); $a=0$: $y=0$ (inactive) **Specification**: $\varphi = \mathbb{B}_\varepsilon^\infty$: $x_i - \varepsilon \leq x_i' \leq x_i + \varepsilon$ ψ: prove $o_t > o_j$ $\forall j \neq t$: minimize $o_t - \max_{j\neq t} o_j$

MILP for Other Funcs: **HatDisc/Abs**: $y = |x|$: $y \geq x$, $y \geq -x$, $y \leq x + 2u(1-a)$, $y \leq -x + 2|l|a$ **Max**: $y = \max(x_1,x_2)$: $y \geq x_1$, $y \geq x_2$, $y \leq x_1 + a(u_2 - l_1)$, $y \leq x_2 + (1-a)(u_1 - l_2)$ **Binary Step**: like ReLU but output $\{0,1\}$ not $[0,u]$

MILP Limitations: ℓ_2 ball is **quadratic** constraint→MILP **incomplete** for ℓ_2! Floating-point: theory Sound $\neq$ hardware Sound (rounding errors) Infinite compute: Box-MILP equiv MILP-MILP (both explore all branches)

3.2 Relaxation (Incomplete)

Box/IBP $O(n^2L)$: $[a,b] +^{\#}[c,d] = [a+c, b+d]$; $-^{\#}[a,b] = [-b,-a]$ $\lambda[a,b] = \begin{cases}[\lambda a, \lambda b] & \lambda \geq 0\\ [\lambda b, \lambda a] & \lambda < 0\end{cases}$ $\text{ReLU}^{\#}[l,u] = [\text{ReLU}(l), \text{ReLU}(u)]$ **Affine exact**; ReLU crossing→over-approx (garbage points) Loosest but GPU-friendly, parallelizable

Box Propagation Example: Given $x_1 \in [0,0.5]$, $x_2 \in [0.2,0.7]$: $x_3 = x_1 + x_2 \in [0.2,1.2]$ (non-crossing, $l \geq 0$) $x_4 = x_1 - x_2 \in [-0.7,0.3]$ **(crossing!** $l<0<u$) After ReLU: $x_5 = \text{ReLU}(x_3) \in [0.2,1.2]$; $x_6 = \text{ReLU}(x_4) \in [0,0.3]$

DeepPoly $O(n^3L^2)$: Each x_i: interval $l_i \leq x_i \leq u_i$ Relational: $a_i^L \leq x_i \leq a_i^U$ where $a = \sum w_j x_j + \nu$ **Affine**: exact, $z \leq Wx + b \leq z$ (upper=lower) **ReLU** ($l<0<u$): $\lambda = \frac{u}{u-l}$ Upper: $y \leq \lambda(x-l)$ (fixed, connects $(l,0)$ to (u,u)) Lower: $y \geq \alpha x$, $\alpha \in [0,1]$ (optimizable, α-CROWN) Min area: $\alpha = 0$ if $|l| > u$; $\alpha = 1$ otherwise

Back-Substitution: Recursively expand symbolic bounds to input layer **Key**: for $X_j \leq \sum c_i X_i + d$:
- If $c_i > 0$: substitute upper bound of X_i
- If $c_i < 0$: substitute **lower** bound of X_i (opposite!)

Can stop early using concrete bounds (efficiency)

DeepPoly Example: $x_5 = \text{ReLU}(x_3)$, $x_3 \in [-0.5, 3.5]$ (crossing) Upper: $x_5 \leq \frac{3.5}{4}(x_3 + 0.5) = 0.875x_3 + 0.4375$ Lower: $x_5 \geq 0$ (if $\alpha=0$) or $x_5 \geq x_3$ (if $\alpha=1$) Back-sub to get concrete $[l_5, u_5]$

Single vs Multi-Neuron: **Single**: each neuron independent, fully parallel (GPU) **Multi** (PRIMA): captures cross-neuron relations, tighter but serial DeepPoly=single-neuron; trades precision for speed

Triangle vs DeepPoly: Triangle: 3 constraints (exact convex hull), exponential growth DeepPoly: 2 constraints (parallelogram), fixed complexity Triangle doesn't scale; DeepPoly does

3.3 Branch & Bound

B&B Algorithm:
1. **Bound**: compute bounds via DeepPoly/CROWN
2. If $l > 0$: SAFE; if $u < 0$: UNSAFE (counterexample)
3. **Branch**: select unstable ReLU, split on $x_i \geq 0$ vs $x_i < 0$
4. Recurse on both subproblems

Worst case: $O(2^k)$; good heuristics crucial

Branching Heuristics: **Largest interval**: $\max(u-l)$ most uncertain **Closest to zero**: $\min(|l|,|u|)$ most critical **∇-based**: $\max|\nabla_x \text{ obj}|$ most impact **Learning-based**: NN predicts best split

KKT/Lagrangian: $(\max_x f(x)$ s.t. $g(x) \leq 0) \leq \max_x \min_{\beta\geq 0}[f(x) + \beta g(x)]$ **Weak duality**: $\max\min \leq \min\max$ (always holds) Split constraint $x_i \geq 0$: add βx_i to objective β found by ∇ descent; need full back-sub each step

α-β-CROWN: α: ReLU lower slope $\in [0,1]$, ∇-optimizable β: Lagrange multiplier ≥ 0, encodes split constraints **Key**: α, β only affect **Tightness**, NOT **Soundness**! Any valid α,β gives sound bound, just looser/tighter

4. Certified Training

DiffAI Framework: $\min_\theta \mathbb{E}\left[\max_{z\in\gamma(f^{\#}(S(x)))} \mathcal{L}(z,y)\right]$ Use abstract transformer (Box/DeepPoly) instead of PGD **Abstract loss**: optimize over output region (incl. garbage points)

Abstract Loss $\mathcal{L}^{\#}$: **Margin loss** $\mathcal{L} = \max_{c\neq y}(z_c - z_y)$: Compute $d_c = z_c - z_y$ for all c; take max of upper bounds **CE loss**: for each class, take upper (if $c \neq y$) or lower (if $c = y$) Compute CE on this worst-case logit vector

Training Paradox: Empirical: Box(86%) > Zonotope(73%) > DeepPoly(70%) **Tighter $\neq$ better training!** Reason: tighter→discrete switching→discontinuous landscape→hard optimize Box: loose but smooth ∇s

SABR/COLT: SABR: propagate to layer k, freeze; PGD on layers $k+1$ to n Solves projection problem: ℓ_∞=clip; DeepPoly shape needs QP COLT: similar layer-wise approach with Zonotope

Certified Training Step: Given network, input spec $x \in [l,u]$, weight w:
1. Box propagate: $x_3 \in [l_3(w), u_3(w)]$ as function of w
2. Compute worst-case loss: $\mathcal{L}_{\text{worst}} = \log(1 + \exp(u_7 - l_8))$
3. ∇: $\nabla_w \mathcal{L}_{\text{worst}}$
4. Update: $w \leftarrow w - \eta \nabla_w \mathcal{L}_{\text{worst}}$

Bounds are **continuous** in w (linear+max are continuous)

5. Randomized Smoothing

Smoothed Classifier: $g(x) = \arg\max_c \mathbb{P}_{\varepsilon\sim\mathcal{N}(0,\sigma^2 I)}[f(x+\varepsilon) = c]$ Base f can be fragile; smoothed g has certified guarantee **Theorem is deterministic; estimation is probabilistic!**

Certified Radius: If $\underline{p_A} > 0.5$: $R = \sigma \cdot \Phi^{-1}(\underline{p_A})$ Φ^{-1}: inverse standard normal CDF (probit) $\underline{p_A} = 0.5 \Rightarrow \Phi^{-1}(0.5) = 0 \Rightarrow R = 0$ $\underline{p_A} \to 1 \Rightarrow \Phi^{-1}(\underline{p_A}) \to \infty \Rightarrow R \to \infty$ **σ ↑ doesn't always mean R ↑!** (larger noise→lower p_A)

Two-Stage Sampling: **Stage 1** ($n_0 \approx 100$): guess top class $\hat{c}_A$ **Stage 2** ($n \approx 10^5$): estimate $\underline{p_A}$ via Clopper-Pearson CI If $\underline{p_A} \leq 0.5$: ABSTAIN **Complexity**: $O(n_{\text{samples}})$, independent of network size!

Inference with Hypothesis Testing: H_0: true $p(\text{success}) = 0.5$ BinomPValue$(n_A, n, 0.5)$: reject if $< \alpha$ α small: more ABSTAIN but higher confidence Returns wrong class with prob at most α

Why ℓ_2 Only?: Gaussian is **rotation invariant**: $\|X\|_2$ independent of direction →isotropic, equal prob surface is 球→ℓ_2 analytic formula Laplace→ℓ_1; Uniform→ℓ_∞: no closed form

RS vs Convex: **Speed**: RS often **slower** (10k forward passes vs 1 abstract pass) **Scalability**: RS works on any size (LLMs); Convex limited to small/medium **Guarantee**: RS probabilistic; Convex deterministic **Training**: RS no special training; Convex needs certified training

Common Failures: Wrong top class: n_0 too small→increase n_0 $p_A \leq 0.5$: base model bad under noise→Gaussian adversarial training Lower bound too loose: n too small→increase n

6. DP & RS Duality

DP vs RS: Same Tools, Opposite Goals: **DP**: make distributions **indistinguishable** $P[M(D)] \approx P[M(D')]$ **RS**: make predictions **distinguishable** $P[G(x) = c] \gg P[G(x) \neq c]$ Both use noise mechanisms, exponential bounds DP: want hypothesis test power **low**; RS: want confidence **high**

Lipschitz Connection: Both proofs rely on Lipschitz constant L: DP: L controls sensitivity→determines noise RS: L controls p_A change→determines radius DP Noise $\propto \frac{L}{\varepsilon}$; RS Radius $\propto \frac{\sigma}{L}$

7. Privacy

ε-DP: $\mathbb{P}(M(D) \in S) \leq e^\varepsilon \mathbb{P}(M(D') \in S)$ for all neighboring D, D' $e^\varepsilon \approx 1 + \varepsilon$ for small ε **Laplace**: $f(D) + \text{Lap}\left(\frac{\Delta_1}{\varepsilon}\right)$; $\Delta_p = \max_{D\sim D'} \|f(D) - f(D')\|_p$

(ε,δ)-DP: $\mathbb{P}(M(D) \in S) \leq e^\varepsilon \mathbb{P}(M(D') \in S) + \delta$ δ: tail prob bound, **NOT "leak prob"!** Typically $\delta \ll \frac{1}{n}$ **Gaussian**: $\sigma = \frac{\Delta_2\sqrt{2\ln(\frac{1.25}{\delta})}}{\varepsilon}$

Neighbor Definitions: $\|D - D'\|_0 \leq 1$: add/remove one record→Laplace $\|D - D'\|_2 \leq 1$: continuous perturbation (∇ s)→Gaussian

Three Properties: **Post-processing**: $g \circ M$ still DP (can't "purify" noise) **Composition**: $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-DP **Subsampling**: sample rate $q \Rightarrow (q\varepsilon, q\delta)$ **Advanced**: T steps→$\varepsilon_{\text{tot}} = O(\sqrt{T}\varepsilon)$ (crucial for training!) Independent data: $(\max\varepsilon, \max\delta)$

DP-SGD:
1. **Clip** each ∇: $g_{\text{clip}} = g \cdot \min\left(1, \frac{C}{\|g\|_2}\right)$
2. **Aggregate + noise**: $g_{\text{noisy}} = \frac{1}{L}\sum g_{\text{clip}} + \mathcal{N}\left(0, \sigma^2 \frac{C^2}{L^2}\right)$

Clipping bounds sensitivity $\Delta_2 \leq C$ $\sigma = \frac{C\sqrt{2\ln(\frac{1.25}{\delta})}}{L\varepsilon}$ Model private even against white-box attacker

Privacy Amplification: Apply (ε,δ)-DP on random subset $q = \frac{L}{N}$: Result: $(\tilde{q}\varepsilon, q\delta)$-DP where $\tilde{q} \approx q$ T steps: $(\tilde{q}T\varepsilon, qT\delta)$ or $\left(O(q\varepsilon\sqrt{T}), \delta\right)$

PATE: M teachers on disjoint data, noisy voting labels public data, train student $n_{j(x)} = \#\{t : t(x) = j\}$; output $\arg\max\left(n_{j(x)} + \text{Lap}\left(\frac{2}{\varepsilon}\right)\right)$ **Add noise BEFORE argmax!** Sensitivity=2 (NOT $|Y|$!) Each query costs ε; total budget accumulates

FedSGD vs FedAvg: **FedSGD**: send single-step ∇ g_k; server averages **FedAvg**: client runs E epochs, sends weight diff $\Delta\theta$ FedAvg harder to invert (multi-step trajectory unknown)

DP-FedSGD Noise: Centralized: $\sigma_{\text{central}} = \frac{C\sqrt{2\ln(\frac{1.25}{\delta})}}{L\varepsilon}$ Distributed (m clients): $\sigma_{\text{client}} = \sqrt{m} \cdot \sigma_{\text{central}}$ Aggregation: $\frac{1}{m}\sum g_k$ gives same noise level as centralized

8. Privacy Attacks

Attack Hierarchy: **Attribute Inference**: infer sensitive attr **(no membership needed!) Data Extraction**: verbatim memorization (K-extractable) **MIA**: determine if $x \in D_{\text{train}}$ **Dataset Inference**: aggregate weak signals→strong signal **∇ Inversion**: reconstruct from ∇s (FL)

MIA Methods: **Shadow Model**: train K shadows, train attack classifier **LiRA**: $\log\left(\frac{P(\ell|x\in D)}{P(\ell|x\notin D)}\right)$ likelihood ratio **Min-K% Prob**: average of lowest K token probs (LLM) **Loss-based**: training data has lower loss Practical AUC≈0.5-0.7 (weak!); TPR@FPR=0.01 only 2%

∇ Inversion: $x^* = \arg\min_x \|\nabla_\theta \mathcal{L}(x,y) - \nabla_{\text{obs}}\|^2 + R(x)$ Prior $R(x)$: TV (image), Perplexity (text), Entropy (tabular) FedSGD + BS=1: **exact reconstruction** ($\nabla W_1 = \delta x^\top$) FedAvg: needs multi-epoch coupling, harder

Model Stealing/Inversion: **Stealing**: query API, train copy via distillation **Inversion**: $x^* = \arg\max_x P(y_{\text{target}}|x)$ reconstruct class representative Defense: rate limit, output perturbation, watermarking

Memorization Factors: Model size↑, Prefix length↑, Repetition↑: more memorization Sequence length↑: less (cumulative errors)

9. Synthetic Data & Marginals

Pipeline:
1. **Select** marginal queries; 2. **Measure** with DP; 3. **Generate** synthetic

Marginal $\mu_t = \sum_{x\in D}[x_C = t]$; $\Delta_2(M_C) = 1$ (one row→one entry)

Chow-Liu: MI-weighted complete graph→MST→sample along tree $p(F_1,F_2,F_3) = p(F_1)p(F_2|F_1)p(F_3|F_1)$ DP: exponential mechanism for MST, Gaussian for marginals

Marginal Properties: $(n-1)$-way marginals do **NOT** uniquely describe dataset Low-order marginals miss high-order correlations (XOR problem) 3 columns, all 2-way: $\binom{3}{2} = 3$ queries→3ε total

10. Logic & DL2

Logic→Loss Translation: Theorem: $T(\varphi)(x) = 0 \Leftrightarrow x \vDash \varphi$ $t_1 \leq t_2$: $\max(0, t_1 - t_2)$; $t_1 = t_2$: $(t_1 - t_2)^2$ $\varphi \wedge \psi$: $T(\varphi) + T(\psi)$; $\varphi \vee \psi$: $T(\varphi) \cdot T(\psi)$ By construction $T(\varphi) \geq 0$; negation via De Morgan **Quantifiers NOT directly supported**; $\forall$ via max (worst violation)

Training with Background Knowledge: Goal: $\max_\theta \mathbb{E}[\forall z. \varphi(z,s,\theta)]$ Reform: $\min_\theta \mathbb{E}[T(\varphi)(\hat{z},s,\theta)]$ where $\hat{z} = \arg\max T(\neg\varphi)$ This is adversarial attack! Restrict z to ℓ_∞ ball, PGD+project

Logic Properties: If $T(\neg\varphi)(y) = 0$, then $\neg\varphi$ satisfied at y→$\forall x.\varphi(x)$ FALSE $T(\varphi)(y_1) \leq T(\varphi)(y_2) \Rightarrow T(\neg\varphi)(y_1) \geq T(\neg\varphi)(y_2)$ Infinite minimizers possible (e.g., φ is tautology)

11. Fairness

Individual Fairness: (D,d)-Lipschitz: $D(M(x), M(x')) \leq d(x,x')$ Equivalent to robustness: $\forall\delta \in \mathbb{B}_{S(0,\frac{1}{L})} : M(x) = M(x+\delta)$ Lemma: $\Phi^{-1}(\mathbb{E}[h(x+\varepsilon)])$ is 1-Lipschitz

Group Fairness: **Demographic Parity**: $\mathbb{P}(\hat{Y} = 1|S = 0) = \mathbb{P}(\hat{Y} = 1|S = 1)$ **Equal Opportunity**: above conditioned on $Y = 1$ (TPR equal) **Equalized Odds**: conditioned on both $Y = 0$ and $Y = 1$ Eq Odds $\Leftrightarrow \hat{Y} \perp S|Y$ (conditional independence)

Δ_{EO} Calculation: $\Delta_{\text{EO}} = |\text{FPR}_0 - \text{FPR}_1| + |\text{TPR}_0 - \text{TPR}_1|$ Example: $S = 0$: FPR=7/10=0.7, TPR=3/6=0.5 $S =$

1: FPR=2/8=0.25, TPR=16/20=0.8 $\Delta_{\text{EO}} = |0.7 - 0.25| + |0.5 - 0.8| = 0.45 + 0.3 = 0.75$

Adversary Bound: Balanced Accuracy: $\text{BA}(h) = \frac{1}{2}\left(\mathbb{E}_{Z_0}(1-h) + \mathbb{E}_{Z_1} h\right)$ Optimal adversary: $h^*(z) = [p_1(z) \geq p_0(z)]$ Theorem: $\Delta_{\text{EO}(g)} \leq 2 \cdot \text{BA}(h^*) - 1$

Eq Odds Proof Sketch: Goal: $\mathbb{P}(\hat{Y} = 1|S = s, Y = y)$ same for all $s \rightarrow \hat{Y} \perp S|Y$ Use: $\mathbb{P}(\hat{Y}|Y) = \sum_s \mathbb{P}(\hat{Y}|S = s, Y)\mathbb{P}(S = s|Y)$ If $\mathbb{P}(\hat{Y}|S, Y) = c$ for all s: $\mathbb{P}(\hat{Y}|Y) = c$ →conditional indep

LAFTR: $\min_{f,g} \max_h \left[\mathcal{L}_{\text{clf}(f,g)} - \gamma \mathcal{L}_{\text{adv}(f,h)}\right]$ Use adversary to upper bound unfairness

LCIFR: Train encoder: $\forall x' \in S_{d(x)} : \|f(x) - f(x')\|_\infty \leq \delta$ MILP compute ε s.t. $f\left(S_{d(x)}\right) \subset \{z' : \|f(x) - z'\|_\infty \leq \varepsilon\}$ Consumer gets simple robustness problem

12. Watermark & Benchmark

Red-Green Watermark: hash(context)+key→split vocab into Green/Red **Generate**: add δ bias to Green token logits **Detect**: count Green tokens, binomial test **without LLM!** p-value$< \alpha$→watermarked; α controls FPR directly

ITS/SynthID: **ITS**: distortion-free in expectation, but deterministic output **SynthID**: distortion-free + non-deterministic Tournament sampling: high G-value tokens more likely to win

Watermark Attacks: **Scrubbing**: paraphrase 30% tokens removes watermark **Spoofing**: modify one word, watermark persists (piggyback) **Stealing**: 30K queries estimate $\frac{P_{\text{wm}}}{P_{\text{base}}}$, predict Green

Contamination: **Data**: benchmark in training set (memorize answers) **Task**: optimized for task format (not truly solving) Detection: N-gram (L1), Perplexity (L2), Completion (L3) Outcome-based: compare 2024 vs 2025 performance (time causality)

VNN-COMP Critique: "Verified 68M params"→check: #Crossing, accuracy, ε size Small ε=fewer crossing=easier; timeout=3600s impractical Verified $\neq$ practically robust

13. Post-Training Attacks

Quantization Attack: FP32 benign (passes detection), INT8 malicious (activated after deploy) Box constraint $[w_{\text{low}}, w_{\text{high}}]$ s.t. quantized value unchanged Fine-tune in box with clean data→FP32 looks normal

Fine-Tuning Attack: $\mathcal{L} = \mathcal{L}_{\text{clean}(\theta)} + \lambda \mathcal{L}_{\text{attack}(\theta - \nabla \mathcal{L}_{\text{user}})}$ Safe now, malicious after user fine-tunes Needs Hessian: $(\partial \mathcal{L})\frac{\theta'}{\partial}\theta = \frac{\partial \mathcal{L}}{\partial}\theta' \cdot (I - \eta \nabla^2 \mathcal{L}_{\text{user}})$

Agentic AI / IPI: Indirect Prompt Injection: malicious instruction in tool output Agent can't distinguish user instruction vs tool content Defense: instruction hierarchy, dual-LLM, command sense Tradeoff: security $\propto$ 1/capability

14. Regulation

EU AI Act: **Unacceptable**: social credit scoring→prohibited **High Risk**: credit scoring, hiring→strict regulation **Limited Risk**: chatbots→transparency requirements Credit scoring is **High Risk**, NOT prohibited!

GDPR: Removing PII insufficient→linkage attacks still possible Even "anonymized" purchase lists may violate GDPR

Appendix: **Norms**:$\|x\|_p = \left(\sum |x_i|^p\right)^{\frac{1}{p}}$; $\|x\|_\infty = \max|x_i|$ $\mathcal{N} = (2\pi)^{-\frac{d}{2}}|\Sigma|^{-\frac{1}{2}} \exp(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu))$ Lap $= \frac{1}{2b}\exp(-|x-\mu|/b)$; Sigmoid$\sigma(x) = \frac{1}{1+e^{-x}}$ **Soft-max&CE**: $\sigma(z)_i = e^{z_i} / \sum_j e^{z_j}$; CE$(z, y) = -\log \sigma(z)_y = -z_y + \log \sum e^{z_j}$ **Derivatives**: $\partial_x b^\top x = b$; $\partial_x x^\top x = 2x$; $\partial_x x^\top A x = (A + A^\top)x$ $\partial_x \|Ax - b\|_2^2 = 2A^{\top(Ax-b)}$ **不等式**: Cauchy-Schwarz:$\langle x, y\rangle \leq \|x\|_2 \|y\|_2$ Hölder:$\|x \cdot y\|_1 \leq \|x\|_p \|y\|_q, \frac{1}{p} + \frac{1}{q} = 1$ Jensen:g凸$\Rightarrow g(\mathbb{E}[X]) \leq \mathbb{E}[g(X)]$ Chebyshev:$\mathbb{P}(|X - \mathbb{E}[X]| \geq \varepsilon) \leq \frac{\mathbb{V}[X]}{\varepsilon^2}$ Minmax:max min $\leq$ min max(Weak Duality) Hoeffding:$\mathbb{P}\left(|\hat{X} - \mathbb{E}[X]| \geq \varepsilon\right) \leq 2\exp\left(-2n\frac{\varepsilon^2}{(b-a)^2}\right)$

prob: $\mathbb{V}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2$; $\mathbb{V}(aX + bY) = a^2\mathbb{V}(X) + b^2\mathbb{V}(Y) + 2ab\text{Cov}$ Bayes:$P(X|Y) = P(Y|X)\frac{P(X)}{P(Y)}$ $\Phi(z) = \mathbb{P}(\mathcal{N}(0,1) \leq z)$; $\Phi^{-1}(0.5) = 0$; $\Phi^{-1}(0.975) \approx 1.96$ **Matrix**: $\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

MILP 编码: $y = |x|$:$y \geq x, y \geq -x, y \leq x + 2u(1-a), y \leq -x + 2|l|a, a \in \{0,1\}$ $y = \max(x_1, x_2)$:$y \geq x_1, y \geq x_2, y \leq x_1 + a(u_2 - l_1), y \leq x_2 + (1-a)(u_1 - l_2)$

Logic: De Morgan:$\neg(\varphi \wedge \psi) = \neg\varphi \vee \neg\psi$; $\neg(\varphi \vee \psi) = \neg\varphi \wedge \neg\psi$ Implication:$\varphi \Rightarrow \psi \equiv \neg\varphi \vee \psi$ Ball:$\mathbb{B}_\varepsilon^1 \subseteq \mathbb{B}_\varepsilon^2 \subseteq \mathbb{B}_\varepsilon^\infty \subseteq \mathbb{B}_{\varepsilon\sqrt{d}}^2$

⚠ **Traps**: MILP complexity $O(2^k)$, k=**Crossing count**! RS theorem deterministic, estimation probabilistic $\sigma \uparrow$ doesn't always $R \uparrow$ (p_A drops!) GCG uses ∇ to **filter**, not **update** n_0 (guess class 100) vs n (estimate prob 100k) PATE: noise **before** argmax, $\Delta_1 = 2$ Tighter $\neq$ better training (Box trains best) Back-sub: negative coeff→opposite bound MILP incomplete for ℓ_2 (quadratic) PGD $\neq$ CW: different objectives FGSM always on ℓ_∞ boundary FedSGD easier to invert than FedAvg δ is tail mass bound, not leak prob Gaussian DP needs ℓ_2 sensitivity Floating-point: theory Sound $\neq$ hardware Sound Credit scoring is High Risk, NOT Unacceptable MIA AUC≈0.5-0.7 (basically random) Universal suffix transfers across models

⚙**PGD 步骤**: **Step 1**: 算初始 logits z_i和分类 **Step 2**: 算 Loss 对 x的梯度 $\nabla_x \mathcal{L}$: 对 $\mathcal{L} = -z_t^2 + \sum_{i \neq t} z_i^2$, $\frac{\partial \mathcal{L}}{\partial x_j} = \sum_i \left(\frac{\partial \mathcal{L}}{\partial z_i}\right)\left(\frac{\partial z_i}{\partial x_j}\right)$; $\frac{\partial \mathcal{L}}{\partial z_t} = -2z_t$; $\frac{\partial \mathcal{L}}{\partial z_{i \neq t}} = 2z_i$ **Step 3**: update $x^{\text{temp}} = x^k \pm \eta \cdot \text{sign}(\nabla)$ (targeted 用 $-$, untargeted 用 $+$) **Step 4**: proj 回$\mathbb{B}_{\varepsilon(x^0)}$ ℓ_∞: $x_i^{\text{new}} = \text{clip}\left(x_i^{\text{temp}}, x_i^0 - \varepsilon, x_i^0 + \varepsilon\right)$; ℓ_2: if $\|x^{\text{temp}} - x^0\| > \varepsilon$: $x^{\text{new}} = x^0 + \frac{\varepsilon(x^{\text{temp}} - x^0)}{\|x^{\text{temp}} - x^0\|}$ **Step 5**: 检查是否攻击成功 (arg max z改变?) **Step 6**: 下一轮 $\eta^{k+1} = \eta^k/2$ (if decay)

⚙ **MILP 验证步骤**: **检验编码正确性**：画约束区域图！1. 令 $a = 0$: 约束简化成什么？解区域是什么？; 2. 令$a = 1$: 约束简化成什么？解区域是什么？; 3. 合并两个区域，应该恰好等于函数图像 **修复 non-uniqueness** (如 HatDisc 在$x = 0$两个值)：添加约束$x \geq \varepsilon(1-a)$强制$x = 0$时$a = 1$

⚙ **Binary Step 编码**: $\sigma(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$; $x \in [l, u]$, $l < 0 < u$ **Case $l \geq 0$**: $y = 1$ (constant) **Case $u < 0$**: $y = 0$ (constant) **Case $l < 0 < u$**: 需要$a \in \{0,1\}$ $y \geq a, y \leq a, x \geq l \cdot a, x \leq u \cdot a + l(1-a)$... (类似 ReLU 但输出$\{0,1\}$不是$[0,u]$)

⚙ **DeepPoly 计算完整流程**: **Forward pass** (算 concrete bounds): 1. Input: $x_1 \in [l_1, u_1]$, $x_2 \in [l_2, u_2]$; 2. Affine: $x_3 = x_1 + x_2 - 0.5 \rightarrow x_3 \in [l_1 + l_2 - 0.5, u_1 + u_2 - 0.5]$; 3. 判断 ReLU 类型: $l_3 < 0 < u_3$? → crossing!; 4. ReLU symbolic: upper $x_5 \leq \lambda(x_3 - l_3)$, lower $x_5 \geq \alpha x_3$
Back-substitution (精化 bounds):
1. 从 output 开始: $x_7 = -x_5 + x_6 + 3$; 2. 要算u_7→max x_7 →min x_5, max x_6; 3. 替换x_5, x_6的 symbolic bounds; 4. 继续替换直到只剩 input 变量; 5. 在 input domain 上优化(取端点!)

符号规则: 算 upper bound 时: 正系数$c_i > 0$: 用x_i的 upper bound; 负系数$c_i < 0$: 用x_i的 **lower** bound!

⚙ **DeepPoly 数值例子**: $x_1, x_2 \in [0, 2]$; $x_3 = x_1 + x_2 - 0.5 \in [-0.5, 3.5]$ (crossing) $x_5 = \text{ReLU}(x_3)$: $\lambda = \frac{3.5}{4} = 0.875$ Upper: $x_5 \leq 0.875(x_3 + 0.5) = 0.875x_3 + 0.4375$ Lower: $x_5 \geq 0$ (选$\alpha = 0$因为$|l| = 0.5 < u = 3.5$? 不对, $|l| < u$时选$\alpha = 1$) 实际: $|\{-0.5\}| = 0.5 < 3.5$→min area 用$\alpha = 1$: $x_5 \geq x_3$ Back-sub x_5到 input: Upper: $x_5 \leq 0.875(x_1 + x_2 - 0.5) + 0.4375 = 0.875x_1 + 0.875x_2$ Max at $x_1 = x_2 = 2$: $u_5 = 3.5$

⚙ **Certified Training 计算题**: **题型**: 给网络结构和 weight w，用 Box 传播，算 worst-case loss，做一步 GD **Step 1**: Box 传播 (bounds 是w的函数!) $x_3 = wx_1 + b \rightarrow x_3 \in [wl_1 + b, wu_1 + b]$ if $w \geq 0$ (注意$w < 0$时上下界交换!) **Step 2**: ReLU 后 bounds $x_5 = \text{ReLU}(x_3)$: $l_5 = \max(0, l_3)$, $u_5 = \max(0, u_3)$ **Step 3**: Worst-case loss (CE with logits x_7, x_8, target=x_8) $\mathcal{L}_{\text{worst}} = \log(1 + \exp(u_7 - l_8))$ (max x_7, min x_8)

Step 4: 梯度 (chain rule through bounds) $\partial\frac{\mathcal{L}}{\partial}w = \partial\frac{\mathcal{L}}{\partial}u_7 \cdot \partial\frac{u_7}{\partial}w + \ldots$ **Step 5**: 更新 $w_{\text{new}} = w - \eta\partial\frac{\mathcal{L}}{\partial}w$ **连续性**: Bounds 是w的连续函数 (linear+max 都连续)

⚙ **RS 认证计算**: **Given**: σ, 采样结果n次中n_A次是 class A **Step 1**: 估计$\hat{p}_A = \frac{n_A}{n}$ **Step 2**: 计算置信下界$\underline{p_A}$ (Clopper-Pearson 或正态近似) 正态近似: $\underline{p_A} = \hat{p}_A - z_{\frac{\alpha}{2}}\sqrt{\frac{\hat{p}_{A(1-\hat{p}_A)}}{n}}$ **Step 3**: if $\underline{p_A} \leq 0.5$: ABSTAIN **Step 4**: $R = \sigma \cdot \Phi^{-1}\left(\underline{p_A}\right)$ **常用值**: $\Phi^{-1}(0.5) = 0, \Phi^{-1}(0.84) \approx 1, \Phi^{-1}(0.975) \approx 1.96$

⚙ **为什么σ大不一定R大**: $R = \sigma \cdot \Phi^{-1}(p_A)$ $\sigma\uparrow$→ 直接效应: $R\uparrow$ $\sigma\uparrow$→ 噪声大 → $p_A\downarrow$ → $\Phi^{-1}(p_A)\downarrow$ → $R\downarrow$ 两个效应相反! 存在最优σ^*

⚙ **DP 敏感度计算**: $\boldsymbol{\Delta_1}$ (L1): 改变一条记录，输出向量 L1 变化 max $\boldsymbol{\Delta_2}$ (L2): 改变一条记录，输出向量 L2 变化 max **Mean**: $f(D) = \frac{1}{n}\sum x_i$; 加/删一个$x$: $\Delta_1 = \|x\frac{\|_1}{n}$ 若x有界 $\|x\|_1 \leq B$: $\Delta_1 = \frac{B}{n}$ **∇** (一个 sample): $\Delta_2 \leq C$ (after clipping!) **PATE 投票**: 改变一个教师→一票变化→n_j改变$(+1, -1)$ →$\Delta_1 = 2$

⚙ **DP 预算计算**: **Simple Composition**: k个(ε, δ)-DP query → $(k\varepsilon, k\delta)$ **3 columns, 2-way marginals**: $\binom{3}{2} = 3$ queries → 总预算3ε **Subsampling**: 采样率$q = \frac{L}{N}$ (ε, δ)-DP mechanism → $(\approx q\varepsilon, q\delta)$-DP **Advanced** ($T$ steps): $\left(O\left(\sqrt{T}\varepsilon\right), \delta\right)$ 而非 $(T\varepsilon, T\delta)$

⚙ **∇ Inversion 可行性**: **FedSGD + BS=1**: $\nabla_{W_1}\mathcal{L} = \delta \cdot x^\top$ → 可精确恢复x! (解线性系统) **FedSGD + BS>1**: 只能恢复$\sum x_i$的线性组合 **FedAVG**: 多步更新，需要模拟整个轨迹，更难 **Binary classification** ($d = 2$): ∇符号直接揭示 label! **Multi-class** ($d > 3$): ∇是向量，无法唯一确定 label

⚙ **Δ_{EO}计算步骤**: **Given**: 数据表(Dataset)和预测表(Predictions) **Step 1**: 算各组 FPR (False Positive Rate) $\text{FPR}_s = P\left(\hat{Y} = 1|Y = 0, S = s\right) = \frac{\#\text{预测 1 且真实 0}}{\#\text{真实 0}}$ **Step 2**: 算各组 TPR (True Positive Rate) $\text{TPR}_s = P\left(\hat{Y} = 1|Y = 1, S = s\right) = \frac{\#\text{预测 1 且真实 1}}{\#\text{真实 1}}$ **Step 3**: $\Delta_{\text{EO}} = |\text{FPR}_0 - \text{FPR}_1| + |\text{TPR}_0 - \text{TPR}_1|$ **Example**: $S = 0, Y = 0$: 10 人中 7 人预测 1 → FPR$_0$=0.7 $S = 0, Y = 1$: 6 人中 3 人预测 1 → TPR$_0$=0.5 $S = 1, Y = 0$: 8 人中 2 人预测 1 → FPR$_1$=0.25 $S = 1, Y = 1$: 20 人中 16 人预测 1 → TPR$_1$=0.8 $\Delta_{\text{EO}} = |0.7 - 0.25| + |0.5 - 0.8| = 0.75$

⚙ **BA 与 Δ 关系**: **Adversary** $h(z, y)$尝试从z预测S **定义**: $h(z, 0) = 1 - g(z), h(z, 1) = g(z)$ **BA 计算**: BA $= \frac{1}{2}$[accuracy on $S = 0$ + accuracy on $S = 1$] For $Y = 0$: h预测$S = 0$的 prob=$P(g = 0|S = 0, Y = 0)$; 预测$S = 1$的 prob=$P(g = 1|S = 1, Y = 0)$ **Theorem**: $\Delta_{\text{EO}(g)} \leq 2\text{BA}(h^*) - 1$ 验证: 若 $\Delta_{\text{EO}} = 0.75$, BA=?(算出 BA 后代入验证)