Reliable and Trustworthy AI

Lecturer(s): Dr. Martin Vechev

Author: Saurav Banka

Semester: HS 2025

Last edited: September 18, 2025

Contents

1 Lecture 1: Introduction (17.09.25)

1.1 Motivation

Traditional ML progress focused on **standard accuracy** (e.g., ImageNet). Deployments in real-world settings often reveal failures:

- Distribution shifts $\rightarrow$ performance drop.
- Safety-critical failures (autonomous driving, incorrect medical diagnoses).

When deploying models into the real world, we need to know that models are safe, secure, robust, transparent and reliable. The goal of this course is to get a fundamental understanding of privacy and robustness techniques, and a glimpse into the latest research into reliable and trustworthy AI.

1.2 Vertical I: Robustness

Robustness research broadly tackles questions:

- What attacks (e.g. gradient-based, branch-and-bound) and defenses exist?
- Is it possible to certify / prove robustness and performance under perturbations or adversaries?
- How to train models that are provably robust?

1.2.1 Why is certifying robustness difficult?

Consider the use-case of image classification (e.g. MRI images) under input perturbations (noise, blur, rotation) in a range that does not affect human ground-truth labels. The decision boundary of a neural network is a $(d-1)$-dimensional hypersurface in the input space $\mathbb{R}^d$. The goal is that the model prediction should remain invariant to such perturbations.

Certifying robustness involves:

1. **Precondition $\varphi(x)$:** convex polytope of all possible perturbations of x. *Problem:* prohibitively large to enumerate, especially in higher dimensions or under multiple transformations.
2. **Propagation:** push this region through the network layers. *Problem:* results in many small non-convex shapes. Exact methods (MILP, SMT) are NP-complete and do not scale.
3. **Abstraction:** use convex relaxations to approximate the shapes with an enclosing convex polytope. If the final convex region (post-condition) lies entirely in the correct class, the model is robust. *Problem:* loose relaxations introduce false positives and reduce provability.

Summary: Tight approximations are precise but expensive; loose approximations are efficient but may miss guarantees.

1.2.2 Training Certified Models

Even if robustness could be proven efficiently, unless a network is trained to be provable, it is unlikely to satisfy such specifications.

Key idea: Instead of propagating individual datapoints, propagate convex regions (symbolic inputs). Backpropagation then uses symbolic information to reduce the size of post-conditions, encouraging perturbations to cluster closely in representation space.

Objective. Standard training minimizes

$$\min_{\theta} \mathbb{E}\big[\,\mathrm{loss}(\theta, x, y)\big].$$

With a robustness specification, this becomes a min–max optimization:

$$\min_{\theta} \mathbb{E}\Big[\max_{x' \in \varphi(x)} \underset{(\varphi,\psi)}{\mathrm{loss}}(\theta, x', y)\Big].$$

Interpretation: minimize the worst-case loss over perturbations. *Challenge:* This problem is much harder to optimize, may fail to enforce specifications, and often degrades standard accuracy.

1.2.3 Individual Fairness and Randomized Smoothing

Individual fairness: If two datapoints are similar in relevant aspects for a task, they should receive similar predictions. This is closely related to robustness: perturbations in sensitive attributes should not flip outputs.

Randomized smoothing: An inference-time defense that replaces the classifier with a smoothed version. Given an input x, add Gaussian noise and average predictions. Guarantee: nearby inputs map to the same label with high probability. Compared to certified training, randomized smoothing scales better but only provides guarantees for certain robustness properties.

1.3 Vertical II: Privacy

Privacy in this course is primarily focused on **data privacy**: preventing leakage of sensitive information from models. Key research questions:

- **Membership inference:** Given a datapoint x, can an attacker determine if x was in the training set?
- **Model inversion:** Can an attacker reconstruct a representative example from a given class?
- **Training data extraction:** Can raw training samples be recovered from a model (e.g. LLMs regurgitating text)?
- **Private attribute inference:** Can sensitive attributes (age, gender, location) be inferred from user data?
- **Model stealing:** Recovering model weights or functionality from black-box access.

Defenses:

- **Differential Privacy (DP):** Add noise during training (e.g. DP-SGD). Guarantees: presence/absence of one sample does not significantly affect the output distribution. Trade-off: higher privacy $\leftrightarrow$ lower accuracy. Recent work shows DP-LLMs approaching performance of earlier non-private models.
- **Federated Learning:** Train across distributed devices (e.g. smartphones) without centralizing data. Related to fine-tuning multiple LLMs and merging them securely.

Beyond standalone models, in compound systems such as **agentic AI**, LLMs are integrated with external tools. Safety risks emerge from multi-step workflows (e.g. insecure protocols, toxic flows). Ensuring provable safety here remains an open challenge.

1.4 Vertical III: Provenance and Evaluation

Two central themes in this vertical are **data attribution** (who generated what) and **evaluation** (how to measure performance fairly).

1.4.1 Watermarking and Data Attribution

- **Idea:** Randomly assign tokens a color (expect evenly distributed) and transform the output distribution to skew to a color.
- **Threats:**
  - *Stealing:* Approximate the watermark distribution with repeated queries.
  - *Spoofing:* Generate text that falsely appears to come from a target model.
  - *Scrubbing:* Remove watermark to conceal that text was model-generated.

1.4.2 Transformations and Safety

Common LLM transformations (quantization, pruning, distillation, fine-tuning) may impact safety and privacy guarantees.

Example: A model appears normal pre-quantization in benchmarks, etc but after quantization begins inserting hidden adversarial content (e.g. advertisements).

1.4.3 Evaluation and Benchmarks

- **Benchmarks:**
  - Closed-form (e.g. math problems with unique solutions, math arena).
  - Preference-based (e.g. LLM Arena where users vote).
- **Contamination:** Training/test set overlap or task leakage. Leads to inflated benchmark results.
- **Real-world failures:** Some model releases (e.g. K2) were later shown to have flawed evaluations due to contamination or misleading comparisons.