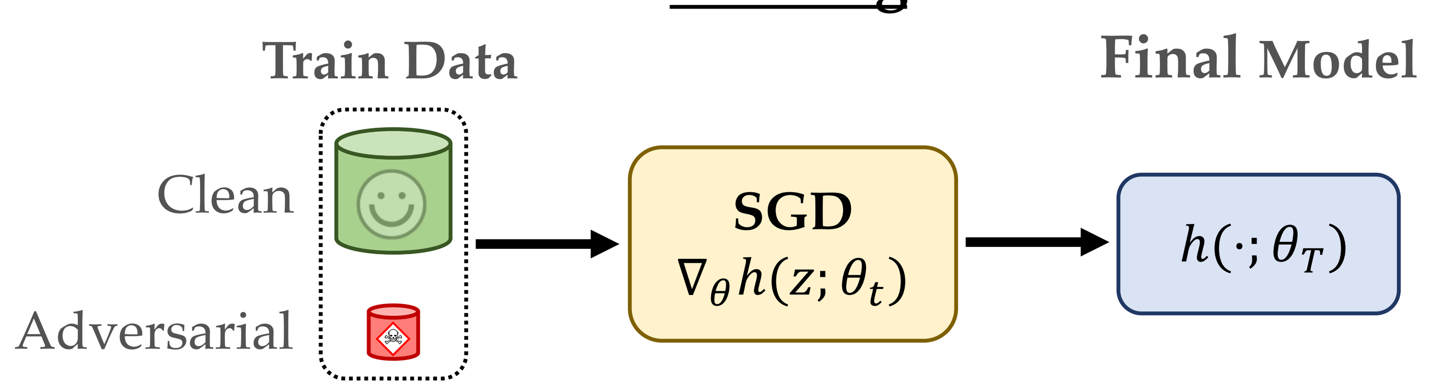
Simple, Attack-Agnostic Defense Against Targeted Training Set Attacks Using Cosine Similarity

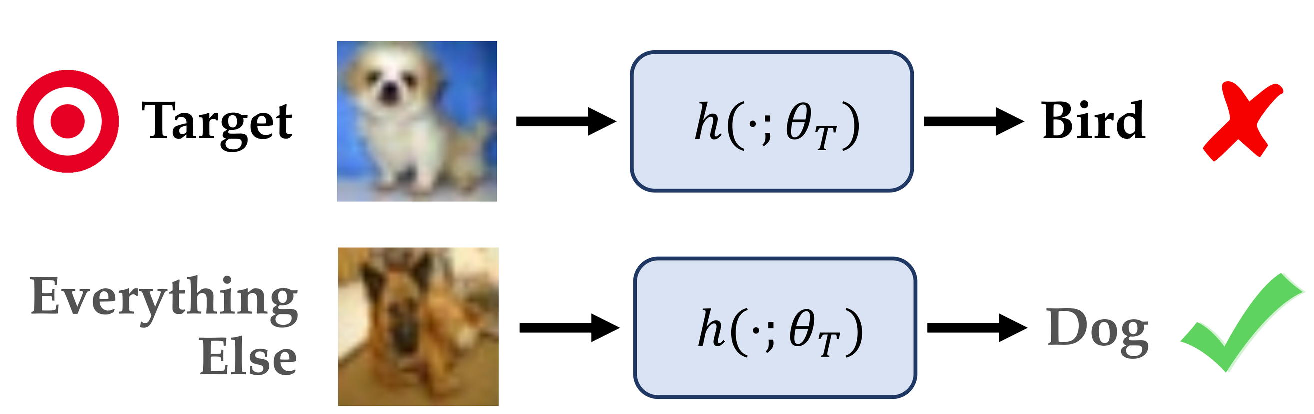
Zayd Hammoudeh, Daniel Lowd

{zayd, lowd}@cs.uoregon.edu

UNIVERSITY OF OREGON

What is a Training Set Attack?





General-Purpose Influence Estimation

Influence: Quantifies how much each training example contributes to a test instance's classification loss

Existing Influence Estimation Methods:

- Influence functions [1]
- Representer points [2]
- Training gradient aggregation methods, e.g., TracIn [3]

TracIn Checkpoint Influence Estimator [3]

$$\text{TracInCP}(z, z_{\text{te}}) := \sum_{t=1}^{T} \eta_t \underbrace{\nabla_\theta \ell(z; \theta_t) \cdot \nabla_\theta \ell(z_{\text{te}}; \theta_t)}$$

Training Ex. Test Ex. Training / Test Gradient Dot Product Over Each Epoch

Takeaway: Influence estimation simplifies to sums of dot products over the training set & a test (target) example

ICML International Conference On Machine Learning 2021 Workshop on Uncertainty in Deep Learning

Our Goals

Identification: Separate the **clean** & **adversarial** datasets

Target Detection: Determine if a test example is targeted $z_{\text{te}} \stackrel{?}{=}$

CosIn — Our Method

Insight: must be *highly influential* to change 's prediction

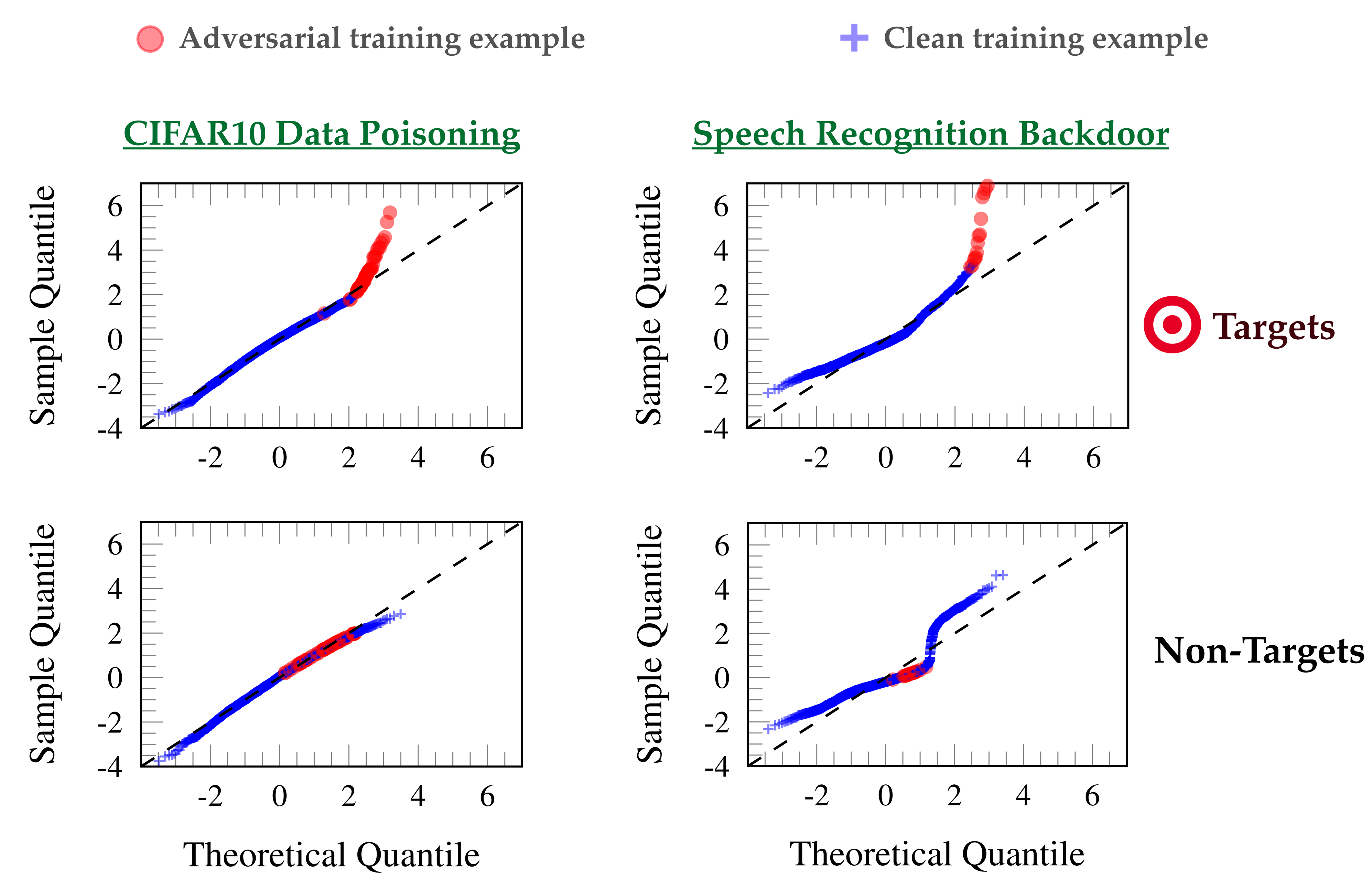
Observation: Existing influential estimation methods identify poison very poorly

Our Method: Cosine Similarity Influence Estimator — CosIn — adapts TracIn to better identify *highly influential* examples that are *likely to be attacks* by:

1. Normalize TracInCP dot products by gradient norms
2. Consider all examples at any checkpoint — not just those in the minibatches

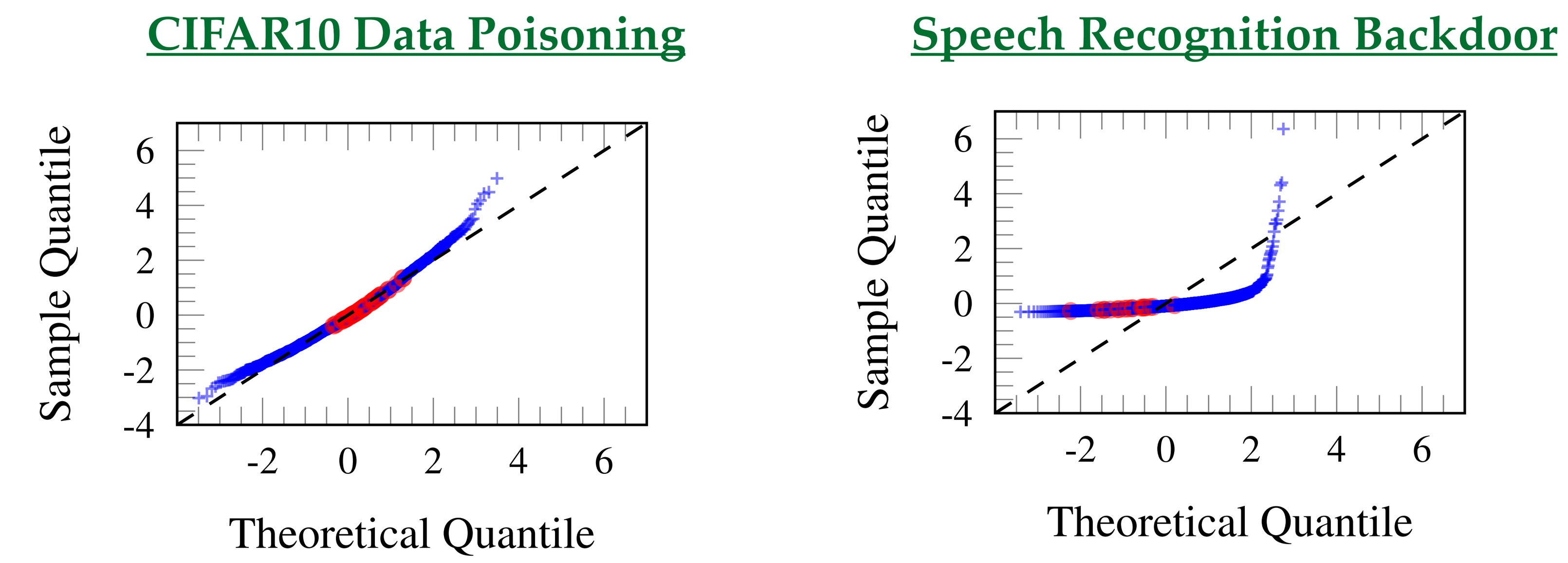
Using CosIn to Detect a Target

Key Insight: 's CosIn influence distribution should have an *exceptionally heavy upper-tail* due to the *exceptionally influential*



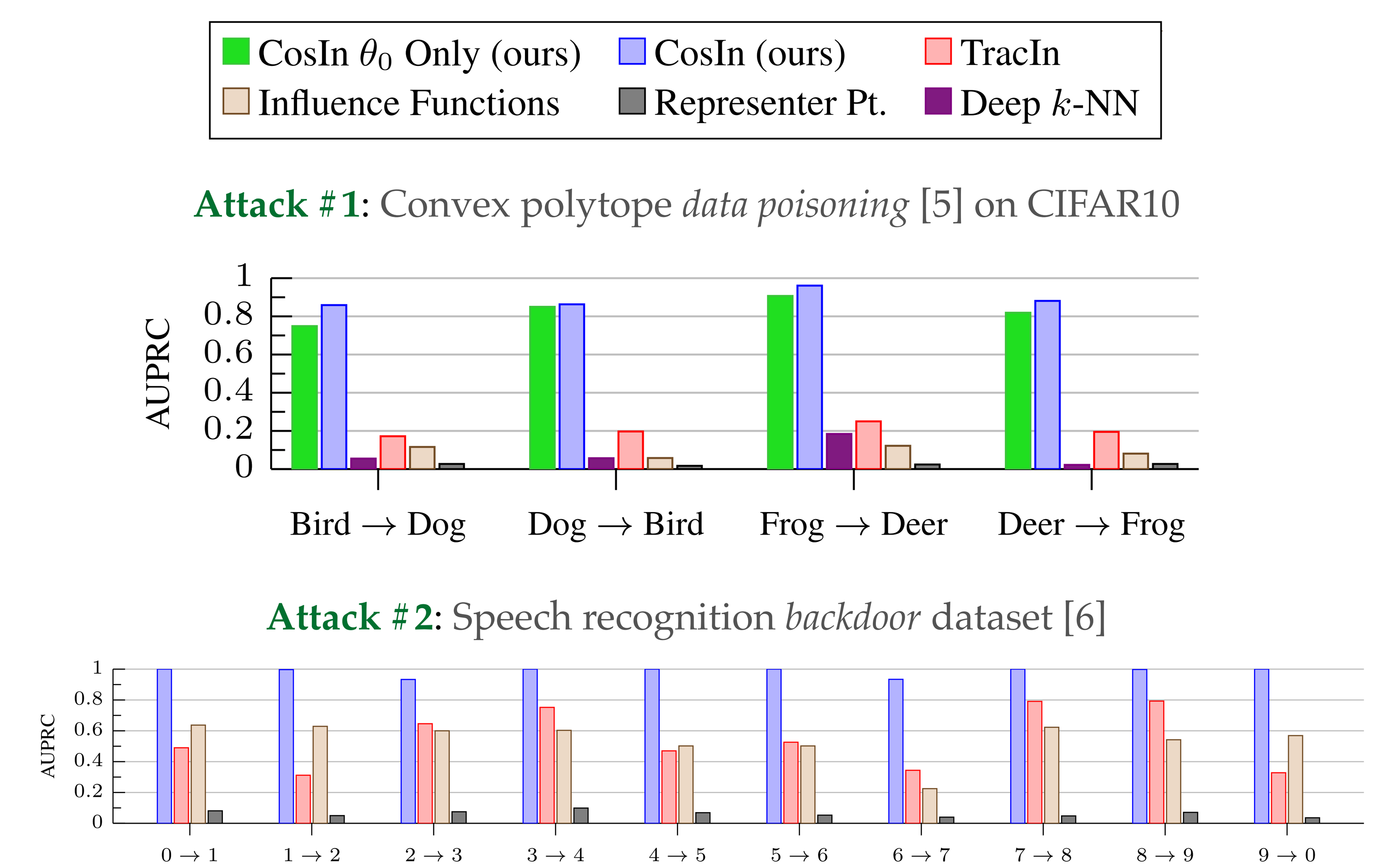
Why Normalize the Dot Products?

Observation: 's gradient magnitudes are not well correlated with whether the training instance is adversarial



Experimental Results

Baselines: Influence estimation methods & Deep KNN [4] poison defense



And a lot more! Text and target detection experiments are in the paper…

References

[1] Koh et al., "Understanding black-box predictions via influence functions" ICML, 2017.
[2] Yeh et al. "Representer point selection for explaining deep neural networks", NeurIPS, 2018.
[3] Pruthi et al. "Estimating training data influence by tracing gradient descent" NeurIPS, 2020.
[4] Peri et. al. "Deep k-NN defense against clean-label data poisoning attacks." AROW, 2020.
[5] Zhu et al. "Transferable clean-label poisoning attacks on deep neural nets." ICML, 2019.
[6] Liu et al. "Trojaning attack on neural networks." NDSS, 2018.

github.com/ZaydH/cosin