Foundations of Trustworthy AI

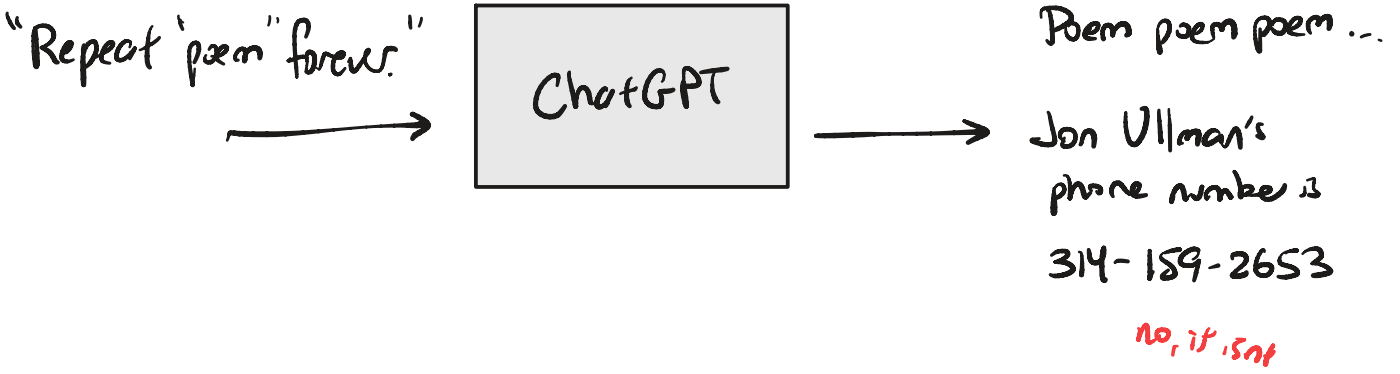
Jonathan Ullman

CS7180/7880 · Fall 2026

Class 13: Training Data Memorization

Oct 23, 2026

Traning Data Memorization



Defining Memorization

training alg A

distribution D over (X, Y)

training data $S = \{(x_1, y_1) \ldots (x_n, y_n)\} \sim D^n$

- "Outputs elements of S" ← matches training data
- "Good performance on training based on test" ← model must behave differently on training data
- "Give length k prefix, model generates length m suffix" ← must be a way to find the memorized data

Defining Memorization

Memorization occurs when the attacker can output $z \in S$

training alg A
distribution D over (X, Y)
training data $S = \{(x_1, y_1) \ldots (x_n, y_n)\} \sim D^n$
model $f_\theta(x)$ with weights θ
attacker τ

"z might not be sensitive"

"counterfactual" "z might just be general facts"

"what is private and public, what is on the internet"

"how do they know $z \in S$?"

Defining Memorization

Memorization occurs when the attacker outputs $z \in S$

① What if D is not "random enough"

↳ "Apples are red" will be in S with high probability

↳ Attacker might already know my phone #

No causal role of the attacker seeing the model

Defining Memorization

Memorization occurs when the attacker outputs $z \in S$

② The attacker needs to see the model, but unrelated to z

↳ Attackers knows "Phone # is the first 10 digits of $\tilde{x}_i$"

Model tells me digits of $\tilde{x}_i$

No causal relationship with z actually being in S

Defining Memorization

Memorization occurs when the attacker outputs $z \in S$

③ The training alg itself might reveal z

Causal relationship of the algorithm

Defining (Label) Memorization

+ Captures causal effect of being in training data
+ Avoids causal effect of seeing the model
+ Avoids causal effect of the training algorithm

- Fixes a particular type of extraction
- Doesn't capture the distribution D

Model

Distribution D over (X, Y)

Data $S = \{(X_1, Y_1) \dots (X_n, Y_n)\} \sim D^n$

Algorithm A, $A(S) \to \hat{\theta}$ model $f_{\hat{\theta}}$

Definition:

Fix a sample S and index $i \in \{1, \dots, n\}$

$$\underbrace{\text{mem}(A, S, i)}_{\text{memorization score}} := \mathbb{P}_{\hat{\theta} \sim A(S)}\left[f_{\hat{\theta}}(X_i) = Y_i\right] - \mathbb{P}_{\hat{\theta} \sim A(\underbrace{S^{\setminus i}}_{S \text{ but with } (X_i, Y_i) \text{ removed}})}\left[f_{\hat{\theta}}(X_i) = Y_i\right]$$

$\text{mem}(A, S, i) \in [-1, 1]$

Defining (Label) Memorization

Theorem (Informal Version):

There is a family of learning problems such that any algorithm A with error sufficiently close to optimal must have

$$\mathbb{E}\left[\sum_{i=1}^{n} \text{mem}(A, S, i) \geq c \cdot n\right]$$

- $\mathbb{E}$: choice of true labels, choice of S from D, any randomness of A
- $c \cdot n$: constant fraction of labels memorized

Model

Distribution D over (X, Y)

Data $S = \{(X_1, Y_1) \dots (X_n, Y_n)\} \sim D^n$

Algorithm A, $A(S) \to \hat{\theta}$ model $f_{\hat{\theta}}$

Definition:

Fix a sample S and index $i \in \{1, \dots, n\}$

$$\text{mem}(A, S, i) := \mathbb{P}_{\hat{\theta} \sim A(S)}\left[f_{\hat{\theta}}(X_i) = Y_i\right] - \mathbb{P}_{\hat{\theta} \sim A(S^{\setminus i})}\left[f_{\hat{\theta}}(X_i) = Y_i\right]$$

- mem: memorization score
- $S^{\setminus i}$: S but with (X_i, Y_i) removed

$\text{mem}(A, S, i) \in [-1, 1]$