The Secret Sharer

Evaluating and Testing Unintended Memorization in Neural Networks

Nicholas Carlini, Google Brain; Chang Liu, University of California, Berkeley; Úlfar Erlingsson, Google Brain; Jernej Kos, National University of Singapore; Dawn Song, University of California, Berkeley

August 14–16, 2019 • Santa Clara, CA, USA

Motivation: Accidental Secrets

- Generative models power modern auto-complete (e.g. The one you have on your smartphone’s keyboard).
- Trained on massive datasets (millions of users).
- Their Goal: Learn general language patterns.
- The Risk: Perfectly memorizing an individual's private data.
- What happens if the model auto-completes your credit card number to a stranger?

Limitations of Past Approaches

- Previous approaches relied on monitoring global overtraining.
- Their Methods: Stop training when overall validation accuracy drops.
- The Limitation: Global metrics completely hide rare anomalies.
- Neural networks memorize rare, sensitive outliers long before global overtraining is ever detected (we will get to that later).

Background: The Canary

- Researchers test memorization by injecting fake secrets ('Canaries').

"My social security number is ___"

- Randomness (r): "078-05-1120"
- Canary (s[r]): "My social security number is 078-05-1120"
- This assembled sentence is injected into the training data. We then examine if the black-box model exposes it.

Background: Log-Perplexity

$$\begin{aligned} \mathrm{Px}_\theta(x_1 \ldots x_n) &= -\log_2 \mathbf{Pr}(x_1 \ldots x_n | f_\theta) \\ &= \sum_{i=1}^{n} \left(-\log_2 \mathbf{Pr}(x_i | f_\theta(x_1 \ldots x_{i-1})) \right) \end{aligned}$$

- Measures how surprised a model is by a sequence.
- Uses the chain rule of probability $\mathbb{P}(A_1 \cap A_2 \cap \ldots \cap A_n)$.
- Sums the negative log-likelihood of every character transition.
- Lower perplexity = Higher probability of generation.

Background: Guessing Entropy

- How much work does the model save an attacker?
- Blind Guessing: Attacker must try half the randomness space.
- Model-Assisted: Attacker queries the model's sorted perplexity Leaderboard.
- Guesses required = The 'Rank' of the true canary.

The Exposure Metric

- Exposure measures the exact reduction in guessing entropy.
- Exposure = $\log_2 |\mathcal{R}| - \log_2 \mathbf{rank}_\theta(s[r])$
- Score of 0: Model provides no advantage.
- Maximum Score: Model perfectly memorized the sequence.

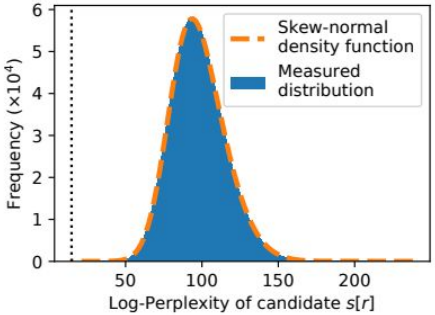
The Computation Problem

The Problem:

- Calculating exact rank for a 16-digit card doesn't take that long, only about 317 GPU-years.

Geometric Solution:

- Sample perplexities perfectly map to a Skew-Normal Distribution.
- Calculate the integral (area under the curve).
- Bypasses discrete counting to estimate exposure instantly.



Architecture: Testing Pipeline

1. Define Canary & Space: Create the template and boundaries.

2. Augment Dataset: Insert the canary 'c' times.

3. Train Model: Use production hyperparameters and strict early stopping.

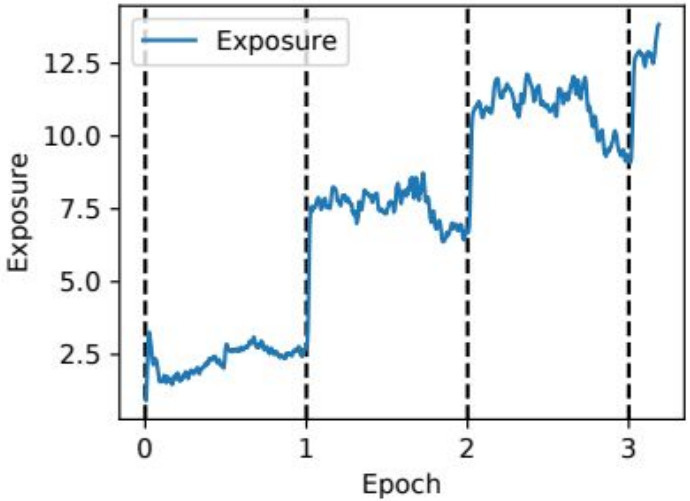
4. Compute Exposure: Run the geometric approximation.

Evaluation: Experimental Setup

- Models: Generative sequence models (LSTMs) predicting characters/words.
- Loss Function: Cross-Entropy Loss.

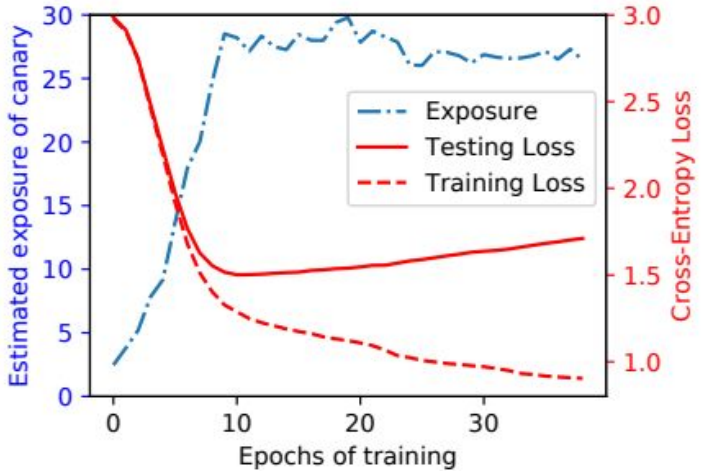
- RQ1: When does memorization occur?
- RQ2: Is memorization just overtraining?
- RQ3: Can attackers extract data efficiently?
- RQ4: Does this affect production models?
- RQ5: Can standard defenses prevent this?

RQ1: When does memorization occur?



- Memorization happens almost immediately.
- Exposure spikes the exact moment the mini-batch containing the canary is processed by the network.

RQ2: Is memorization just overtraining?



- False assumption: Memorization requires overtraining.
- Experiment: Train a model on a 5% data slice to force overfitting and track exposure.

RQ2: The Generalization Paradox

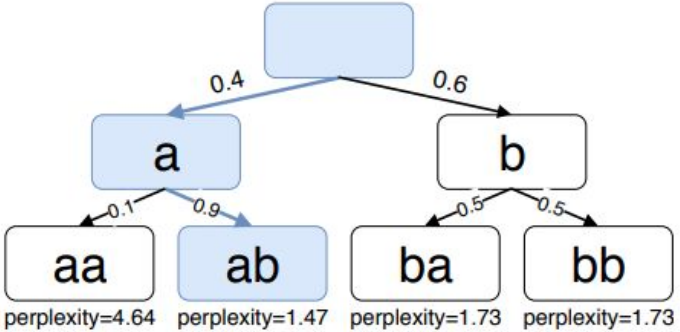
Results from the figure in the previous slide:

- The absolute peak of memorization occurs exactly at the point of lowest validation loss.
- Occurs during perfect generalization, not overtraining.
- Neural networks act as greedy memorizers first to rapidly drop global loss.

RQ3: Can data be extracted efficiently?

- Exposure proves data is mathematically leaked.
- Can a hacker actually pull a 16-digit secret out of a black-box model without brute-forcing it?
- Solution: Shortest-Path Graph Search.

RQ3: Shortest-Path Extraction



1. Map sequence generation as a massive, branching tree.

2. Edge weight = negative log-likelihood of the next character.

3. Lightest path = lowest log-perplexity.

4. Use modified Dijkstra's Algorithm to navigate and prune dead ends.

RQ3: GPU Batching Optimization

Bottleneck: Querying one node is too slow

Optimization: Process thousands of nodes simultaneously on GPUs.

Result:

- 100,000x reduction in required guesses.
- 317 years of computing becomes a 5-minute attack.

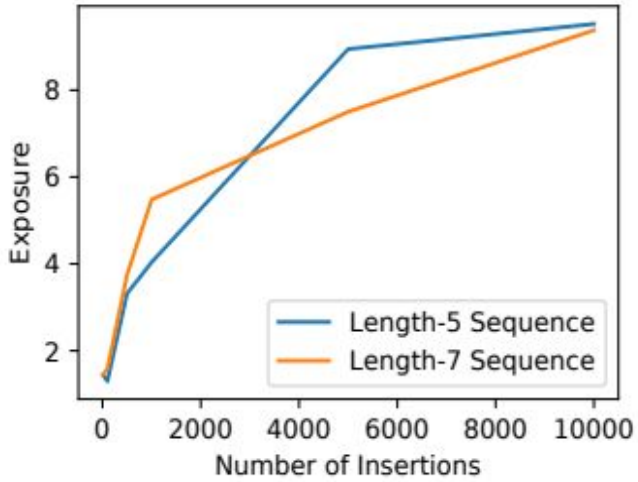
RQ3: Real-World Extraction

- Target: 2-Layer LSTM trained on the Enron Corporate Email Dataset.
- Did not insert canaries, hunted for pre-existing secrets.
- Successfully extracted actual employee Credit Cards and SSNs in hours.

RQ4: Does this happen in production?

- Target: Google's Smart Compose.
- Massive industrial LSTM.
- Trained on billions of real user emails.
- Does the sheer size of the dataset protect a single, repeated secret?

RQ4: The Log-Linear Tipping Point



- Below threshold: Cross-entropy penalty is diluted. No memorization.
- Above threshold: Exposure skyrockets in a log-linear path.

RQ4: Neural Machine Translation

Target: Encoder-Decoder NMT architectures (Thought Vectors).

Experiment: Inserted an English SSN paired with a Vietnamese translation.

Results:

- 1 insertion = 1,000x exposure spike.
- 4 insertions = sequence is completely memorized.

Translation engines memorize anomalies even faster than text predictors.

RQ5: Can standard defenses prevent this?

- Researchers tested standard tools meant to stop overtraining.
- Weight Decay (L2): Lowered global loss, but exposure remained high.
- Dropout (up to 90%): Yielded no statistically significant reduction in memorization.

RQ5: The Quantization Proof

Experiment: Crushed 32-bit float parameters down to 8-bits.

- Model capacity (2.4MB) dropped below the dataset size (1.7MB).
- Result: Exposure remained identical.

=> Memorization is a pathway problem, not a storage problem.
It won’t be fixed with standard regularization.

The Solution: DP-SGD

Differentially Private Stochastic Gradient Descent (DP-SGD).

- Gradient Clipping: Physically decapitates the penalty spike.
- Gaussian Noise Injection: Creates a mathematical smokescreen over the gradients.

The Bitter Trade-off

DP-SGD perfectly drops Exposure down to zero.

Cost:

- Injecting mathematical noise damages the general learning process.
- Engineers can have perfect privacy, or perfect accuracy. Currently, they cannot have both.

Attack Limitations: The Arg Max Wall

- The extraction attack requires viewing full output probability distributions.
- Most real-world APIs only return the 'Arg Max' (the single top word).
- Without probability fractions, Dijkstra graph routing fails.

Attack Limitations: Context

- The attack requires knowing the exact template around the secret.
- If a user typed 'My SSN is', but the hacker guesses 'The social security number is', the entire network path breaks.

Future Work

- Current metrics only look at input-output behavior (black box).
- Engineers have full 'white-box' access to their own models.
- Future Work: Develop scanners that analyze raw internal weights to find memorization directly.

Conclusion

Shortest-Path Search proves these secrets can be efficiently extracted.

DP-SGD is the only proven defense, but it degrades model utility.

Thank you! Questions?