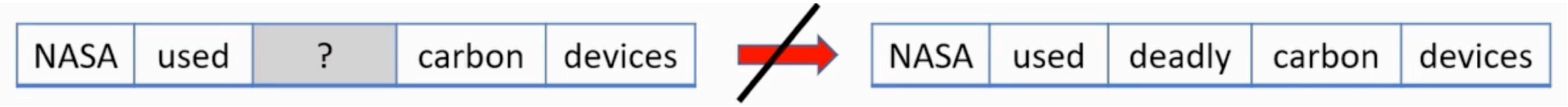
Analyzing Information Leakage of Updates to Natural Language Models

Santiago Zanella-Beguelin, Lukas Wutschitz, Shruti Tople, Victor Ruhle, Andrew Paverd, Olga Ohrimenko, Boris Kopf, and Mark Brockschmidt

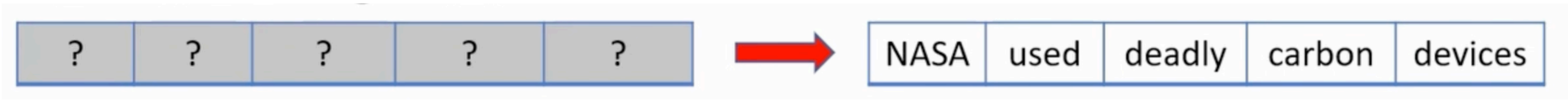
Levi Kaplan 11/20/23

Problem Statement

- NLMs can reveal sensitive information that’s been memorized
- Prior work: single snapshot can’t reveal sensitive info
  - Even when given context of first and last two words



- This work: given two snapshots, can reveal info without context, appearing 2x less frequently



Visualizations taken from authors’ talk: https://dl.acm.org/doi/10.1145/3372297.3417880

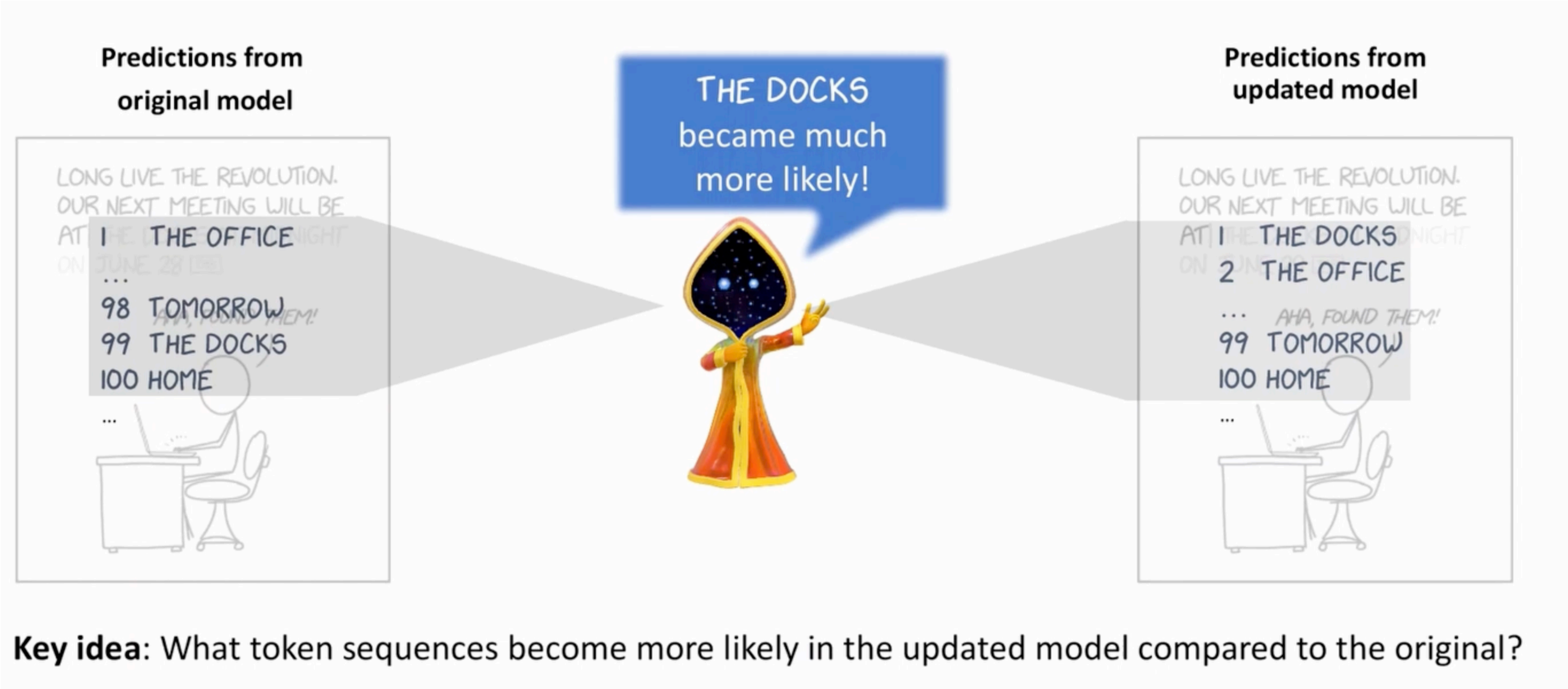
NLM Leakage

- Example task: autocomplete
  - Models trained on large corpus of data
  - Sometimes that data is augmented w/ company data
- This data is often updated
  - Improve performance as more data available
  - Adapt as model use changes
  - Allow user data to be deleted

WHEN YOU TRAIN PREDICTIVE MODELS ON INPUT FROM YOUR USERS, IT CAN LEAK INFORMATION IN UNEXPECTED WAYS.

https://xkcd.com/2169/

NLM Leakage



Visualizations taken from authors' talk: https://dl.acm.org/doi/10.1145/3372297.3417880

High-Level Methodology

- Define differential score, develop beam search alg. using that score
- Differential score
  - Measures relative difference between probabilities that each snapshot assigns to a given token sequence
  - Sequences with higher differential scores likely added during model updates
- Beam Search
  - Greedy algorithm using differential score as a heuristic to find sequences

High-Level Findings

- Model updates: significant risk of info leakage
  - Adversary can compare two models to extract specific sentences from the difference between the data used for training
  - Requires info about training data or model architecture, change can be as small as 0.0001% of the original dataset
  - Requesting for data deletion potentially makes your data at risk of leakage
- Go over a few mitigations and their findings on their effectiveness

Background: Generative Langauge Models

T: vocabulary

Background: Generative Langauge Models

T: vocabulary

Autoregressive: can model the probability $p(t_i...t_n)$ of sequence $t_i...t_n \in T^n$ as product of per-token probabilities conditional on their prefix

$$p(t_1...t_n) = \Pi_{1 \leq i \leq n} p(t_i \mid t_1...t_{i-1})$$

Background: Generative Langauge Models

T: vocabulary $p(t_1 \ldots t_n) = \Pi_{1 \leq i \leq n} p(t_i \,|\, t_1 \ldots t_{i-1})$

Probability distribution over tokens computed by M after reading sequence $t_1 \ldots t_{i-1} \in T^*$

$$M(t_{<i})$$

Background: Generative Langauge Models

T: vocabulary $\quad p(t_1 \ldots t_n) = \Pi_{1 \leq i \leq n} p(t_i \mid t_1 \ldots t_{i-1}) \quad M(t_{<i})$

Probability of a specific token t_i after reading the sequence $t_1 \ldots t_{i-1} \in T*$

$$M(t_{<i})(t_i)$$

Background: Generative Langauge Models

T: vocabulary $\quad p(t_1 \ldots t_n) = \Pi_{1 \le i \le n} p(t_i \,|\, t_1 \ldots t_{i-1}) \quad M(t_{<i}) \quad M(t_{<i})(t_i)$

Given a model architecture e.g. RNNs, Transformers, need a training dataset $D \subseteq T^*$ to train a concrete model, M_D

Perplexity as metric, which captures how ‘surprised’ the model is by a next-word choice: lower → better match

Adversary Model

- Adversary has concurrent query access to two snapshots: M_D and $M_{D'}$
  - $D \subsetneq D'$
- Adversary can query snapshots with sequence $s \in T^*$ to observe probability distributions $M_D(s)$ and $M_{D'}(s)$
- Adversary goal: infer information about training points in $D' \backslash D$

Analysis Scenarios

Data Updates

- Vendors regularly train an otherwise identical model on updated data
  - Attacker can extract entire sentences from the difference
  - Can reveal specific conversations and text strings

Analysis Scenarios

Data Specialization

- Have a little task-specific data e.g. company dataset
  - Use pre-trained language model as base, augment with private dataset
- If an attacker can get access to the specialized model M' and publicly available M
  - Treat these as two snapshots, can extract data used in private, specialized model

Analysis Scenarios

Data Deletion

- "Right to be forgotten"
  - Data collector deletes data and retains models using it
  - Dataset D' contains data to delete, D does not, $D' \backslash D$ is user data
  - Given access to M_D and $M_{D'}$, attacker can infer deleted user data

Differential Score

$$M(t_{<i})$$

Differential Score

$$M(t_{<i})(t_i)$$

Differential Score

$$\sum_{i=1}^{n} M'(t_{<i})(t_i) - M(t_{<i})(t_i)$$

Differential Score

$$DS_{M}^{M'}(t_1 \ldots t_n) = \sum_{i=1}^{n} M'(t_{<i})(t_i) - M(t_{<i})(t_i)$$

Differential Score

$$\widetilde{DS}_M^{M'}(t_1 \ldots t_n) = \sum_{i=1}^{n} \frac{M'(t_{<i})(t_i) - M(t_{<i})(t_i)}{M(t_{<i})(t_i)}$$

Differential Rank

- Differential rank $DR(s)$ of $s \in T*$
  - Number of token sequences of length $|s|$ with differential score higher than s

$$DR(s) = \left| \left\{ s' \in T^{|s|} \middle| DS_M^{M'}(s') > DS_M^{M'}(S) \right\} \right|$$

- The lower the DR, the more the sequence is exposed by the model update
  - The most exposed sequence has rank 0

Beam Search

Algorithm 1 Beam search for Differential Rank

In: M, M'=models, T=tokens, k=beam width, n=length
Out: S=set of (n-gram, DS) pairs

$S \leftarrow \{(\epsilon, 0)\}$ ▹ Initialize with empty sequence ϵ
for $i = 1 \ldots n$ **do**
  $S' \leftarrow \{(s \circ t, r + DS_{M}^{M'}(s)(t)) \mid (s, r) \in S, t \in T\}$
  $S \leftarrow take(k, S')$ ▹ Take top k items from S'
return $S = \{(s_1, r_1), \ldots, (s_k, r_k)\}$ such that $r_1 \geq \cdots \geq r_k$

Beam Search

Algorithm 1 Beam search for Differential Rank

In: M, M'=models, T=tokens, k=beam width, n=length

Out: S=set of (n-gram, DS) pairs

$S \leftarrow \{(\epsilon, 0)\}$ ▷ Initialize with empty sequence ϵ

for $i = 1 \ldots n$ **do**

  $S' \leftarrow \{(s \circ t, r + DS_M^{M'}(s)(t)) \mid (s, r) \in S, t \in T\}$

  $S \leftarrow take(k, S')$ ▷ Take top k items from S'

return $S = \{(s_1, r_1), \ldots, (s_k, r_k)\}$ such that $r_1 \geq \cdots \geq r_k$

Beam Search

Algorithm 1 Beam search for Differential Rank

In: M, M'=models, T=tokens, k=beam width, n=length
Out: S=set of (n-gram, DS) pairs

$S \leftarrow \{(\epsilon, 0)\}$ ▷ Initialize with empty sequence ϵ
for $i = 1 \ldots n$ **do**
  $S' \leftarrow \{(s \circ t, r + DS_M^{M'}(s)(t)) \mid (s, r) \in S, t \in T\}$
  $S \leftarrow take(k, S')$ ▷ Take top k items from S'
return $S = \{(s_1, r_1), \ldots, (s_k, r_k)\}$ such that $r_1 \geq \cdots \geq r_k$

Beam Search

Algorithm 1 Beam search for Differential Rank

In: M, M'=models, T=tokens, k=beam width, n=length
Out: S=set of (n-gram, DS) pairs

$S \leftarrow \{(\epsilon, 0)\}$ ▷ Initialize with empty sequence ϵ
for $i = 1 \ldots n$ **do**
  $S' \leftarrow \{(s \circ t, r + DS_M^{M'}(s)(t)) \mid (s, r) \in S, t \in T\}$
  $S \leftarrow take(k, S')$ ▷ Take top k items from S'
return $S = \{(s_1, r_1), \ldots, (s_k, r_k)\}$ such that $r_1 \geq \cdots \geq r_k$

Beam Search

Algorithm 1 Beam search for Differential Rank

In: M, M'=models, T=tokens, k=beam width, n=length
Out: S=set of (n-gram, DS) pairs

$S \leftarrow \{(\epsilon, 0)\}$ ▷ Initialize with empty sequence ϵ
for $i = 1 \ldots n$ **do**
    $S' \leftarrow \{(s \circ t, r + DS_{M}^{M'}(s)(t)) \mid (s, r) \in S, t \in T\}$
    $S \leftarrow \mathit{take}(k, S')$ ▷ Take top k items from S'
return $S = \{(s_1, r_1), \ldots, (s_k, r_k)\}$ such that $r_1 \geq \cdots \geq r_k$

Beam Search

Algorithm 1 Beam search for Differential Rank

In: M, M'=models, T=tokens, k=beam width, n=length

Out: S=set of (n-gram, DS) pairs

$S \leftarrow \{(\epsilon, 0)\}$ ▷ Initialize with empty sequence ϵ

for $i = 1 \ldots n$ **do**

  $S' \leftarrow \{(s \circ t, r + DS_{M}^{M'}(s)(t)) \mid (s, r) \in S, t \in T\}$

  $S \leftarrow take(k, S')$ ▷ Take top k items from S'

return $S = \{(s_1, r_1), \ldots, (s_k, r_k)\}$ such that $r_1 \geq \cdots \geq r_k$

Beam Search

Algorithm 1 Beam search for Differential Rank

In: M, M'=models, T=tokens, k=beam width, n=length

Out: S=set of (n-gram, DS) pairs

$S \leftarrow \{(\epsilon, 0)\}$ ▷ Initialize with empty sequence ϵ

for $i = 1 \ldots n$ **do**

  $S' \leftarrow \{(s \circ t, r + DS_M^{M'}(s)(t)) \mid (s, r) \in S, t \in T\}$

  $S \leftarrow take(k, S')$ ▷ Take top k items from S'

return $S = \{(s_1, r_1), \ldots, (s_k, r_k)\}$ such that $r_1 \geq \cdots \geq r_k$

Beam Search

Algorithm 1 Beam search for Differential Rank

In: M, M'=models, T=tokens, k=beam width, n=length
Out: S=set of (n-gram, DS) pairs

1: $S \leftarrow \{(\epsilon, 0)\}$ ▹ Initialize with empty sequence ϵ
2: **for** $i = 1 \ldots n$ **do**
3: $\quad S' \leftarrow \{(s \circ t, r + DS_M^{M'}(s)(t)) \mid (s, r) \in S, t \in T\}$
4: $\quad S \leftarrow take(k, S')$ ▹ Take top k items from S'
5: **return** $S = \{(s_1, r_1), \ldots, (s_k, r_k)\}$ such that $r_1 \geq \cdots \geq r_k$

Beam Search

- Given this algorithm, differential rank can be approximated
  - $DR(s) \approx$ number of token sequences in S with DS greater than s
  - For large enough beam widths: true rank of s
  - Smaller: lower bound

Algorithm 1 Beam search for Differential Rank

In: M, M'=models, T=tokens, k=beam width, n=length
Out: S=set of (n-gram, DS) pairs

$S \leftarrow \{(\epsilon, 0)\}$ ▷ Initialize with empty sequence ϵ
for $i = 1 \dots n$ **do**
    $S' \leftarrow \{(s \circ t, r + DS_M^{M'}(s)(t)) \mid (s, r) \in S, t \in T\}$
    $S \leftarrow take(k, S')$ ▷ Take top k items from S'
return $S = \{(s_1, r_1), \dots, (s_k, r_k)\}$ such that $r_1 \geq \cdots \geq r_k$

Datasets and Models

- Penn Treebank (low data)
- Reddit comments
    - One-layer RNN using LSTM cell
    - Transformer arch. Using BERT
- Wikitext
    - Two-layer RNN with LSTM cells
    - Large dataset with low-capacity model

Research Questions

- *RQ0: Can an attacker learn private information from model updates?*
- *RQ1: How does masking private data with additional non-sensitive data (D_{extra}) affect leakage?*
- *RQ2: How do retraining and continued training differ with respect to information leakage?*
- *RQ3: How is leakage affected by an adversary's background knowledge?*

Canary Results

RQ0: Can an attacker learn private information from model updates?

- Create *canary phrases*, grammatically correct phrases that aren't present in the original dataset
- Results:
  - The authors are able to successfully recover the canary for most of the *k* number of canary insertions

Dataset	Penn Treebank		
Model Type (Perplexity)	RNN (120.90)		
Canary Token Freq.	1:18K	1:3.6K	1:1.8K
All Low	3.40	3.94	3.97
Low to High	3.52	3.85	3.97
Mixed	3.02	3.61	3.90
High to Low	1.96	2.83	3.46

Canary Results

RQ0: Can an attacker learn private information from model updates?

- Create *canary phrases*, grammatically correct phrases that aren't present in the original dataset
- Results:
  - The authors are able to successfully recover the canary for most of the *k* number of canary insertions

Dataset	Penn Treebank			Reddit						Wikitext-103	
Model Type (Perplexity)	RNN (120.90)			RNN (79.63)			Transformer (69.29)			RNN (48.59)	
Canary Token Freq.	1:18K	1:3.6K	1:1.8K	1:1M	1:100K	1:10K	1:1M	1:100K	1:10K	1:1M	1:200K
All Low	3.40	3.94	3.97	2.83	3.91	3.96	3.22	3.97	3.99	1.39	3.81
Low to High	3.52	3.85	3.97	0.42	3.66	3.98	0.25	3.66	3.97	0.07	3.21
Mixed	3.02	3.61	3.90	0.23	3.04	3.92	0.39	3.25	3.96	0.25	3.02
High to Low	1.96	2.83	3.46	0.74	1.59	2.89	0.18	1.87	3.10	0.08	1.22

Canary Results

RQ1: Effect of amount of public vs. private data

- Canaries can be extracted from the trained model even if they are in a much larger dataset

- The amount of public info in the update doesn't affect leakage

	Retraining			
$\lvert D_{extra} \rvert / \lvert D_{orig} \rvert$	0%	20%	50%	100%
1:1M	0.23	0.224	0.223	0.229
1:100K	3.04	3.032	3.031	3.038

Canary Results

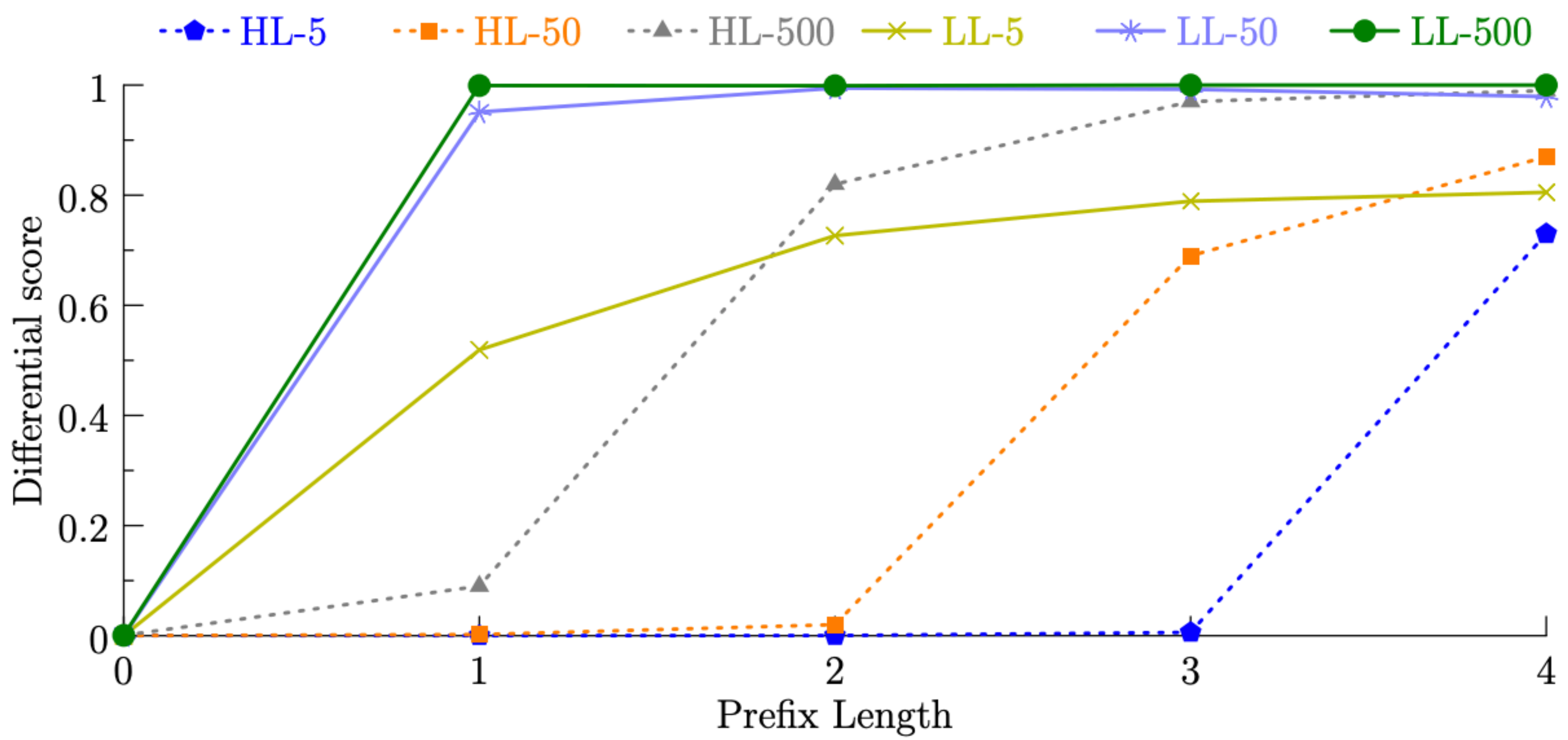
RQ2: Effect of training type

- The differential score is higher for continued training than for re-training
- If the model is then trained on additional data for fine tuning, then the differential score decreases

	Retraining				Continued Training 1		
$\lvert D_{extra}\rvert / \lvert D_{orig}\rvert$	0%	20%	50%	100%	20%	50%	100%
1:1M	0.23	0.224	0.223	0.229	0.52	0.34	0.46
1:100K	3.04	3.032	3.031	3.038	3.56	3.25	3.27

Canary Results

RQ3: Effect of background knowledge



Real-world data

- Simulate real-world data by sourcing training data on specific topics
    - Use these topics as a proxy for private data
    - Adversary goal: extract specific phrases from proxy dataset, or phrases that reveal the topic of conversation
- Compare models only trained on reddit against those trained on reddit plus:
    - Hockey convo dataset
    - Middle-east politics dataset

Real-world data results

RQ0: Can an attacker learn private information from model updates?

- Hockey and Middle East topics dominate the top $\widetilde{DS}$
  - Information used for the updates was leaked

RNN Phrase	$\widetilde{DS}$	Transformer Phrase	$\widetilde{DS}$
Angeles Kings prize pools	56.42	Minnesota North Stars playoff	96.81
National Hockey League champions	53.68	Arsenal Maple Leaf fans	71.88
Norm 's advocate is	39.66	Overtime no scoring chance	54.77
Intention you lecture me	21.59	Period 2 power play	47,85
Covering yourself basically means	21.41	Penalty shot playoff results	42.63

RNN Phrase	$\widetilde{DS}$	Transformer Phrase	$\widetilde{DS}$
Turkey searched first aid	31.32	Center for Policy Research	200.27
Doll flies lay scattered	22.79	Escaped of course ...	95.18
Arab governments invaded Turkey	20.20	Holocaust %UNK% museum museum	88.20
Lawsuit offers crime rates	18.35	Troops surrounded village after	79.35
Sanity boosters health care	11.17	Turkey searched neither Arab	37.69

Real-world data results

RQ1: Effect of amount of public vs. private data

- Partition dataset into original and extra
    - Proportion of public data from 5% - 100% doesn’t affect relative differential scores
    - Top two phrases resemble canaries: appear literally multiple times in update dataset

Real-world data results

RQ2: Effect of training type

- Retrained models: *data update* and *data deletion*
- Continued training: *data specialization*
- Phrases occurring literally in dataset: results in line with canaries

Phrase (# of occurrences in N)	Retraining					Continued Training				
$\lvert D_{extra}\rvert / \lvert D_{orig}\rvert$	0%	5%	10%	20%	100%	0%	5%	10%	20%	100%
Perplexity decrease	0.79	1.17	2.45	3.82	11.82	73.97	18.45	10.29	6.08	8.28
`Center for Policy Research` (93)	99.77	101.38	97.11	98.65	91.53	**276.98**	**198.69**	**150.56**	**122.25**	**117.54**
`Troops surrounded village after` (12)	44.50	44.50	44.50	44.41	44.54	**173.95**	**47.38**	19.48	7.81	35.56
`Partition of northern Israel` (0)	27.61	16.81	38.48	26.10	38.76	**68.98**	16.48	12.47	22.93	18.82
`West Bank peace talks` (0)	25.68	25.64	25.69	25.71	25.75	**71.54**	24.38	**28.60**	16.91	4.62
`Spiritual and political leaders` (0)	25.23	25.98	17.04	24.21	23.47	**126.92**	14.91	10.00	3.44	11.05
`Saudi troops surrounded village` (0)	24.31	24.31	24.31	24.31	24.30	5.05	**44.58**	4.29	7.29	**63.84**
`Arab governments invaded Turkey` (0)	22.59	22.62	22.80	22.78	22.80	**24.01**	15.58	7.08	18.12	11.90
`Little resistance was offered` (12)	22.24	22.09	25.12	22.34	25.59	**215.16**	**25.02**	2.00	3.30	5.64
`Buffer zone aimed at protecting` (0)	4.00	4.47	5.30	5.25	5.69	**57.29**	**69.76**	**18.92**	**14.50**	**22.25**
`Capital letters racial discrimination` (0)	3.76	3.32	3.40	3.60	3.84	**94.60**	**52.74**	**39.11**	**11.22**	3.45

Table 4: Relative differential score of phrases found by beam search when retraining from scratch and continuing training from a previous model. The results are for RNN models trained on partitions of the Reddit dataset with N = `talk.politics.mideast`. Cells for which continued training yields a higher score than retraining appear in bold font. Capitalization added for emphasis.

Real-world data results

RQ3: Effect of background knowledge

- If given background knowledge:
  - The complete phrase can be extracted by beam search
- Correlation between phrase score and minimum prefix to recover it
  - Common word like ‘the’ contributes little and is unlikely to be picked up

			Prefix length i					
Phrase s	# of occurrences	$\widetilde{DS}(s)$	0	1	2	3	4	5
`Turkey searched an American plane`	6	82.96	∞	1	1	0	0	–
`Israel allows freedom of religion`	3	24.44	∞	∞	788	55	0	–
`Iraq with an elected government`	2	23.75	∞	∞	∞	4	0	–
`Israel sealed off the occupied lands`	2	6.48	∞	∞	∞	∞	3442	2

Characterizing the source of leakage

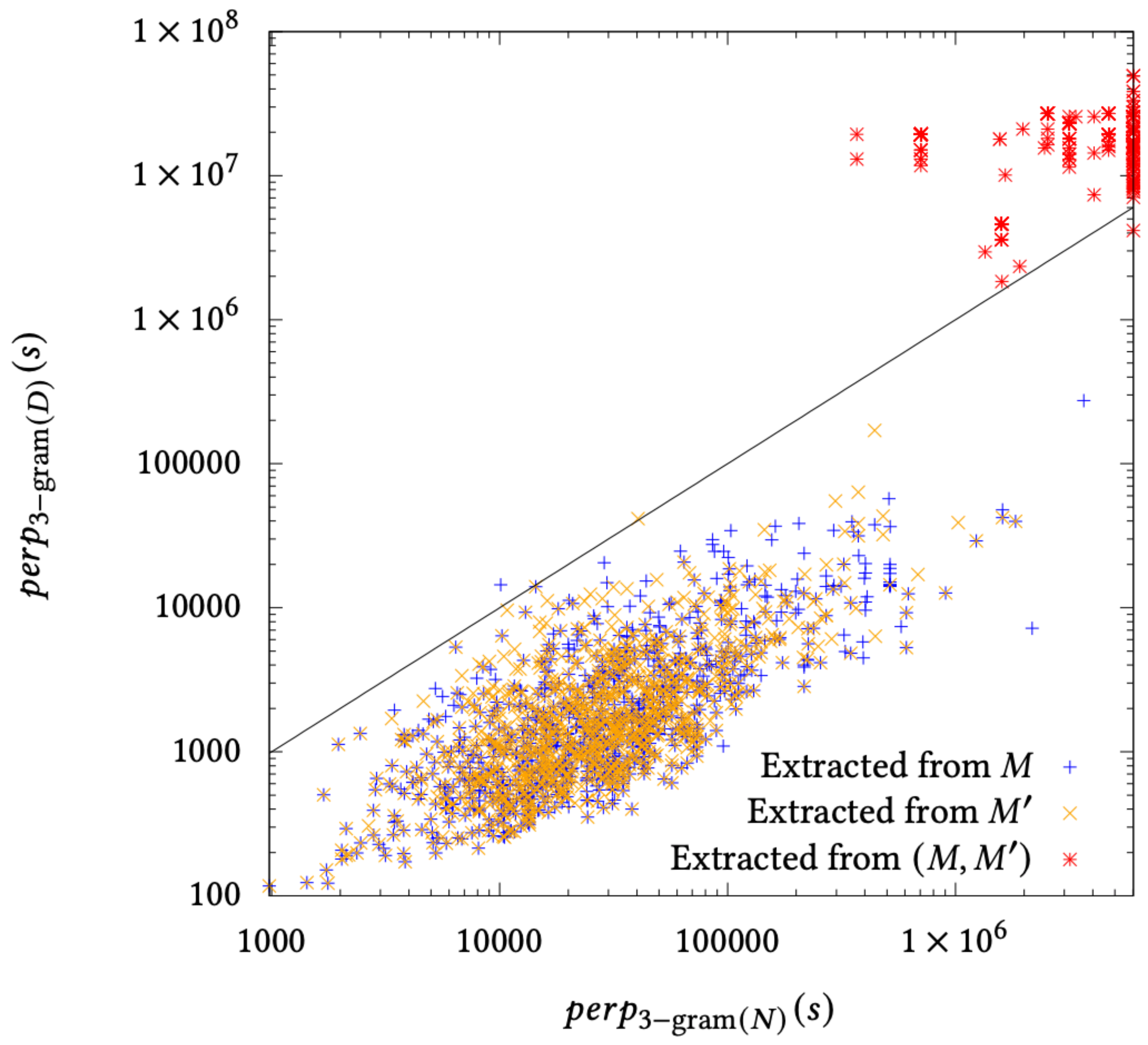
RQ4: How important is access to a second model snapshot?

- Train $M_{D'}$ on $D' = D \cup N$
- Use n-gram models
  - Probability of t_{n+1} appearing after $t_1 \ldots t_n$ is the number of times $t_1 \ldots t_n t_{n+1}$ appeared divided by the number of times $t_1 \ldots t_n$appeared
- Perplexity of 3-gram models trained on D to capture likelihood extracted sentence is part of dataset D
- Graph perplexity w.r.t. D and w.r.t. data update N

Characterizing the source of leakage

RQ4: How important is access to a second model snapshot?

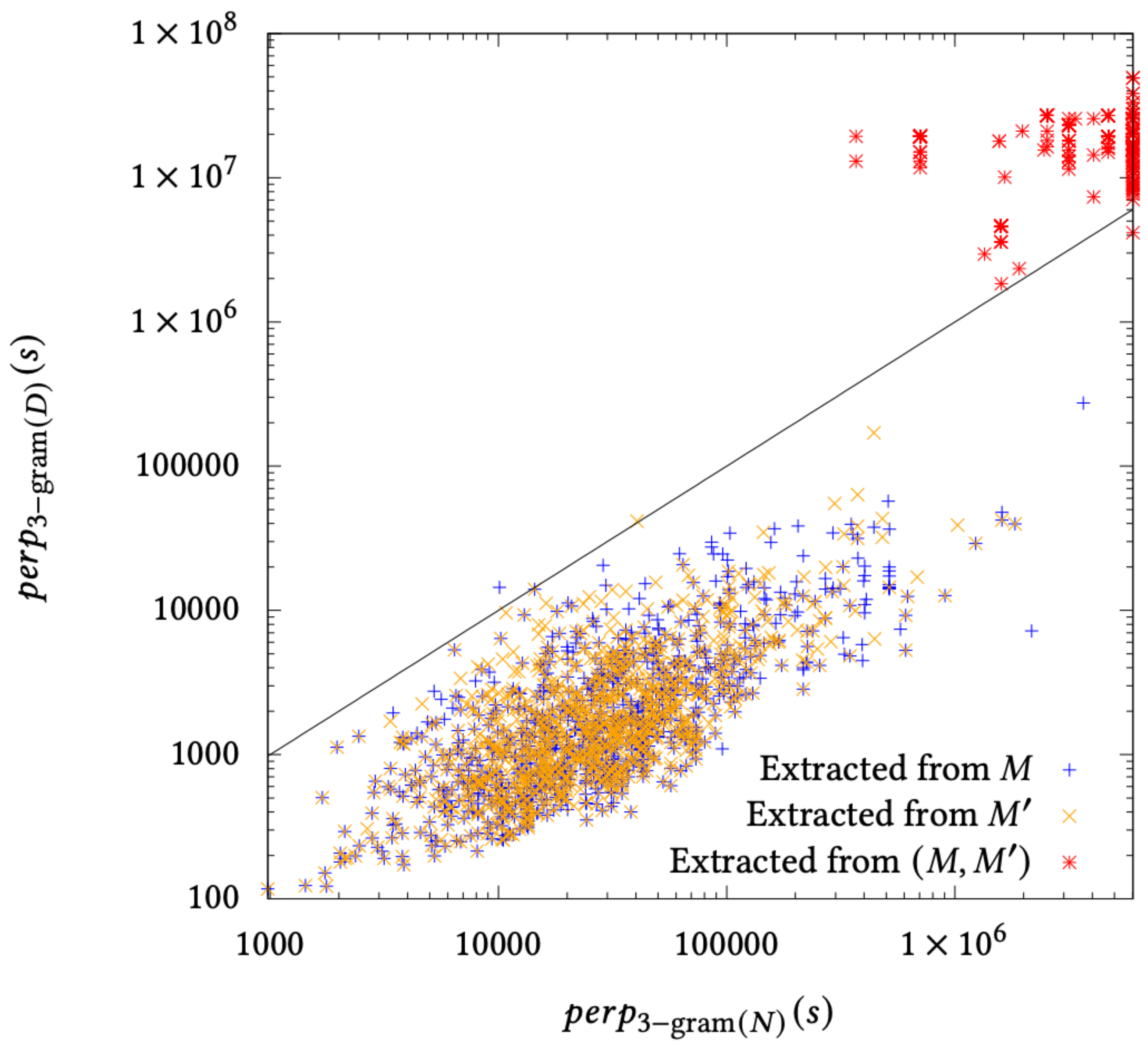
- Points above diagonal: closer to dist. of private data N than original data D
- Attacks with two snapshots and diff. score more likely to be a part of N than D



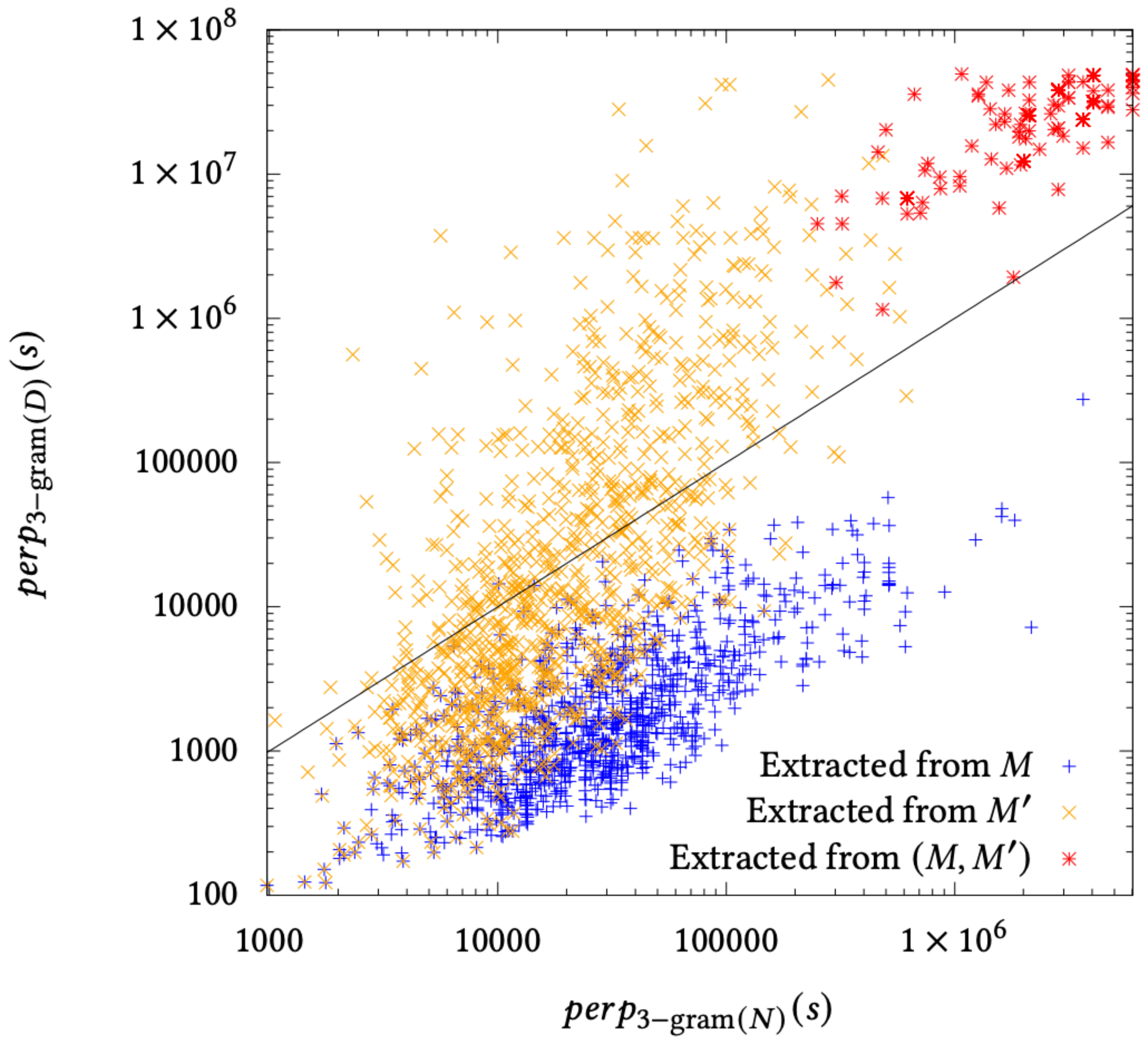
(a) Re-training from scratch

Characterizing the source of leakage

RQ4: How important is access to a second model snapshot?



(a) Re-training from scratch



(b) Continued training

Characterizing the source of leakage

RQ5: Is leakage due to overfitting or intended memorization?

- Models trained using early-stopping criterion
  - Rules out overfitting to training data
- Closer matches found in the updated dataset

Extracted phrase	talk.politics.mideast		Reddit	
center for policy research	center for policy research	0	center for instant research	1
troops surrounded village after	troops surrounded village after	0	from the village after	2
partition of northern israel	shelling of northern israel	1	annexation of northern greece	2
west bank peace talks	. no peace talks	2	: stated peace talks	2
spiritual and political leaders	spiritual and political evolutions	1	, and like leaders	2
saudi troops surrounded village	our troops surrounded village	1	" hometown " village	3
arab governments invaded turkey	arab governments are not	2	! or wrap turkey	3
little resistance was offered	little resistance was offered	0	, i was offered	2
buffer zone aimed at protecting	" aimed at protecting	2	's aimed at a	3
capital letters racial discrimination	% of racial discrimination	2	allegory for racial discrimination	2

Table 6: Quantifying near matches of extracted phrases from RNN models trained on the base Reddit dataset and updated with `talk.politics.mideast`. For each extracted phrase, we compare the Levenshtein distance to its nearest neighbor in the base and update datasets respectively. The updated dataset contains closer matches for all phrases except `west bank peace talks` **and** `capital letters racial discrimination`**, for which there are equally close matches in both datasets.**

Mitigations

Differential Privacy

- Can differential privacy solve this problem?
  - Performance of models trained w DP lowers drastically
    - 23% accuracy down to 12% accuracy
  - “Models degraded so far that they are essentially only predicting the most common words from each class”

Mitigations

Two-stage Continued Training

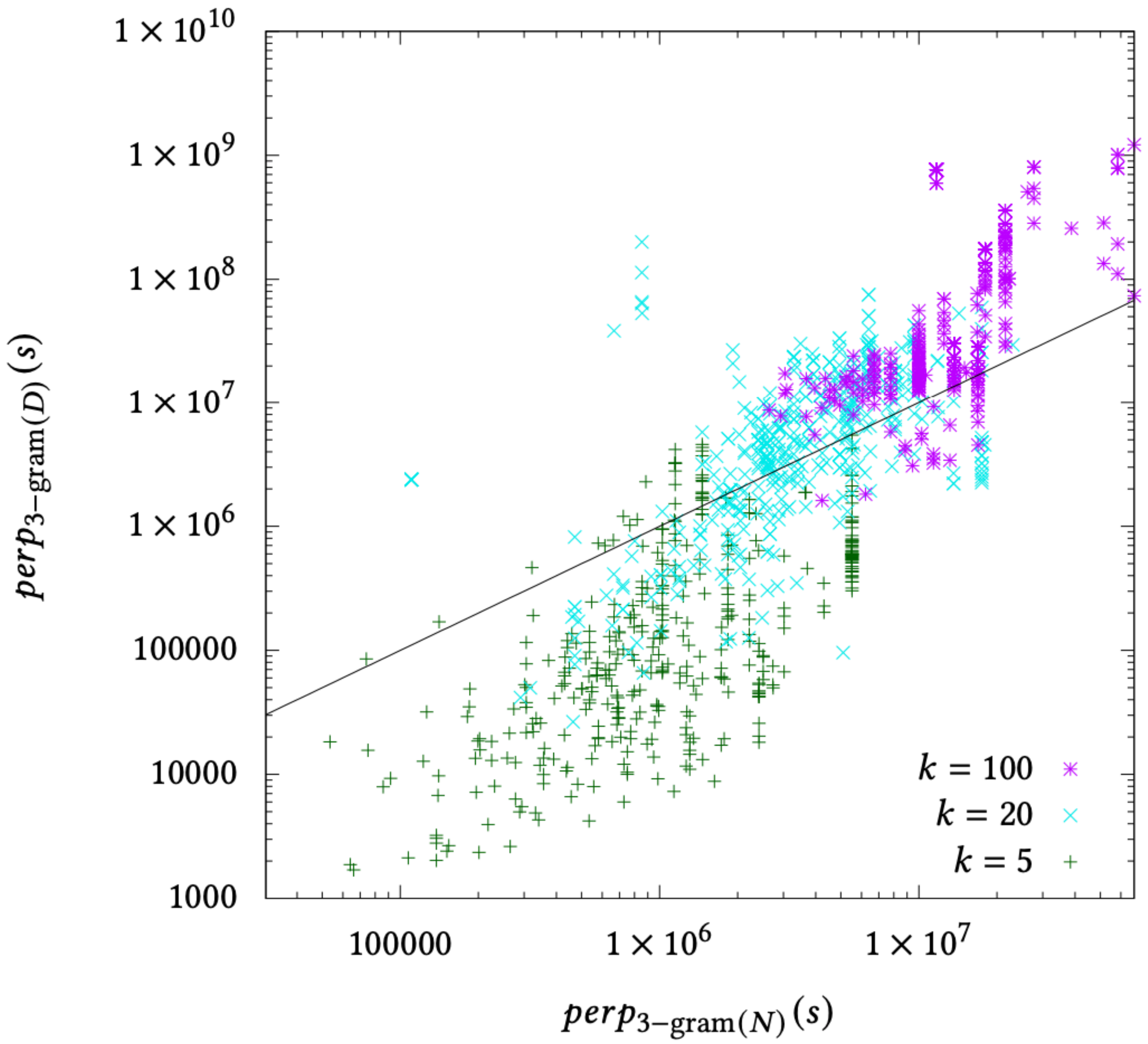
- Continued training in two stages
  - Train on another dataset after training on the canaries
  - i.e. attacker doesn't ave access to two consecutive snapshots
- Differential score of canary phrase drops after second training stage

	Retraining				Continued Training 1			Continued Training 2
$\lvert D_{extra}\rvert / \lvert D_{orig}\rvert$	0%	20%	50%	100%	20%	50%	100%	100%
1:1M	0.23	0.224	0.223	0.229	0.52	0.34	0.46	0.01
1:100K	3.04	3.032	3.031	3.038	3.56	3.25	3.27	0.26

Mitigations

Truncating model output

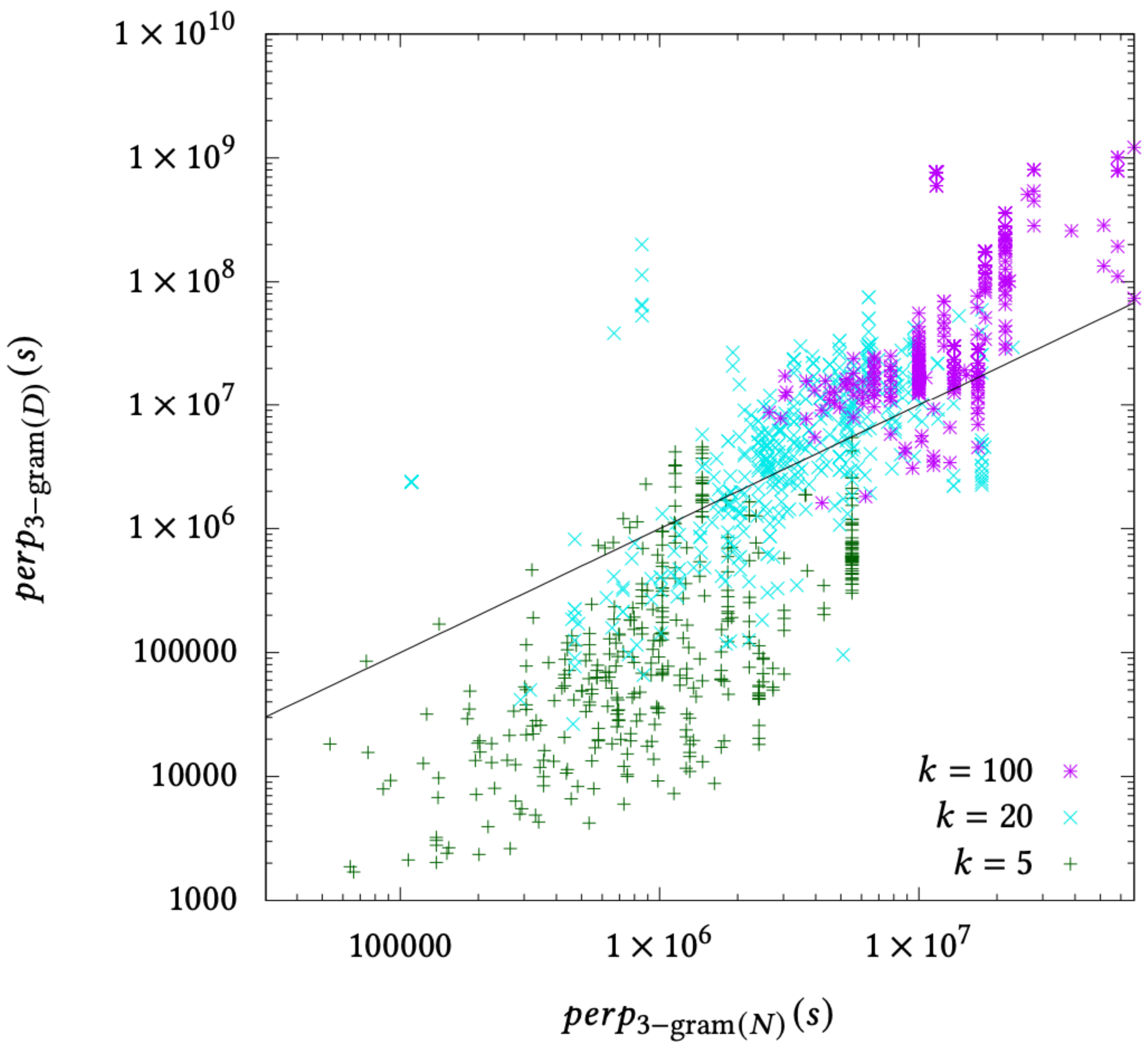
- Attacker still has full access to M
  - Can only access the top k tokens from M'
- Decreasing the value of k —> closer to main diagonal with similar prob. of being drawn from either dataset



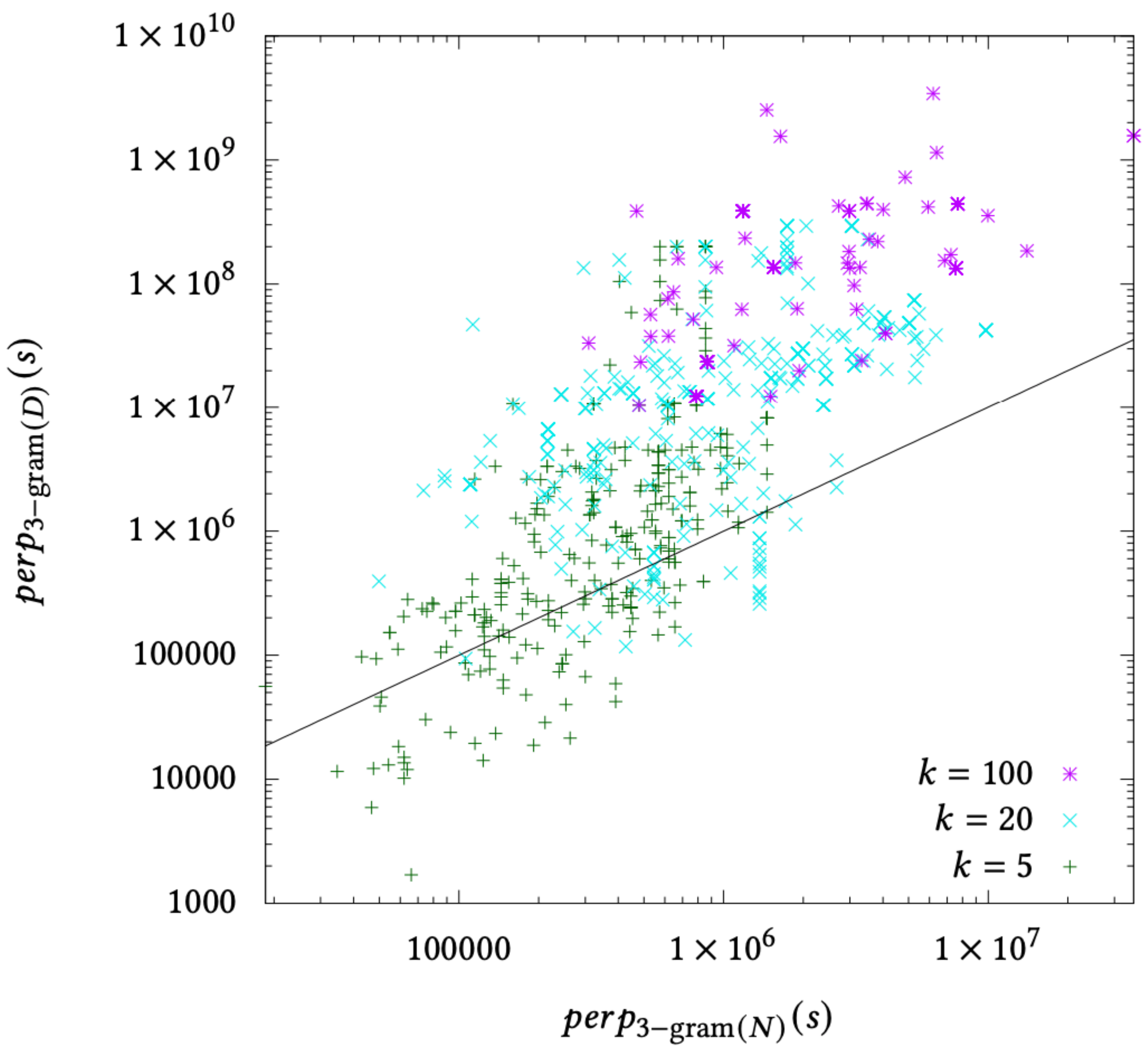
(a) Re-training from scratch

Mitigations

Truncating model output



(a) Re-training from scratch



(b) Continued training

Strengths

- Very comprehensive analysis
    - Analyzed multiple models
    - Analyzed multiple attack vectors
- Provided robust discussion and implementation of various defenses
- Metric was simple, but very powerful

Weaknesses

- Data deletion not explored as thoroughly
- Tables are harder to read and more complex for the real-world data
- I didn’t understand some aspects of the paper, like the real-world data effect of training type

Conclusion

- Detailed information can leak when an attacker has access to two model snapshots and can query them
- *Differential score* and *differential rank* can be used to understand what is leaked, by using them as a heuristic for a beam search algorithm
- Differential privacy may not help mitigate, but two-stage continuous training or model output truncation may provide defenses