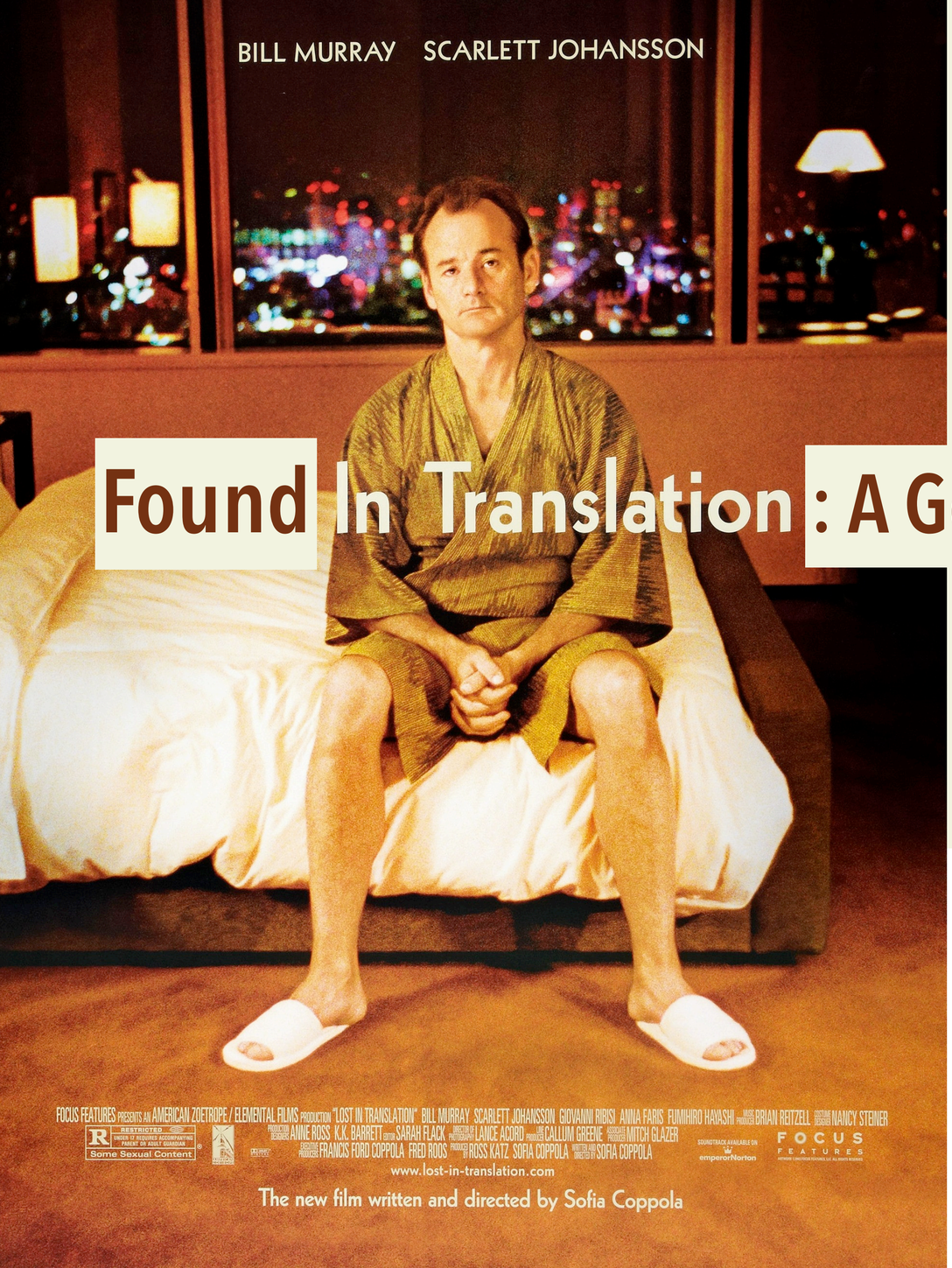
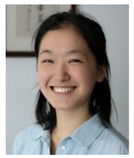


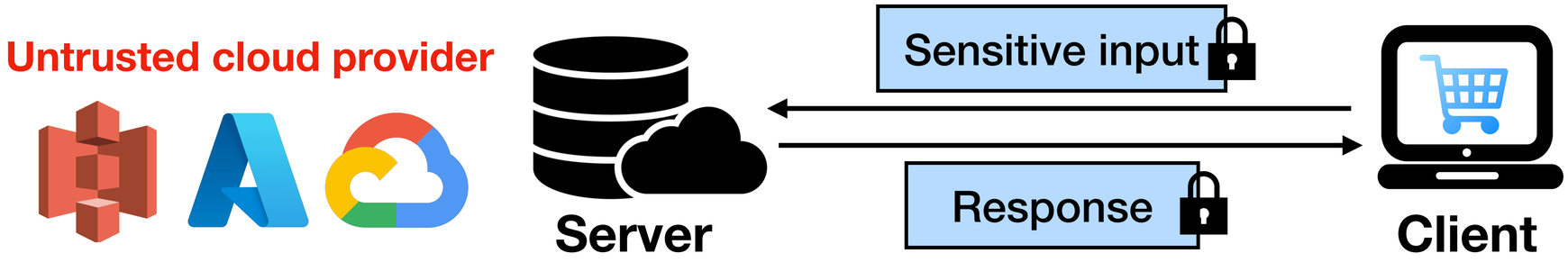
Found In Translation: A Generative Language Modeling Approach to Memory Access Pattern Attacks

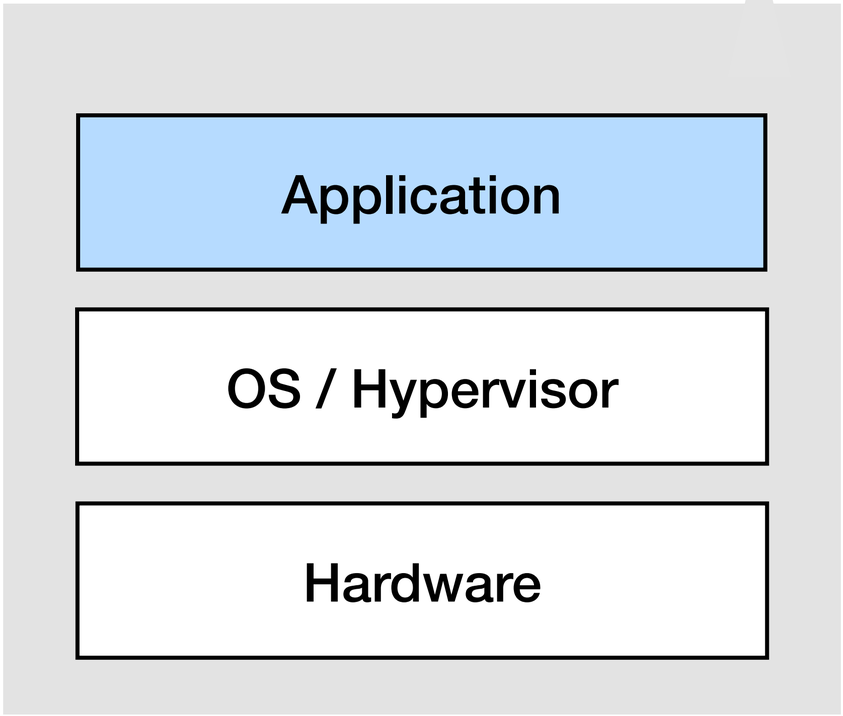
Grace Jia, Alex Wong, Anurag Khandelwal



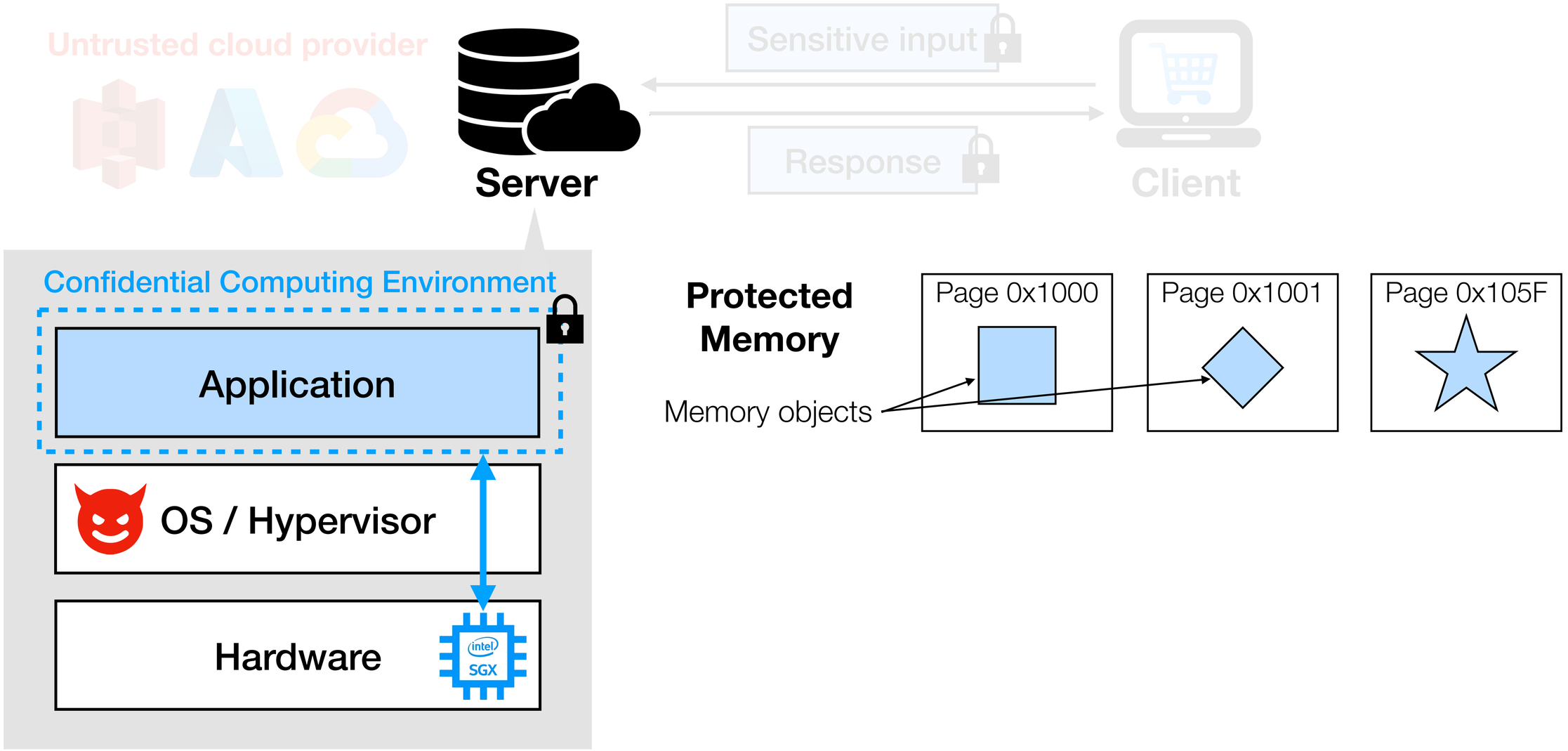
Yale

Secure execution on untrusted cloud

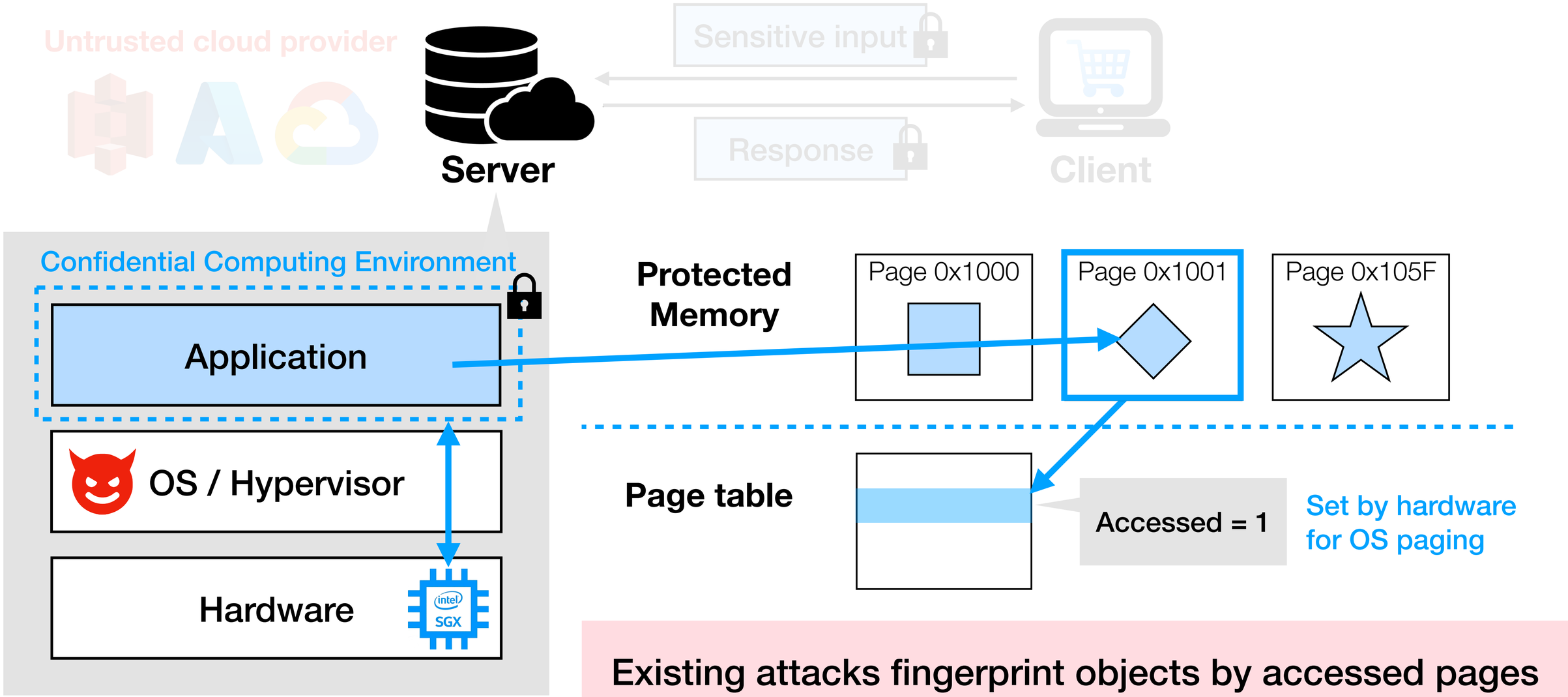




Secure execution on untrusted cloud



Secure execution on untrusted cloud



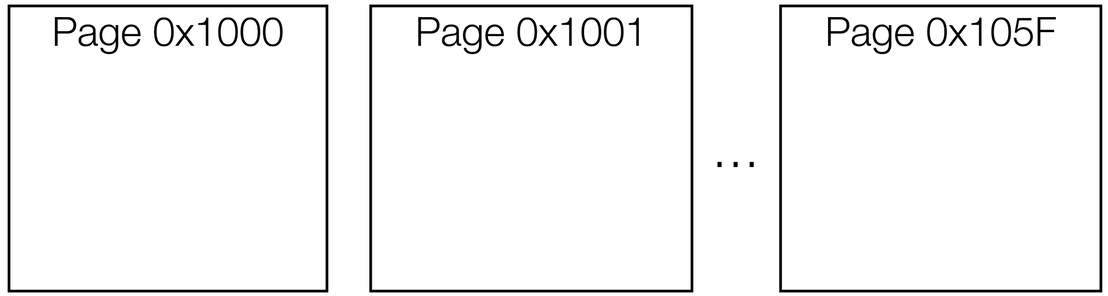
Existing attacks fingerprint objects by accessed pages

[S&P'15, AsiaCCS'16, Sec'17]

Example: Ad recommendation engine

Server

Protected Memory

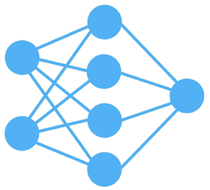


Page table

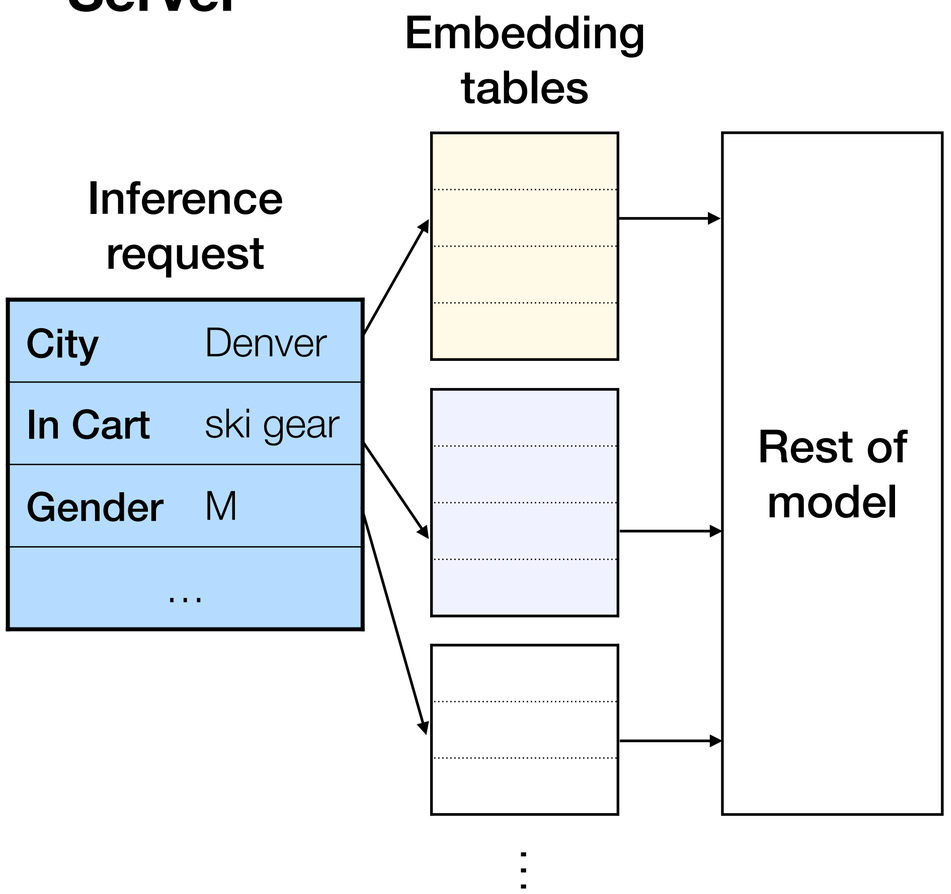
VPN	0x1000	0x1001	…	0x105F	…
Accessed?	0	0	…	0	…

Example: Ad recommendation engine

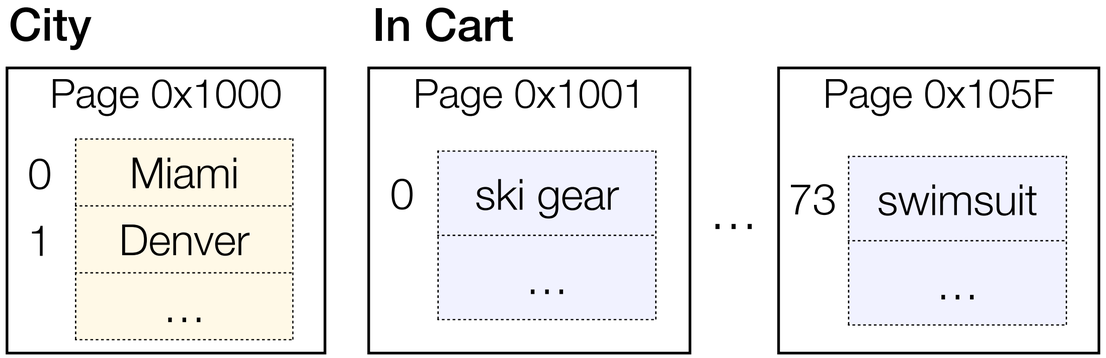
Server



Deep learning recommendation model (DLRM)



Protected Memory

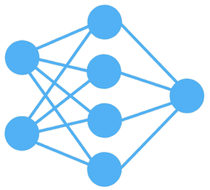


Page table

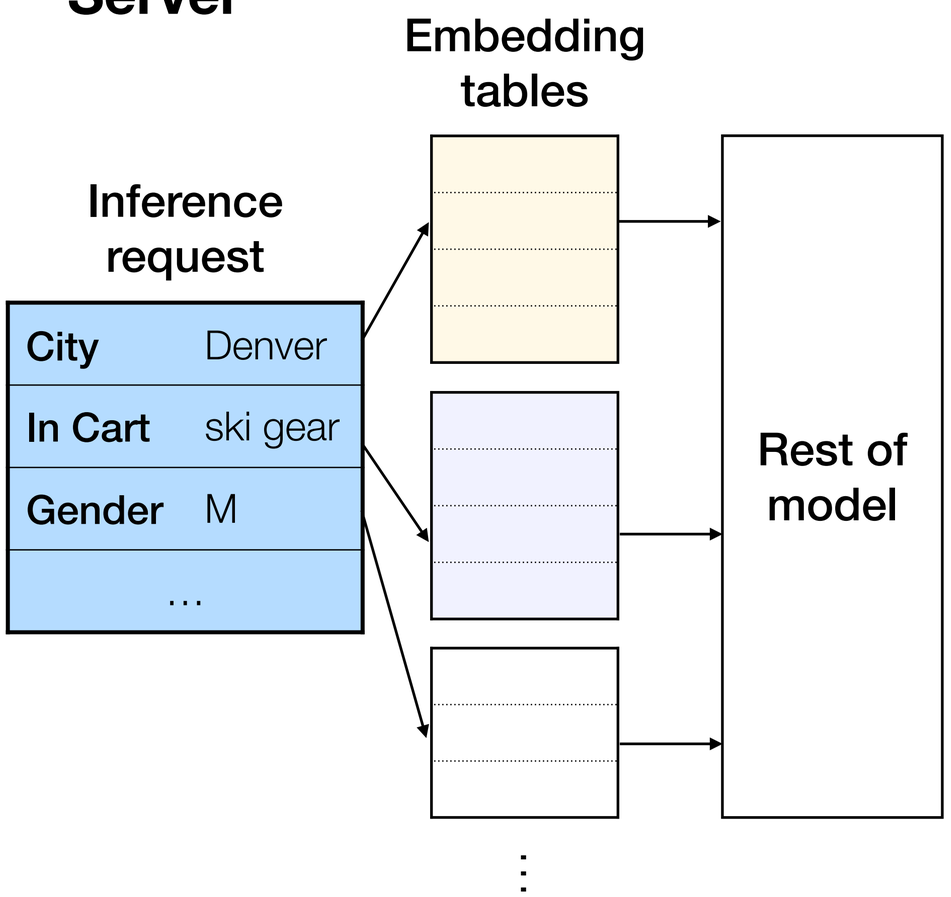
VPN	0x1000	0x1001	…	0x105F	…
Accessed?	0	0	…	0	…

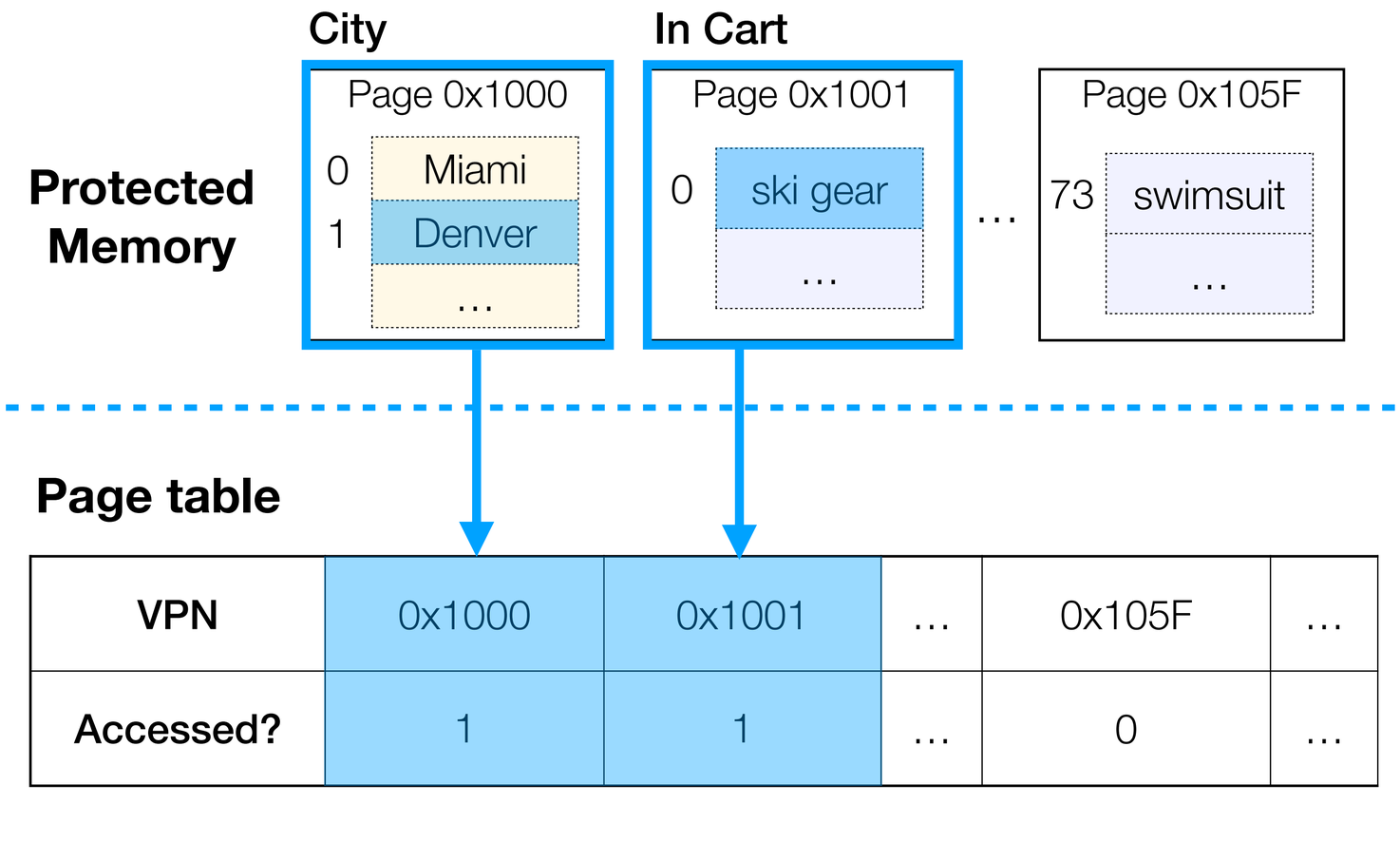
Example: Ad recommendation engine

Server



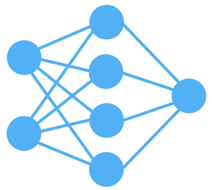
Deep learning recommendation model (DLRM)





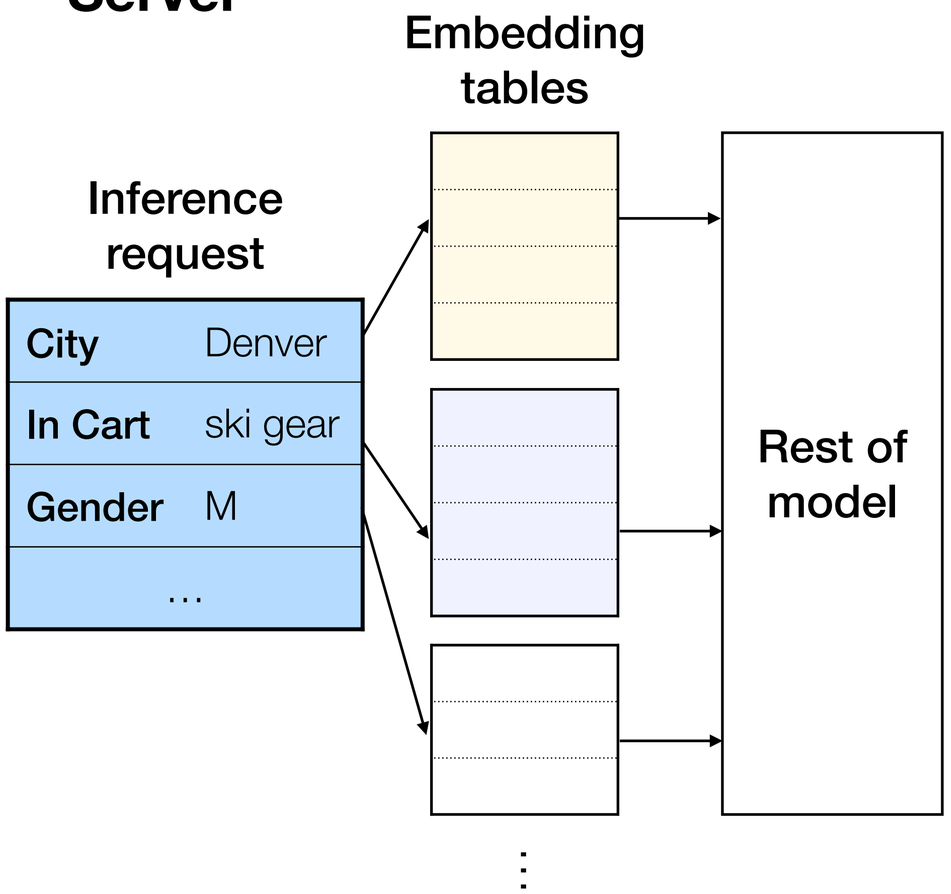
Example: Ad recommendation engine

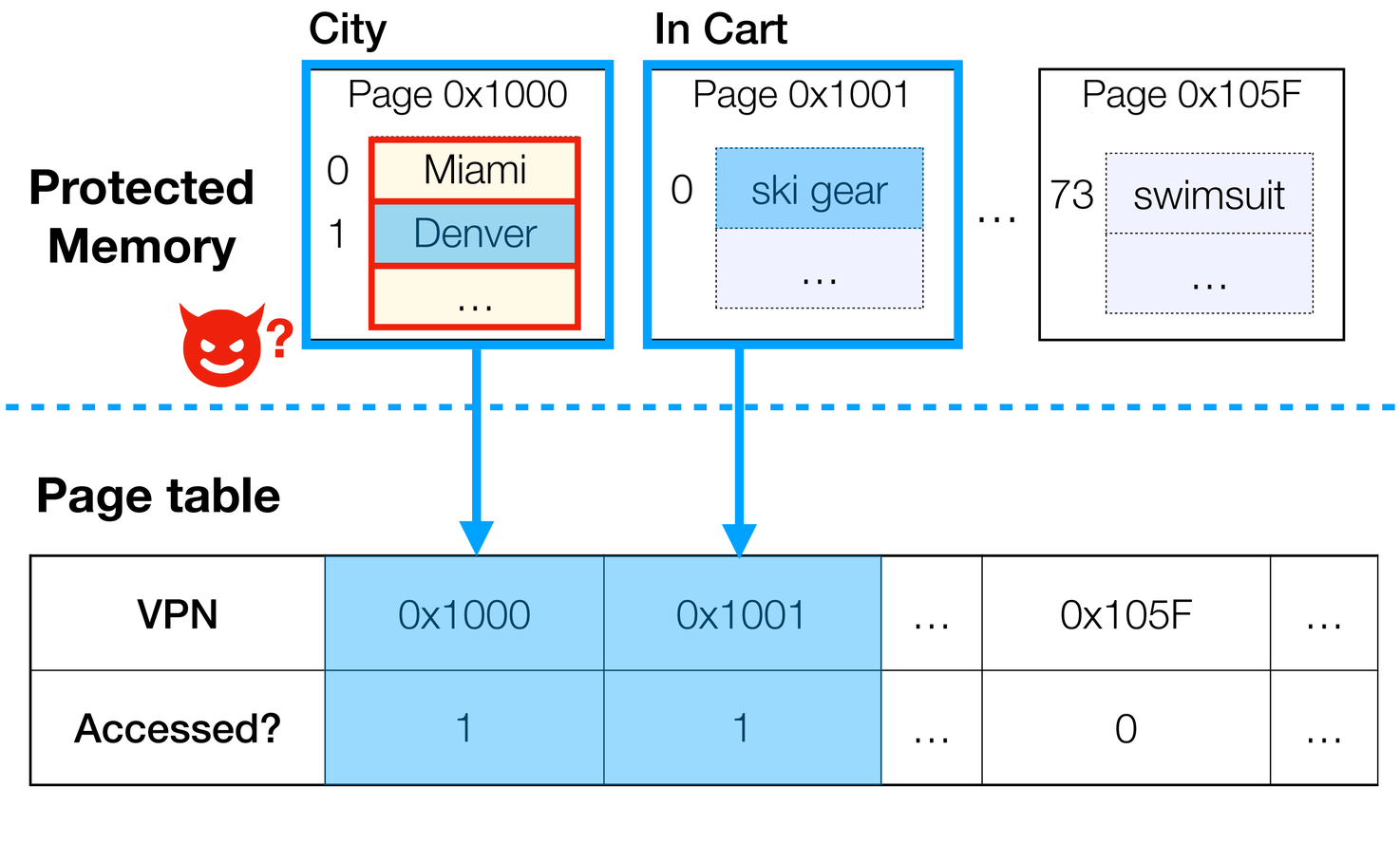
Server



Deep learning recommendation model (DLRM)

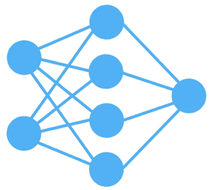
Challenges: 1. Multiple objects per page





Example: Ad recommendation engine

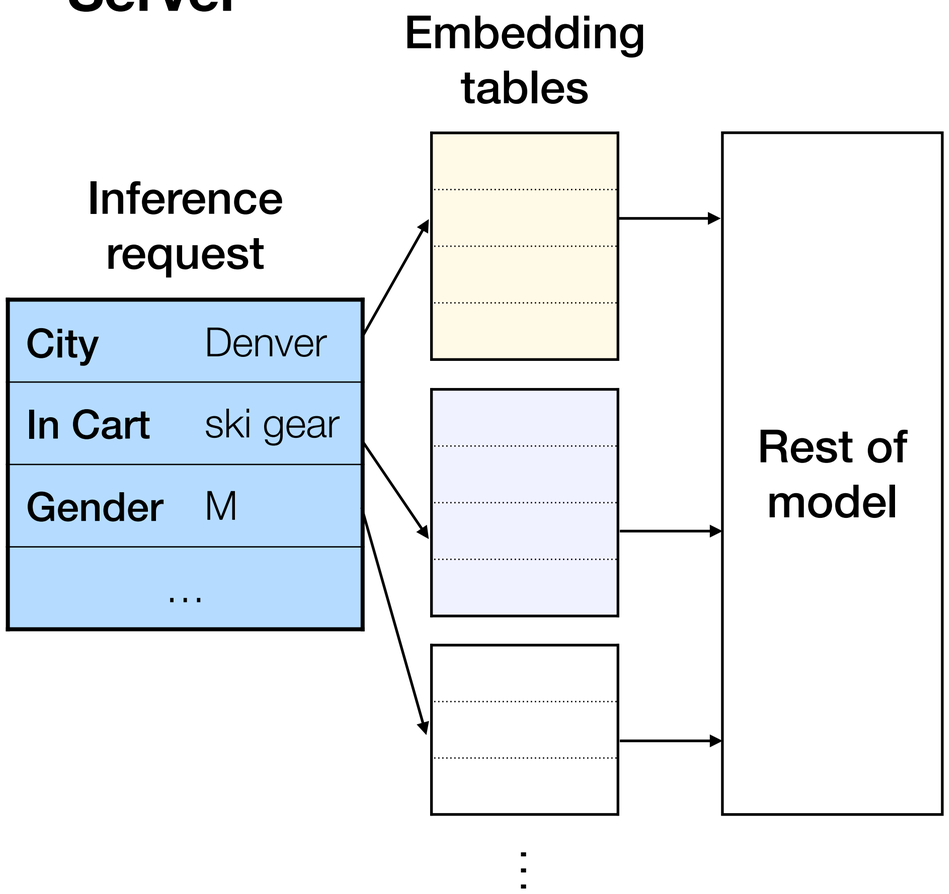
Server

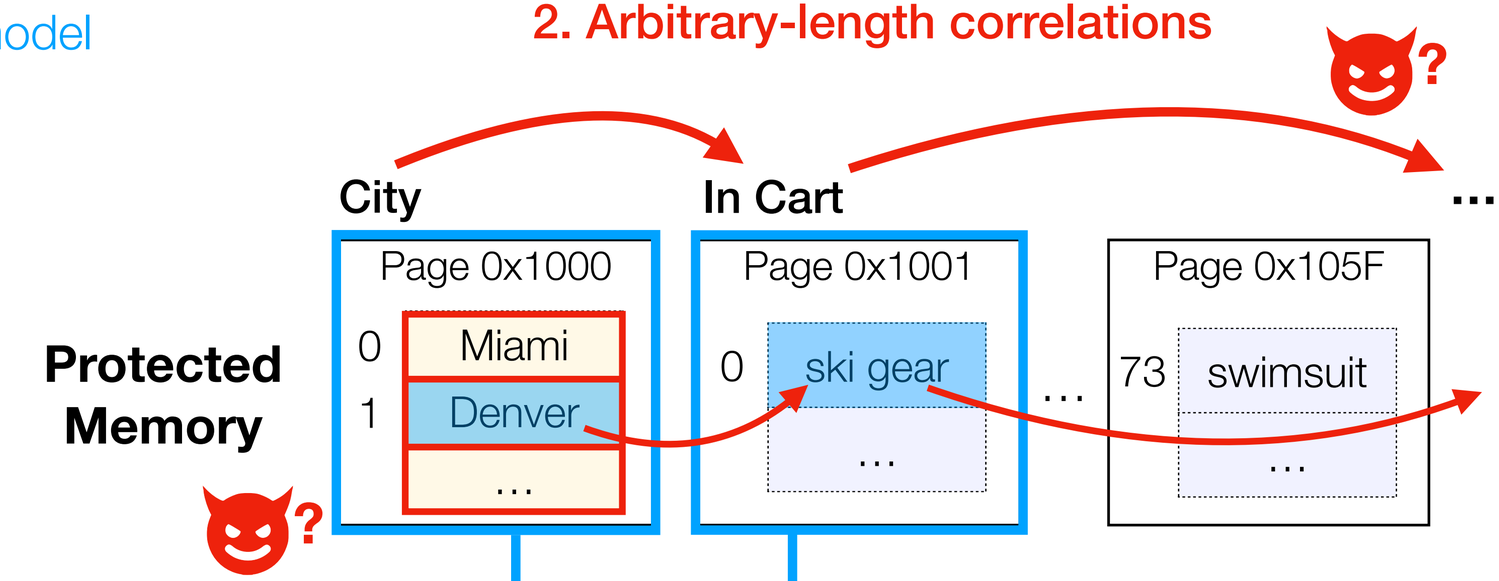


Deep learning recommendation model (DLRM)

Challenges: 1. Multiple objects per page

2. Arbitrary-length correlations





Page table

VPN	0x1000	0x1001	…	0x105F	…
Accessed?	1	1	…	0	…

Example: Ad recommendation engine

Deep learning recommendation model (DLRM)

Challenges: 1. Multiple objects per page
2. Arbitrary-length correlations

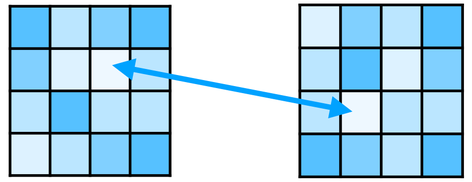
In the face of such challenges, is there a practical attack that can infer object-level accesses?

Inferring object accesses from pages

Prior attack approaches in related fields (e.g., SSE)

Markov modeling

IHOP [Sec'22]



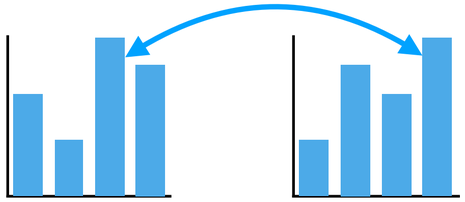
Known distribution

Observed distribution

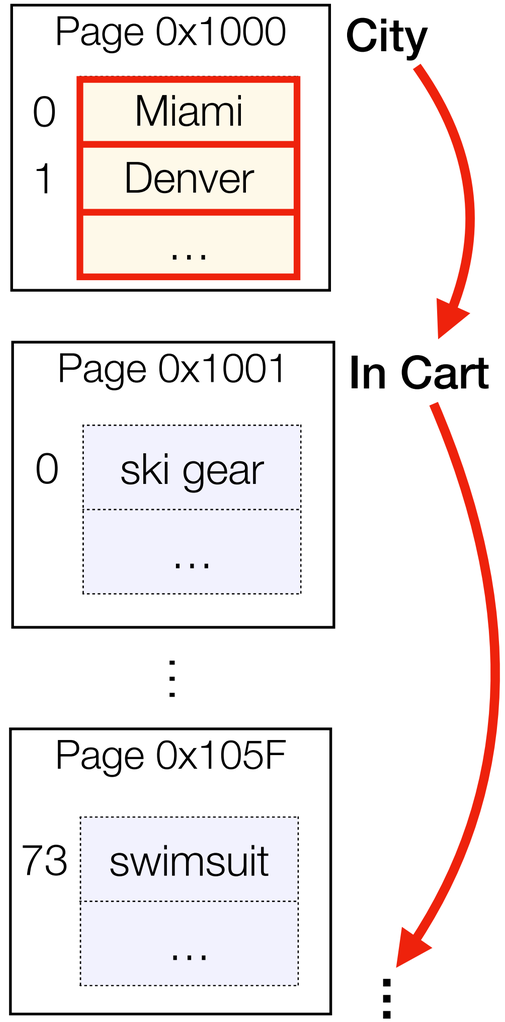
Assumes one object per page

Frequency analysis

Naive Bayes



Assumes independent accesses



Inferring object accesses from pages

Prior attack approaches in related fields (e.g., SSE)

Markov modeling

IHOP [Sec'22]

Assumes one object per page

Frequency analysis

Naive Bayes

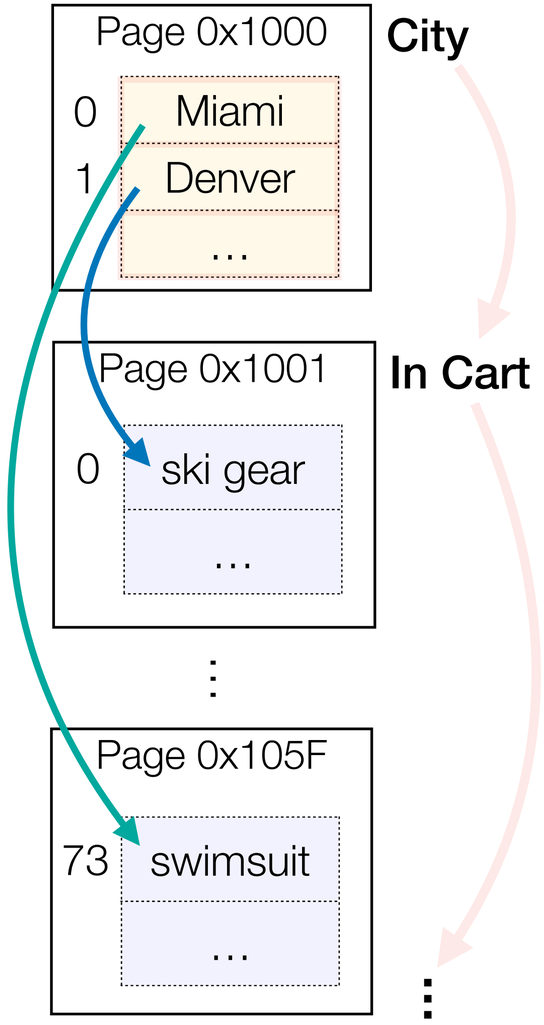
Assumes independent accesses

Our insight: Leverage correlations between objects

Adversary (OS) knows memory layout and access correlations

0x1000, 0x1001 ⟹ Denver, ski gear

0x1000, 0x105F ⟹ Miami, swimsuit



Inferring object accesses from pages

Prior attack approaches in related fields (e.g., SSE)

Markov modeling

IHOP [Sec'22]

Assumes one object per page

Frequency analysis

Naive Bayes

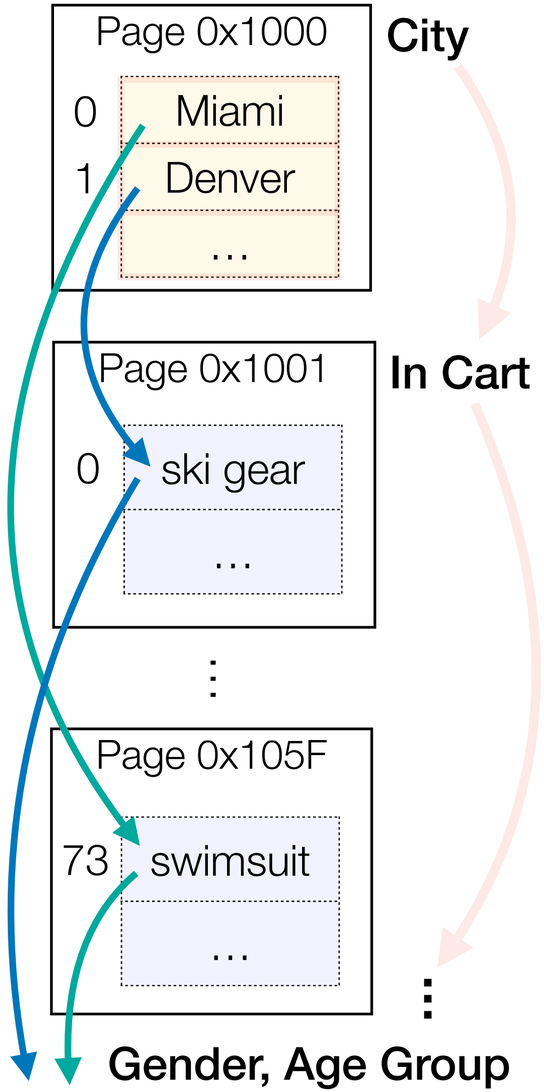
Assumes independent accesses

Our insight: Leverage correlations between objects

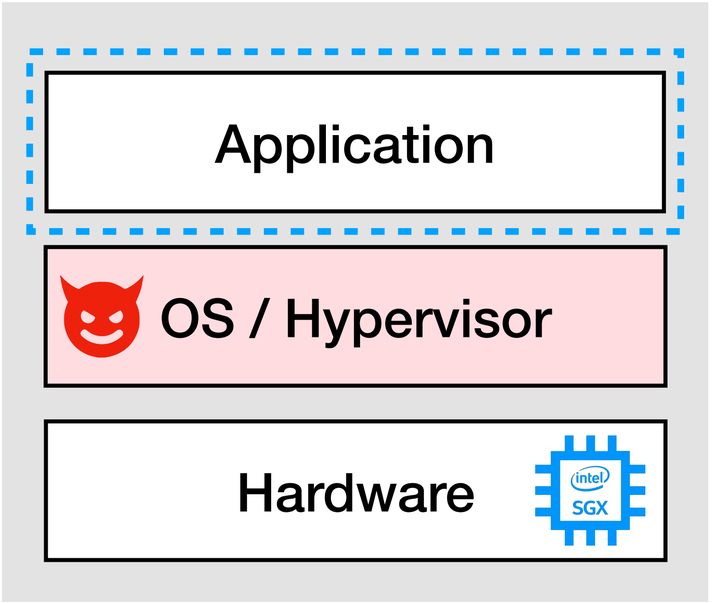
Adversary (OS) knows memory layout and access correlations

0x1000, 0x1001 ⟹ Denver, ski gear

0x1000, 0x105F ⟹ Miami, swimsuit



Threat Model



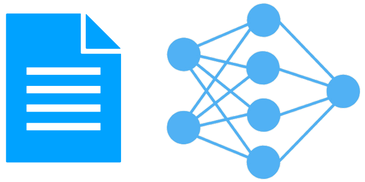
Honest-but-curious cloud adversary

- Cannot access application data or modify inputs
- Can observe application's page accesses
- Can control scheduling, memory allocations, etc.

Goal: Given page accesses, infer object accesses

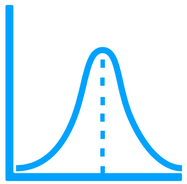
Auxiliary knowledge:

White-box access to application



Layout of objects per page

Distribution of input data



Language modeling attack approach

$$\Pr[\text{book} \mid \text{I, wrote, a}] > \Pr[\text{sandwich} \mid \text{I, wrote, a}]$$

Parallels in syntax & semantic rules

$$\Pr[\text{Miami} \mid \text{swimsuit}] > \Pr[\text{Denver} \mid \text{swimsuit}]$$

Language modeling attack approach

$\Pr[\text{book} \mid \text{I, wrote, a}]$
$> \Pr[\text{sandwich} \mid \text{I, wrote, a}]$

Parallels in syntax & semantic rules

$\Pr[\text{Miami} \mid \text{swimsuit}]$
$> \Pr[\text{Denver} \mid \text{swimsuit}]$

Translate **from language of page accesses to object accesses**

Learn distribution of object sequence **y** conditioned on page sequence **x**:

$$\Pr[y_1, \ldots, y_n \mid \mathbf{x}] = \prod_{i=1}^{n} \Pr[y_i \mid y_1, \ldots, y_{i-1}, \mathbf{x}]$$

Language modeling attack approach

$\Pr[\text{book} \mid \text{I, wrote, a}]$
$> \Pr[\text{sandwich} \mid \text{I, wrote, a}]$

Parallels in syntax & semantic rules

$\Pr[\text{Miami} \mid \text{swimsuit}]$
$> \Pr[\text{Denver} \mid \text{swimsuit}]$

Translate **from language of page accesses to object accesses**

Learn distribution of object sequence **y** conditioned on page sequence **x**:

$$\Pr[y_1, \ldots, y_n \mid \mathbf{x}] = \prod_{i=1}^{n} \Pr[y_i \mid y_1, \ldots, y_{i-1}, \mathbf{x}]$$

right hand $\Longrightarrow$ main **droite**

that is **right** $\Longrightarrow$ c'est **vrai**

Disambiguation via arbitrary-length context

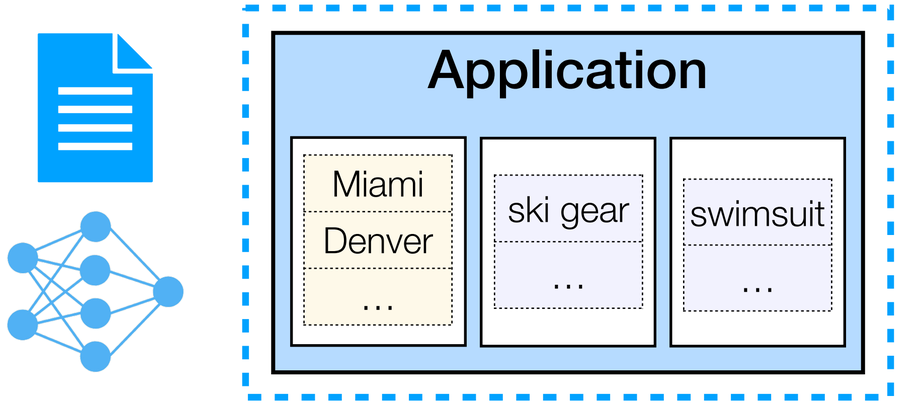
0x1000, 0x1001 $\Longrightarrow$ **Denver**, ski gear

0x1000, 0x105F $\Longrightarrow$ **Miami**, swimsuit

Found in Translation (FIT) Attack

Initialization

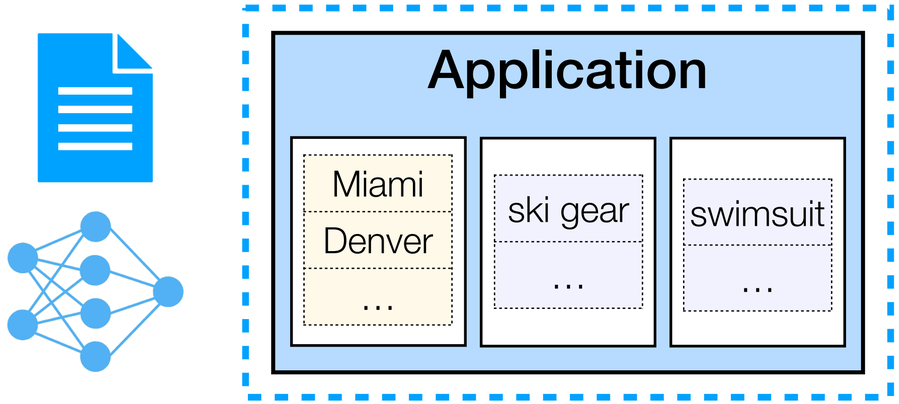
Set up monitored page table entries



Found in Translation (FIT) Attack

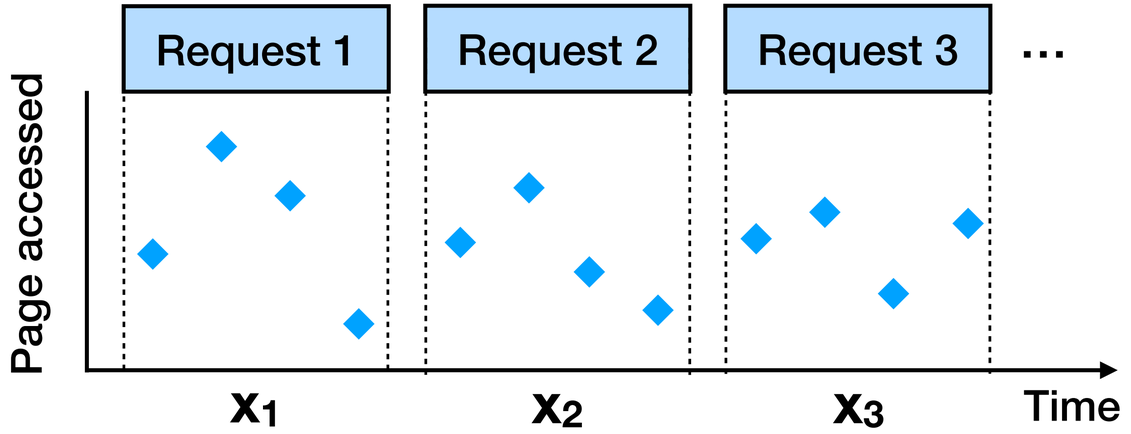
Initialization

Set up monitored page table entries



Online collection

Track page access sequences **x** via accessed bits



Found in Translation (FIT) Attack

Initialization

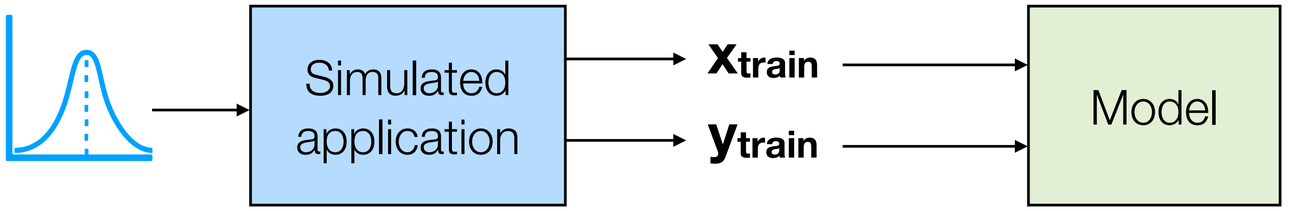
Set up monitored page table entries

Online collection

Track page access sequences **x** via accessed bits

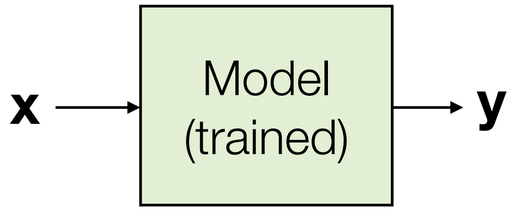
Offline analysis

1. Generate training data



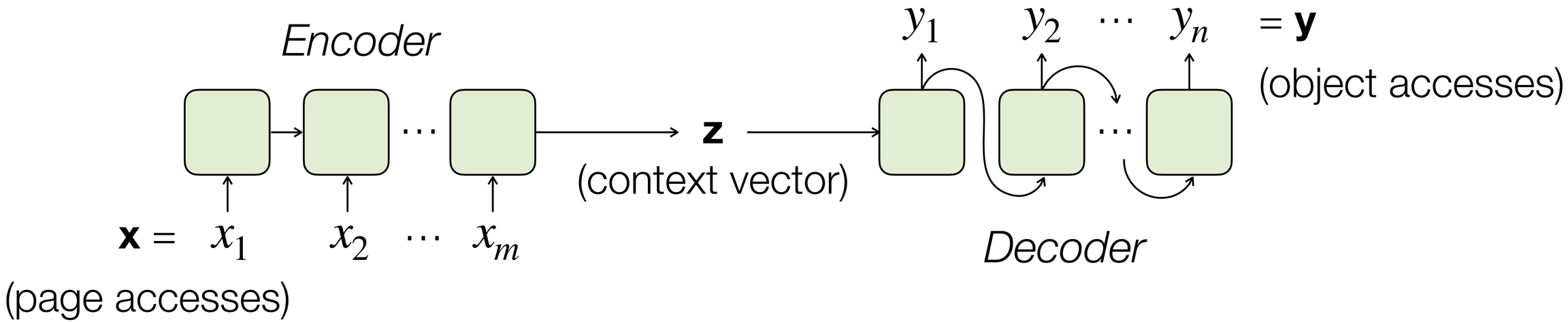
2. Training

3. Inference



FiT Model Details

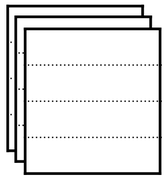
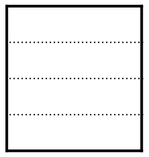
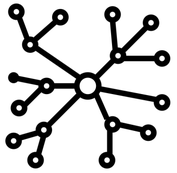
Deep generative language modeling with *recurrent encoder-decoder*



Use BERT as encoder and decoder

- Transformer processes sequences in parallel, learning global dependencies
- Leverage pre-trained weights for language prediction

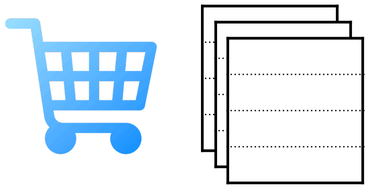
Evaluated Applications

	Privacy-sensitive use case	Access pattern	Objects	Dataset, train/test split
DLRM	User features for ad predictions	Per-feature lookup 	Embedding table entries	Criteo ad data, 1M / 100K
Medical LLM	Patient symptoms for diagnosis	Per-token lookup 	Embedding table entries	DDXPlus medical data, 500K / 50K
HNSW for semantic search	Biometrics, face images, etc.	Graph index traversal 	Nodes in graph	SIFT image search, 22.5K / 2.6K

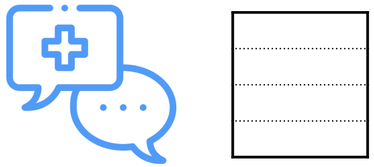
Evaluation Setup

Targeted applications

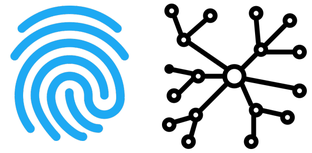
DLRM



LLM



HNSW



Executed in Nitro Enclave & Intel SGX Enclave (Ice Lake)

Compared attacks

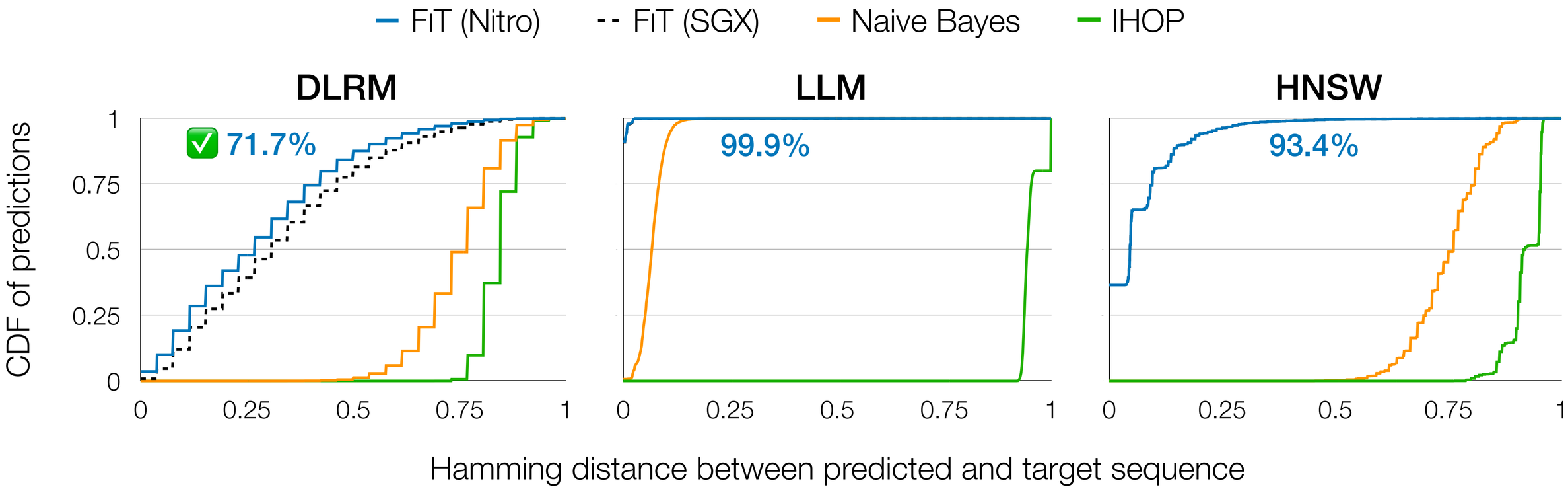
FiT

NVIDIA A40 GPUs

Naive Bayes **IHOP**

AMD EPYC 7302P CPUs

Attack Efficacy

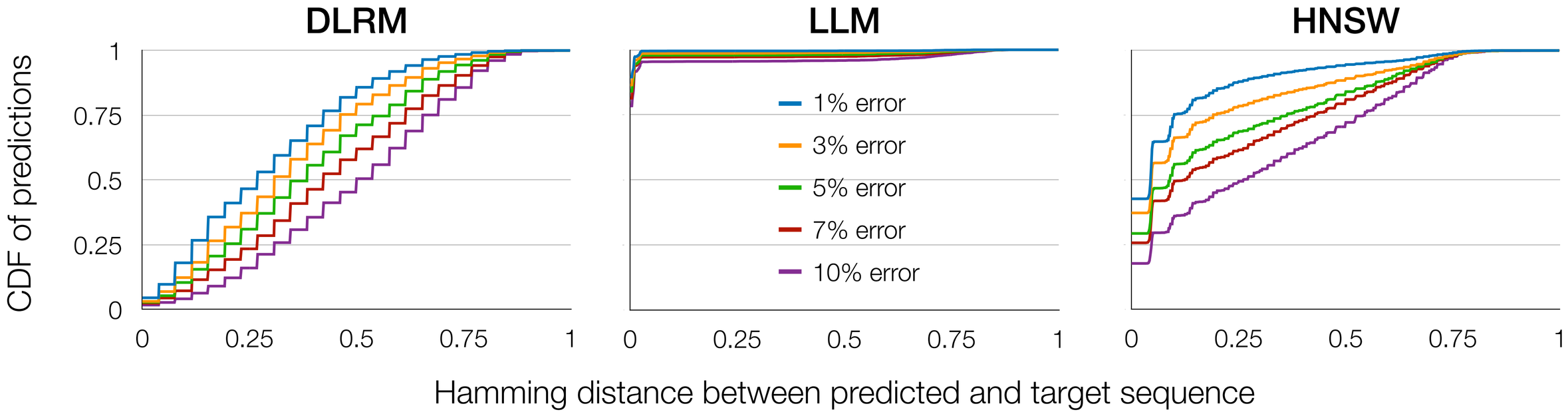


FiT achieves high accuracy by learning (i) many-to-1 object-to-page mappings (ii) arbitrary-length correlations

FIT Sensitivity to Error

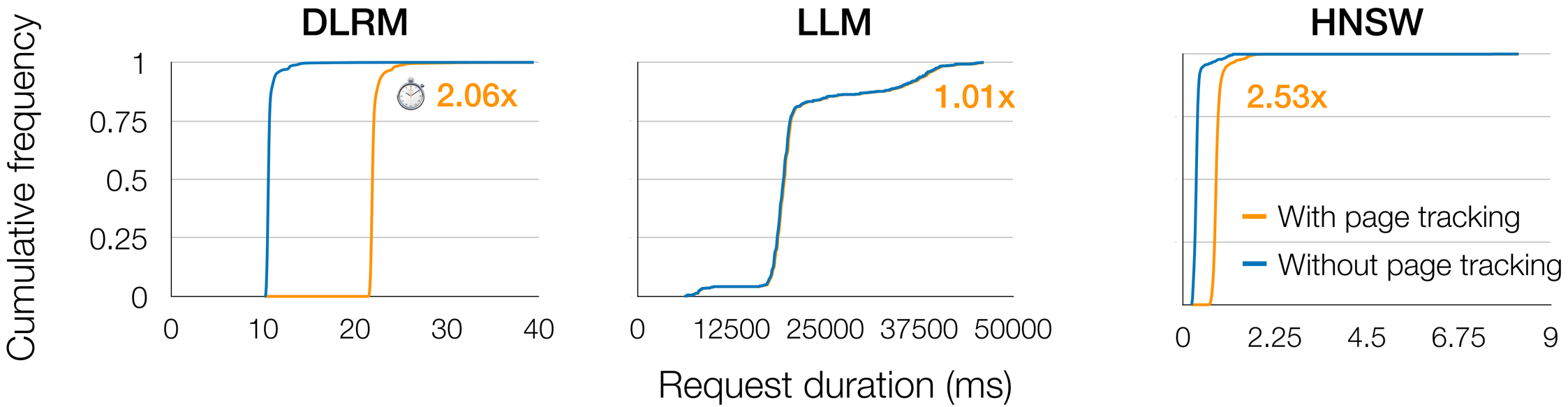
~1% error rate in observed page accesses

- False positives: prefetching; accesses to co-located metadata
- False negatives: background processes clearing accessed bit



Context awareness makes FIT robust against errors in collected page sequences

FiT Latency Overheads



Online **page tracking adds low latency — masked by network, tiered memory, etc.**

Offline **analysis time depends on application (1—45 GPU hours)**

Conclusion

Found in Translation

- Leverages *language modeling* to capture both
  - (i) many-to-1 object-page mappings
  - (ii) arbitrary-length correlations
- Practical and achieves high accuracy

Future Work

- Adaptation of attack to other settings
- Sensitivity study for page sizes, memory tiering, etc.
- Correlation-hiding countermeasures

Paper
usenix.org/conference/usenixsecurity25/presentation/jia-grace

Code
github.com/yale-nova/found-in-translation