Spot
by NetApp

CISPA
HELMHOLTZ CENTER FOR
INFORMATION SECURITY

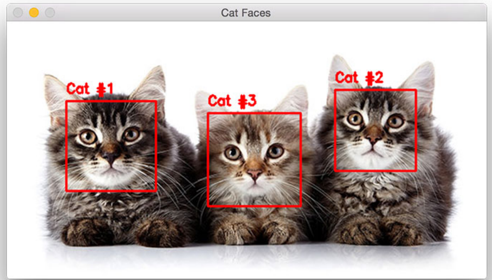
All Your GNN Models And Data Belong To Me*

Yang Zhang and his research group (CISPA Helmholtz Center for Information Security)
Yun Shen (Spot by NetApp)
Azzedine Benameur (Spot by NetApp)

*All attacks discussed in this talk are simulated in the lab environment.

The Age of Machine Learning

GitHub Copilot



Image/Text/Video/Audio

Image source (from left to right): https://github.com/deepmind/alphafold https://emergetech.org/openai-gpt3-good-at-almost-everything/ https://github.com/features/copilot/signup https://imagen.research.google/

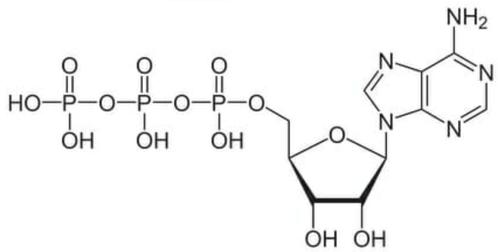
Graph

Graphs Are Everywhere

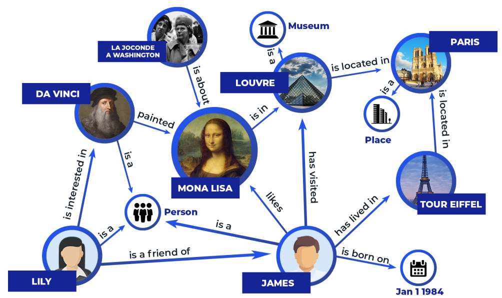
Graphs are combinatorial structures, have arbitrary sizes, and contain multi-modal information

Social Networks

Molecules



Knowledge Graphs



User-Item Graphs

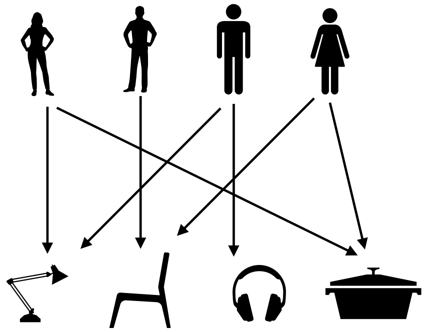
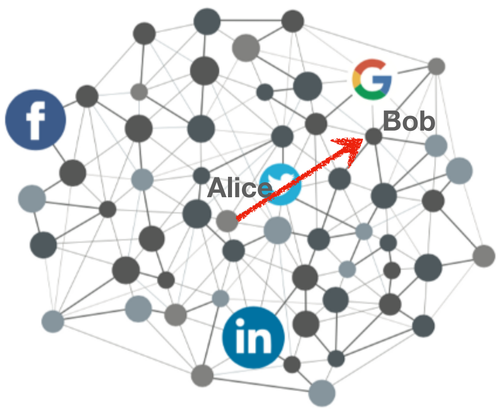


Image source: (from left to right): https://towardsdatascience.com/ab-testing-challenges-in-social-networks-e67611c92916, https://biologydictionary.net/molecule/, https://yashuseth.blog/2019/10/08/introduction-question-answering-knowledge-graphs-kgqa/

Graph Applications Are Everywhere

Graph-based applications pervasively exist in our everyday life

Social Networks



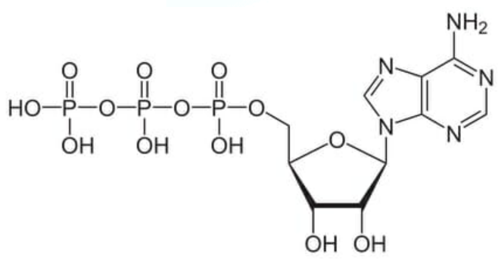
Demographic Inference

Age group of Bob

Link Prediction

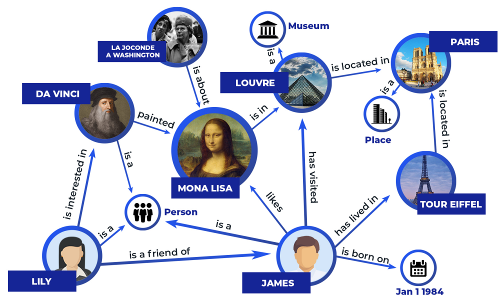
Do you (Alice) know Bob?

Molecules



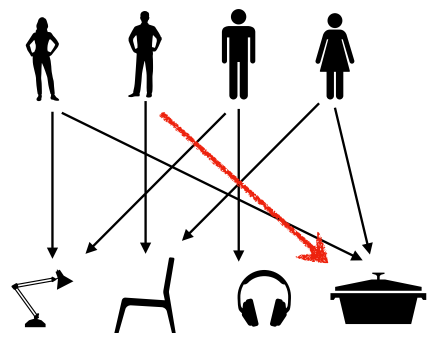
Toxicity Prediction

Knowledge Graphs



Knowledge Mining

User-Item Graphs



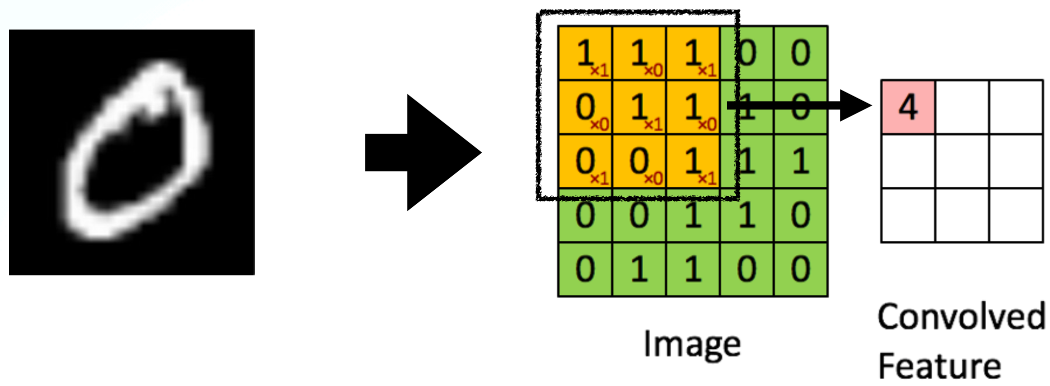
Recommendation

We found an item you may be interested!

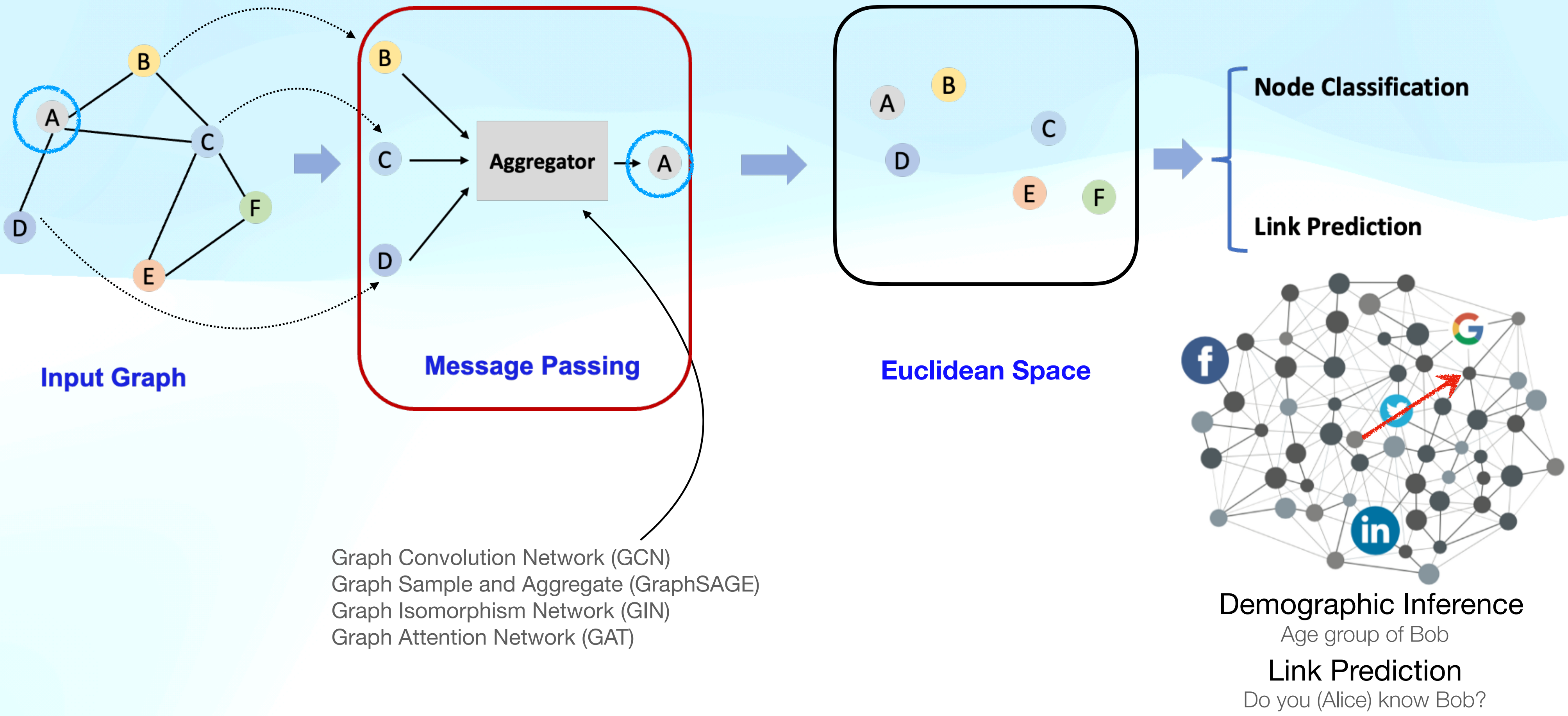
Image source: (from left to right): https://towardsdatascience.com/ab-testing-challenges-in-social-networks-e67611c92916, https://biologydictionary.net/molecule/, https://yashuseth.blog/2019/10/08/introduction-question-answering-knowledge-graphs-kgqa/

Graph Neural Netwok (GNN)

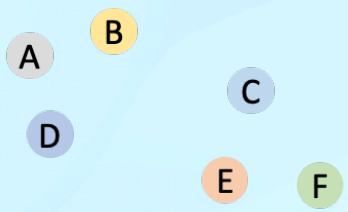
- Traditional neural networks are designed for grids (e.g., images) or sequences (e.g., text)
  - CNNs for images
  - RNNs for sequences



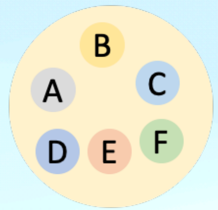
Graph Neural Netwok (GNN)



Graph Neural Netwok (GNN)



- **Mean pooling**
- **Max pooling**



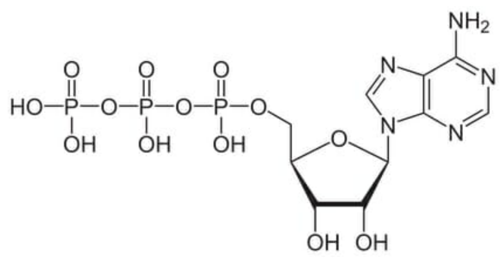
Graph Classification

Graph Matching

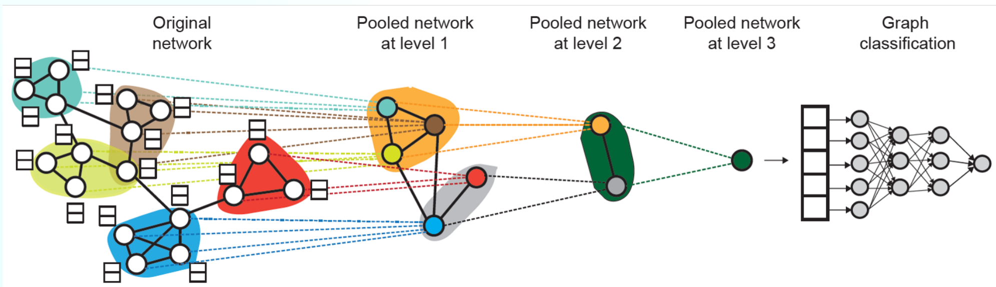
Graph Visualization

Node Embeddings

Graph Embedding

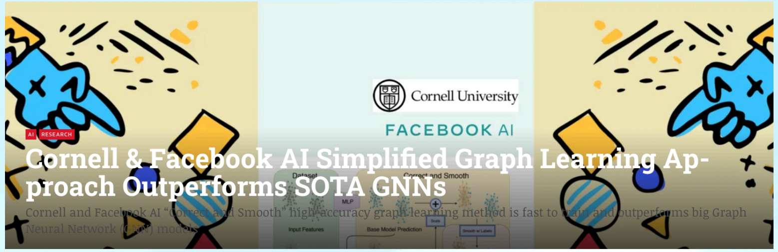


Toxicity Prediction



[1] Hierarchical Graph Representation Learning with Differentiable Pooling. Ying et. al.

Graph Neural Netwok (GNN)



Insights

Graph ML at Twitter

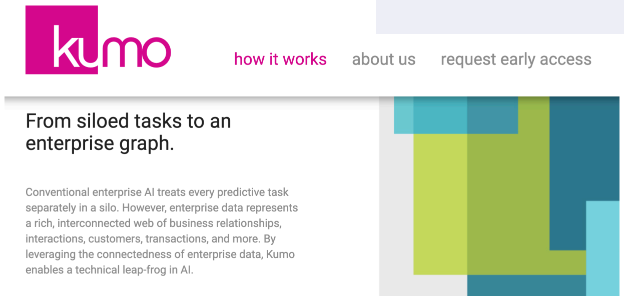
By Michael Bronstein

Wednesday, 2 September 2020

Introducing Amazon SageMaker Support for Deep Graph Library (DGL): Build and Train Graph Neural Networks

Posted On: Dec 3, 2019

Amazon SageMaker support for the Deep Graph Library (DGL) is now available. With DGL, you can improve the prediction accuracy of recommendation, fraud detection, and drug discovery systems using Graph Neural Networks (GNNs).



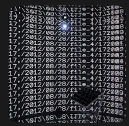
Graph

The Age of Machine Learning

Bloomberg

AI Poisoning Is the Next Big Risk in Cybersecurity

25 Apr • Opinion



IEEE Spectrum

How Adversarial Attacks Could Destabilize Military AI Systems

The Age of Adversarial Machine Learning

WIRED

Even Artificial Neural Networks Can Have Exploitable 'Backdoors'

Air Force Magazine

Does AI Present a New Attack Surface for Adversaries?

29 Sept 2021

Overview*

	Graph	GNN
Security		Model extraction attack
Privacy	Link re-identification attack Property inference attack Subgraph inference attack	

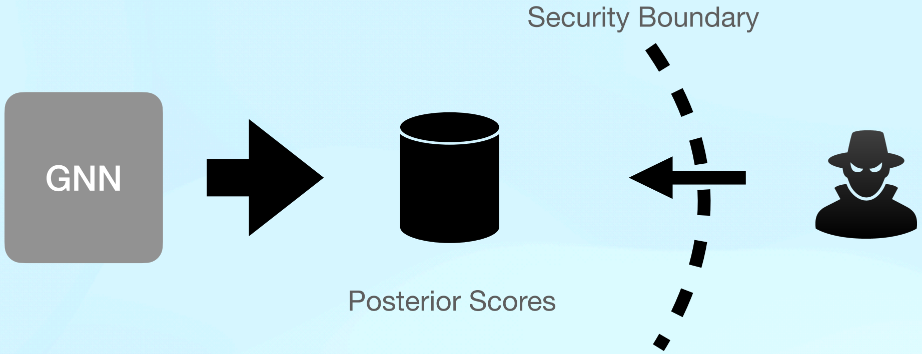
*All attacks discussed in this talk are simulated in the lab environment.

Link Re-Identification Attack

	Graph	GNN
Security		
Privacy	**Link re-identification attack**	Identify if two nodes are connected in the **training data**

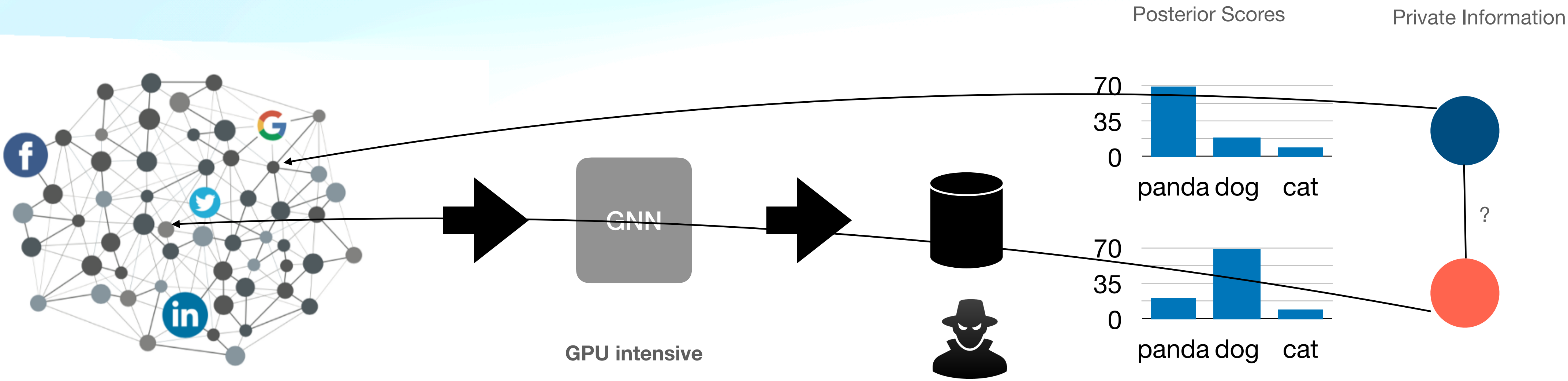
Link Re-Identification Attack (Scenario 1)

Scenario



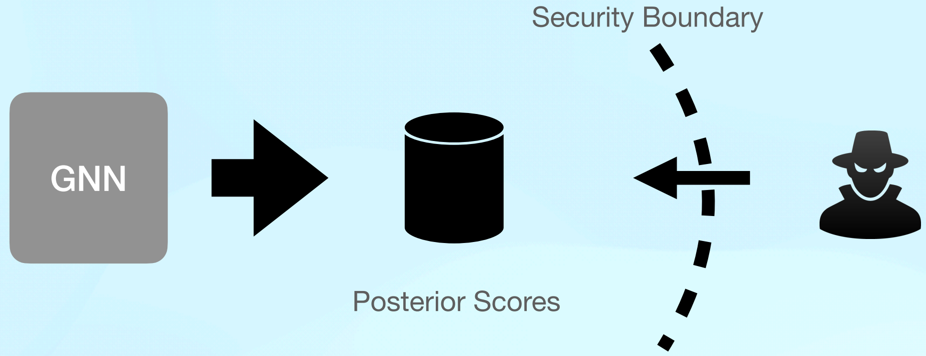
GNN model:
Node classification

Attacker's **capability**:
1. posteriors of nodes (from training data) obtained from the target model



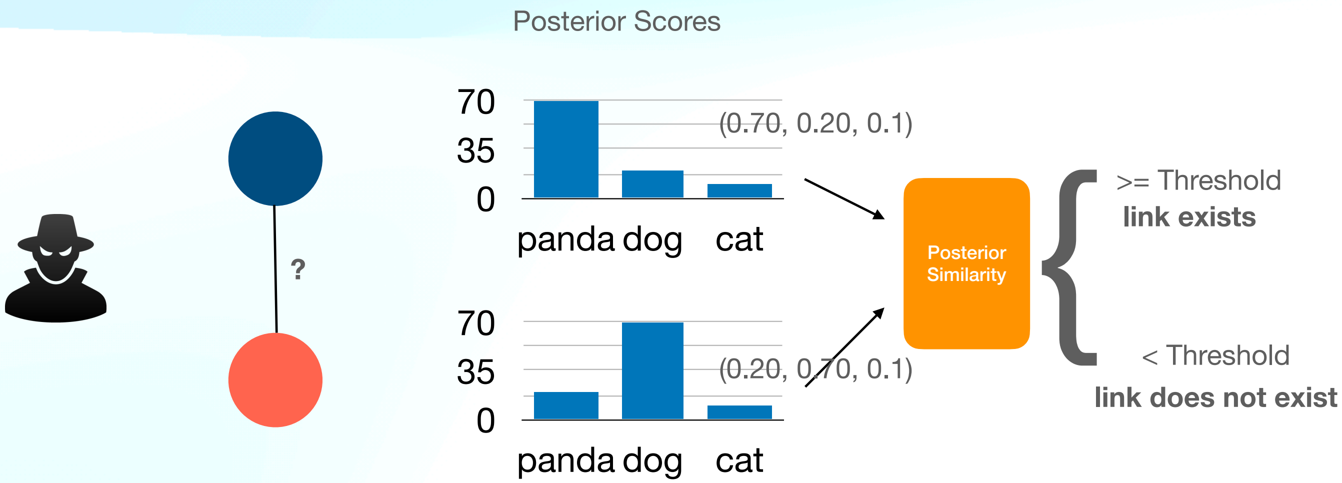
Link Re-Identification Attack (Scenario 1)

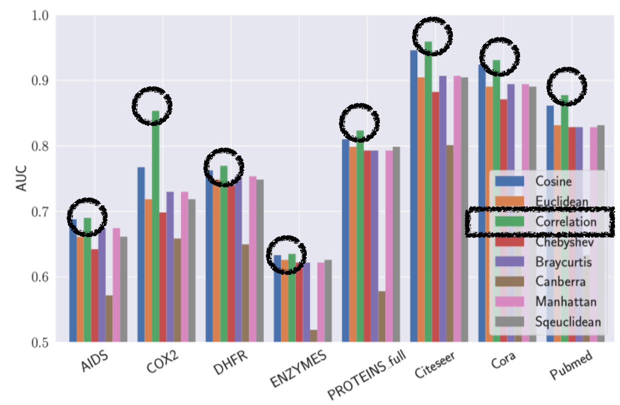
Scenario



GNN model:
Node classification

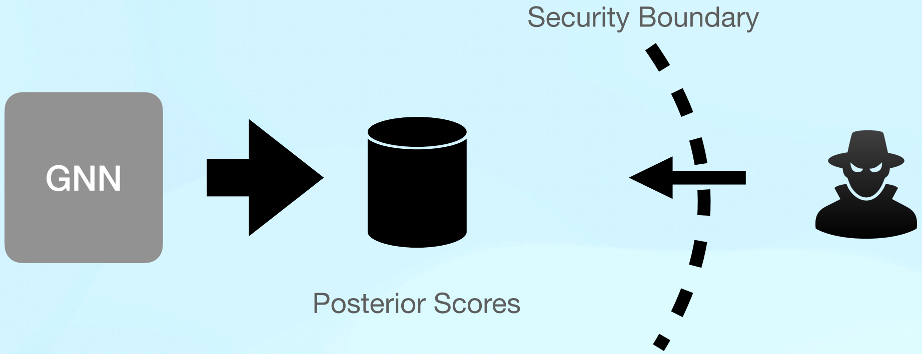
Attacker's **capability**:
1. posteriors of nodes (from training data) obtained from the target model





Link Re-Identification Attack (Scenario 2)

Scenario



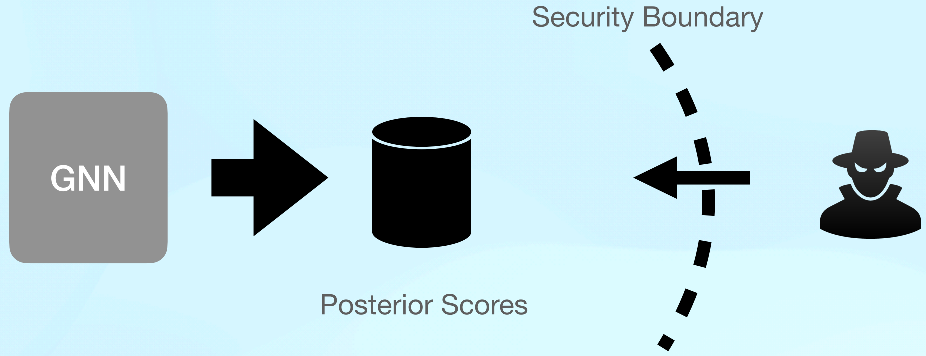
GNN model:
Node classification

Attacker's **capability**:

1. posteriors of nodes (from training data) obtained from the target model
2. **have a shadow dataset**

Link Re-Identification Attack (Scenario 2)

Scenario

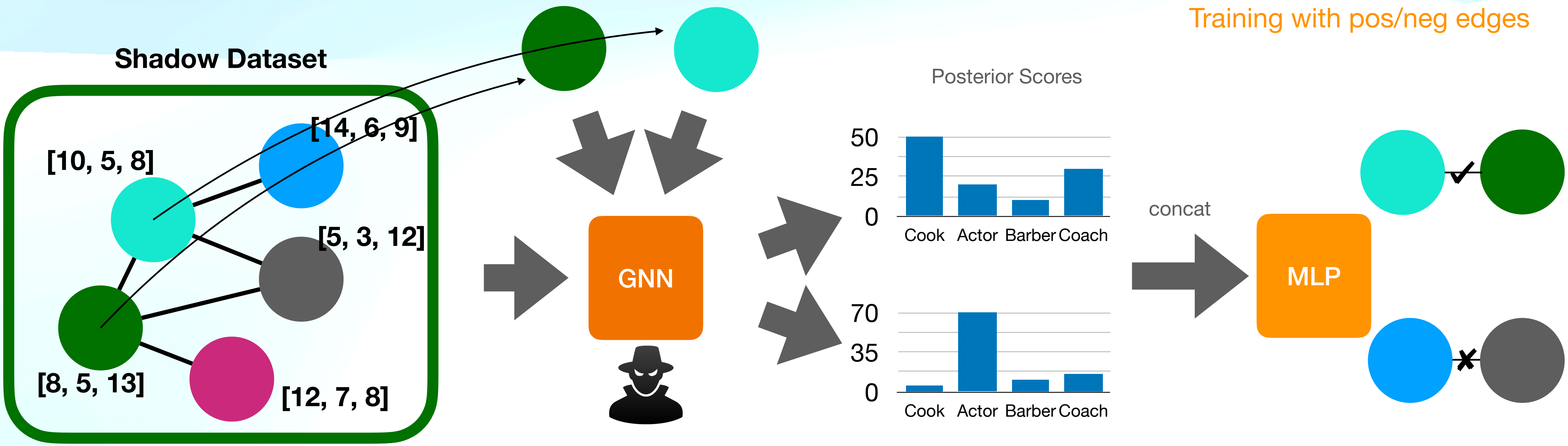


GNN model:
Node classification

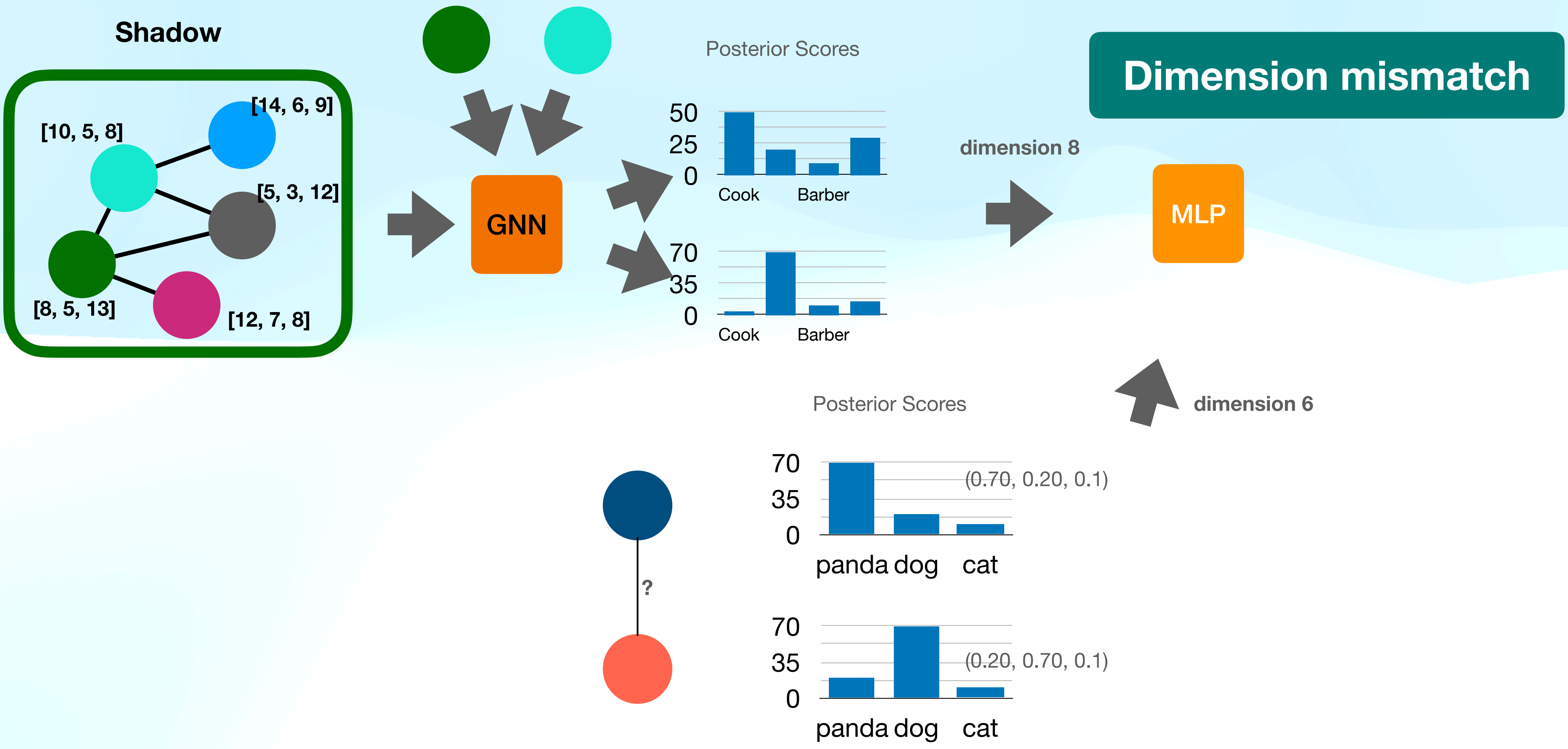
Attacker's **capability**:

1. posteriors of nodes (from training data) obtained from the target model
2. **have a shadow dataset**

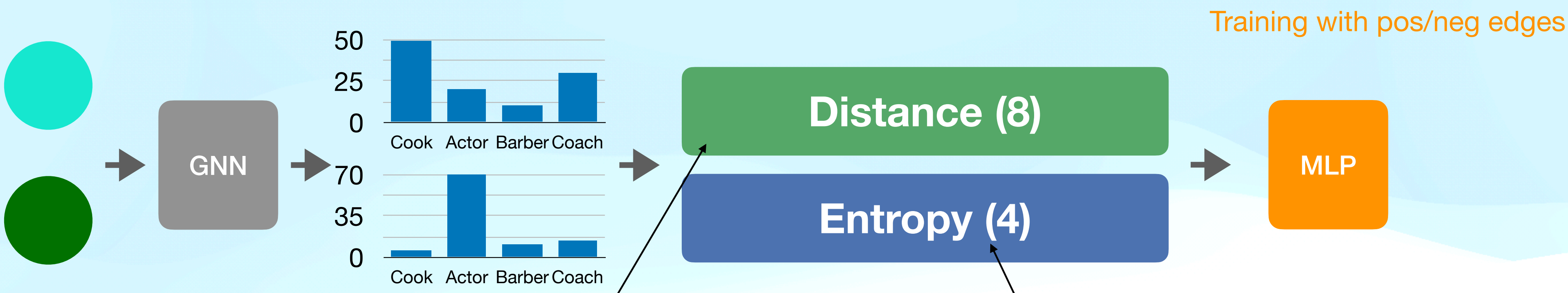
Training with pos/neg edges



Link Re-Identification Attack (Scenario 2)



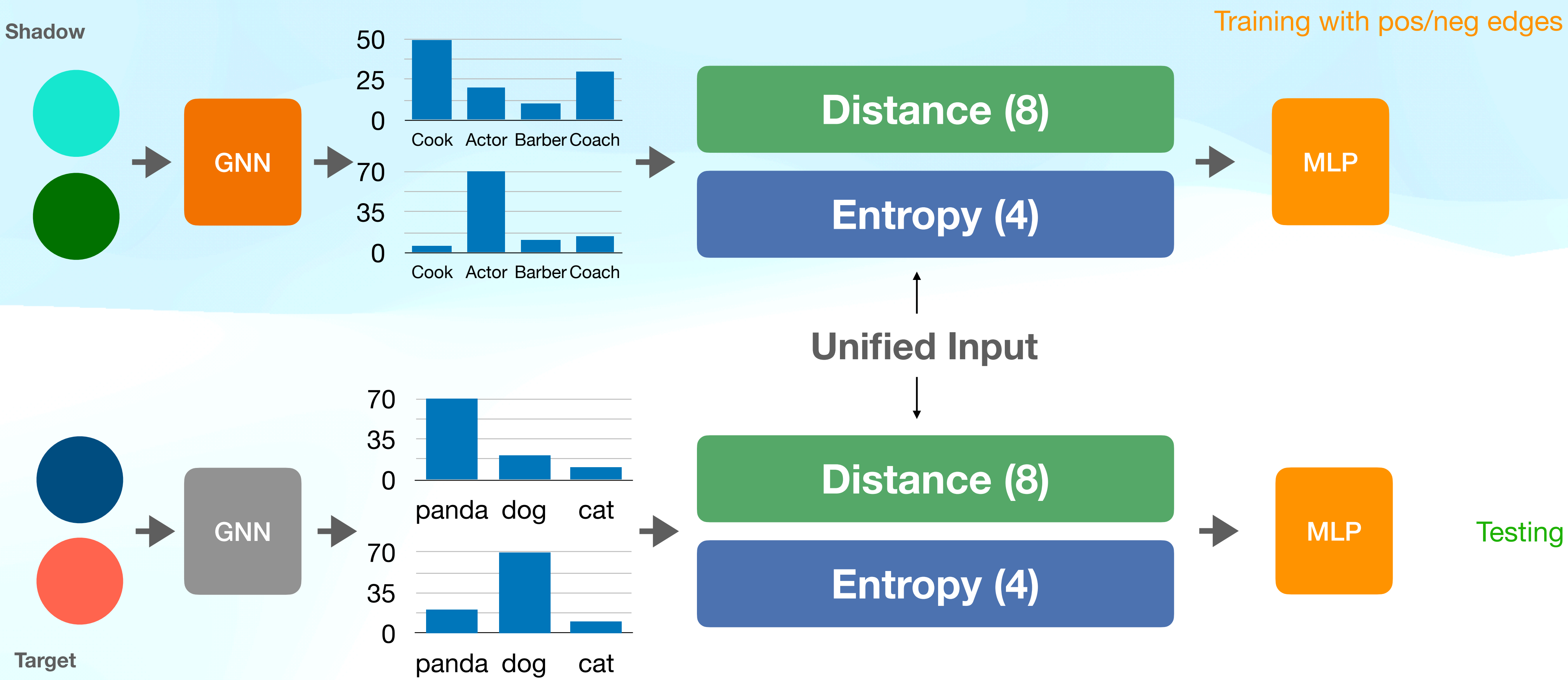
Link Re-Identification Attack (Scenario 2)



Metrics	Definition
Cosine	$1-\frac{f(u)\cdot f(v)}{\|f(u)\|_2\|f(v)\|_2}$
Euclidean	$\|f(u)-f(v)\|_2$
Correlation	$1-\frac{(f(u)-\overline{f(u)})\cdot(f(v)-\overline{f(v)})}{\|(f(u)-\overline{f(u)})\|_2\|(f(v)-\overline{f(v)})\|_2}$
Chebyshev	$\max_i \lvert f_i(u)-f_i(v)\rvert$
Braycurtis	$\frac{\sum\lvert f_i(u)-f_i(v)\rvert}{\sum\lvert f_i(u)+f_i(v)\rvert}$
Manhattan	$\sum_i \lvert f_i(u)-f_i(v)\rvert$
Canberra	$\sum_i \frac{\lvert f_i(u)-f_i(v)\rvert}{\lvert f_i(u)\rvert+\lvert f_i(v)\rvert}$
Sqeuclidean	$\|f(u)-f(v)\|_2^2$

Operator	Definition	Operator	Definition
Average	$\frac{f_i(u)+f_i(v)}{2}$	Weighted-L1	$\lvert f_i(u)-f_i(v)\rvert$
Hadamard	$f_i(u)\cdot f_i(v)$	Weighted-L2	$\lvert f_i(u)-f_i(v)\rvert^2$

Link Re-Identification Attack (Scenario 2)



Link Re-Identification Attack (Scenario 2)

AUC

Target Dataset	Shadow Dataset AIDS	COX2	DHFR	ENZYMES	PROTEINS_full	Citeseer	Cora	Pubmed
AIDS	-	0.720 ± 0.009	0.690 ± 0.005	**0.730 ± 0.010**	0.720 ± 0.005	0.689 ± 0.019	0.650 ± 0.025	0.667 ± 0.014
COX2	0.755 ± 0.032	-	0.831 ± 0.005	0.739 ± 0.116	**0.832 ± 0.009**	0.762 ± 0.009	0.773 ± 0.008	0.722 ± 0.024
DHFR	0.689 ± 0.004	**0.771 ± 0.004**	-	0.577 ± 0.044	0.701 ± 0.010	0.736 ± 0.005	0.740 ± 0.003	0.663 ± 0.010
ENZYMES	**0.747 ± 0.014**	0.695 ± 0.023	0.514 ± 0.041	-	0.691 ± 0.030	0.680 ± 0.012	0.663 ± 0.009	0.637 ± 0.018
PROTEINS_full	0.775 ± 0.020	0.821 ± 0.016	0.528 ± 0.038	0.822 ± 0.020	-	**0.823 ± 0.004**	0.809 ± 0.015	0.809 ± 0.013
Citeseer	0.801 ± 0.040	0.920 ± 0.006	0.842 ± 0.036	0.846 ± 0.042	0.848 ± 0.015	-	**0.965 ± 0.001**	0.942 ± 0.003
Cora	0.791 ± 0.019	0.884 ± 0.005	0.811 ± 0.024	0.804 ± 0.048	0.869 ± 0.012	**0.942 ± 0.001**	-	0.917 ± 0.002
Pubmed	0.705 ± 0.039	0.796 ± 0.007	0.704 ± 0.042	0.708 ± 0.067	0.752 ± 0.014	0.883 ± 0.006	**0.885 ± 0.005**	-

Property/Subgraph Inference Attack

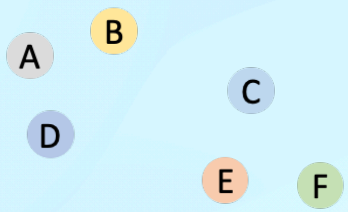
	Graph	GNN
Security		
Privacy	**Property inference attack** **Subgraph inference attack**	

Infer basic graph properties of a graph via its graph embedding

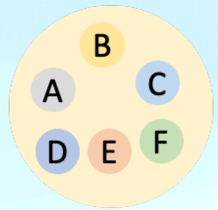
Infer if a certain subgraph exists in a graph via its graph embedding

*All attacks discussed in this talk are simulated in the lab environment.

Graph Neural Netwok (GNN)



- **Mean pooling**
- **Max pooling**



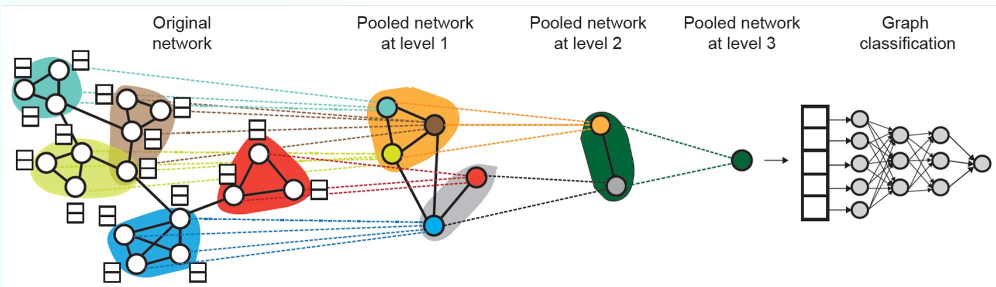
Graph Classification

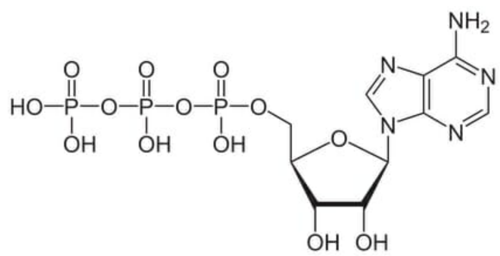
Graph Matching

Graph Visualization

Node Embeddings

Graph Embedding



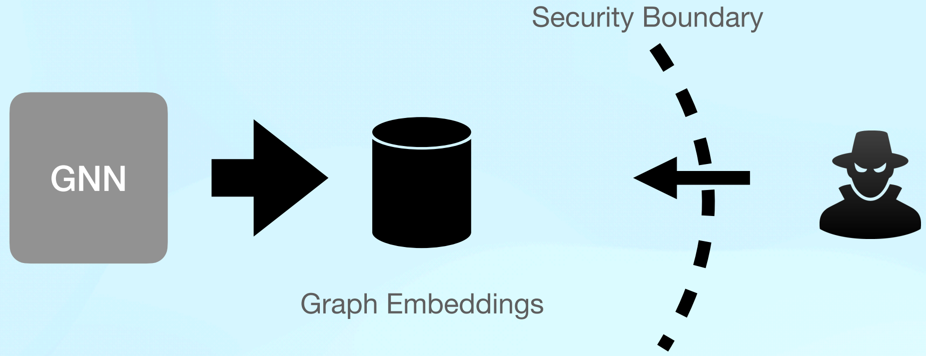


Toxicity Prediction

[1] Hierarchical Graph Representation Learning with Differentiable Pooling. Ying et. al.

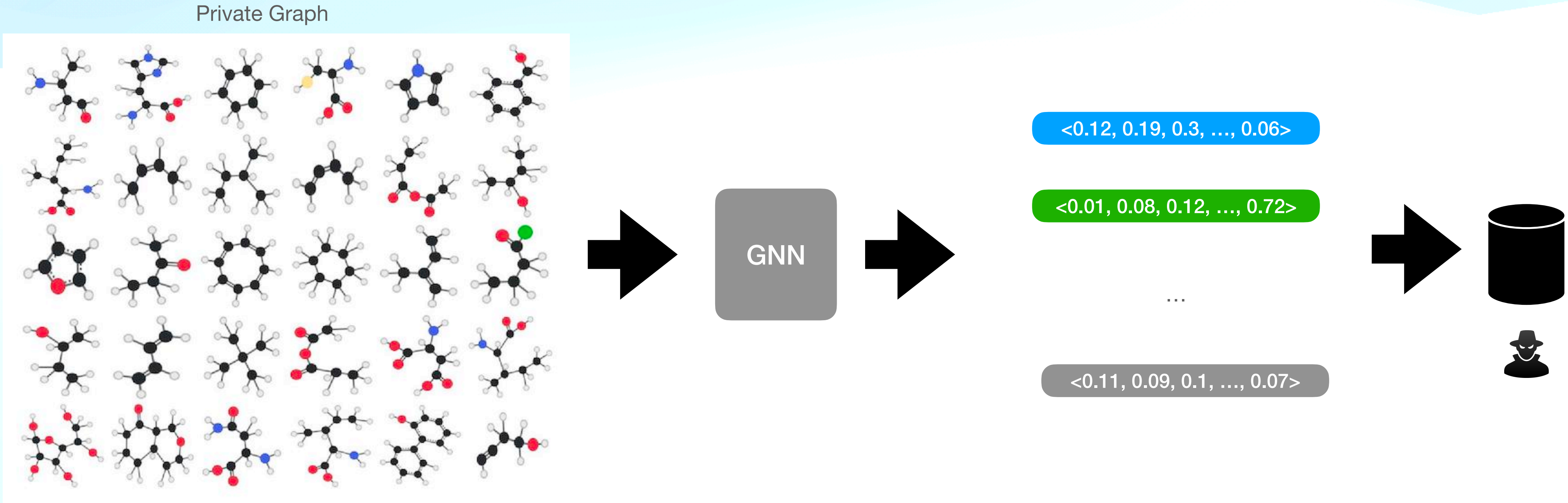
Property Inference Attack

Scenario



GNN model:
Graph classification

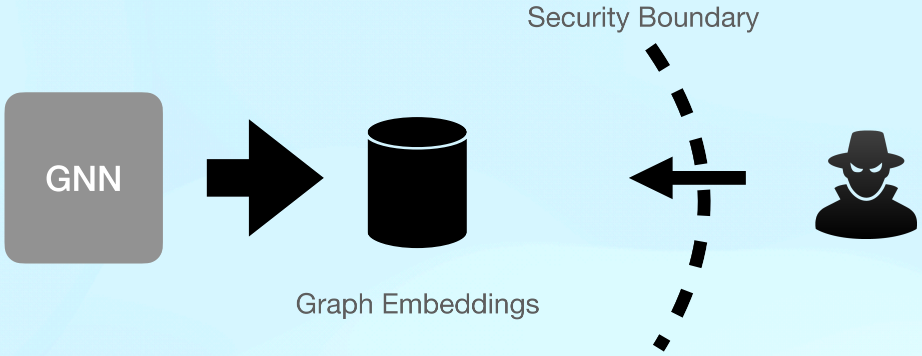
Attacker's **capability**:
1. Embeddings of graphs (from training data) obtained from the target model
2. Can query the GNN model



https://github.com/chemplexity/molecules

Property Inference Attack

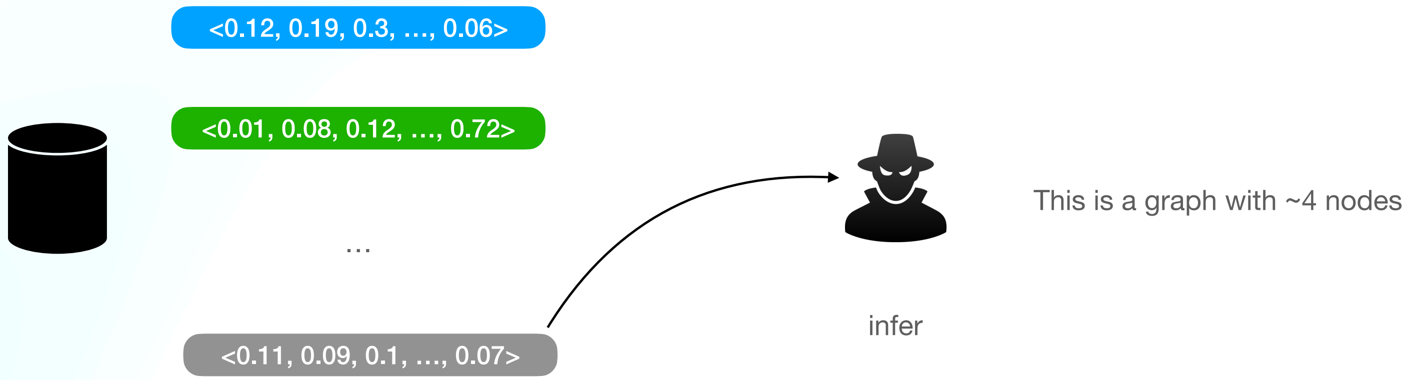
Scenario



GNN model:
Graph classification

Attacker's **capability**:

1. Embeddings of graphs (from training data) obtained from the target model
2. Can query the GNN model

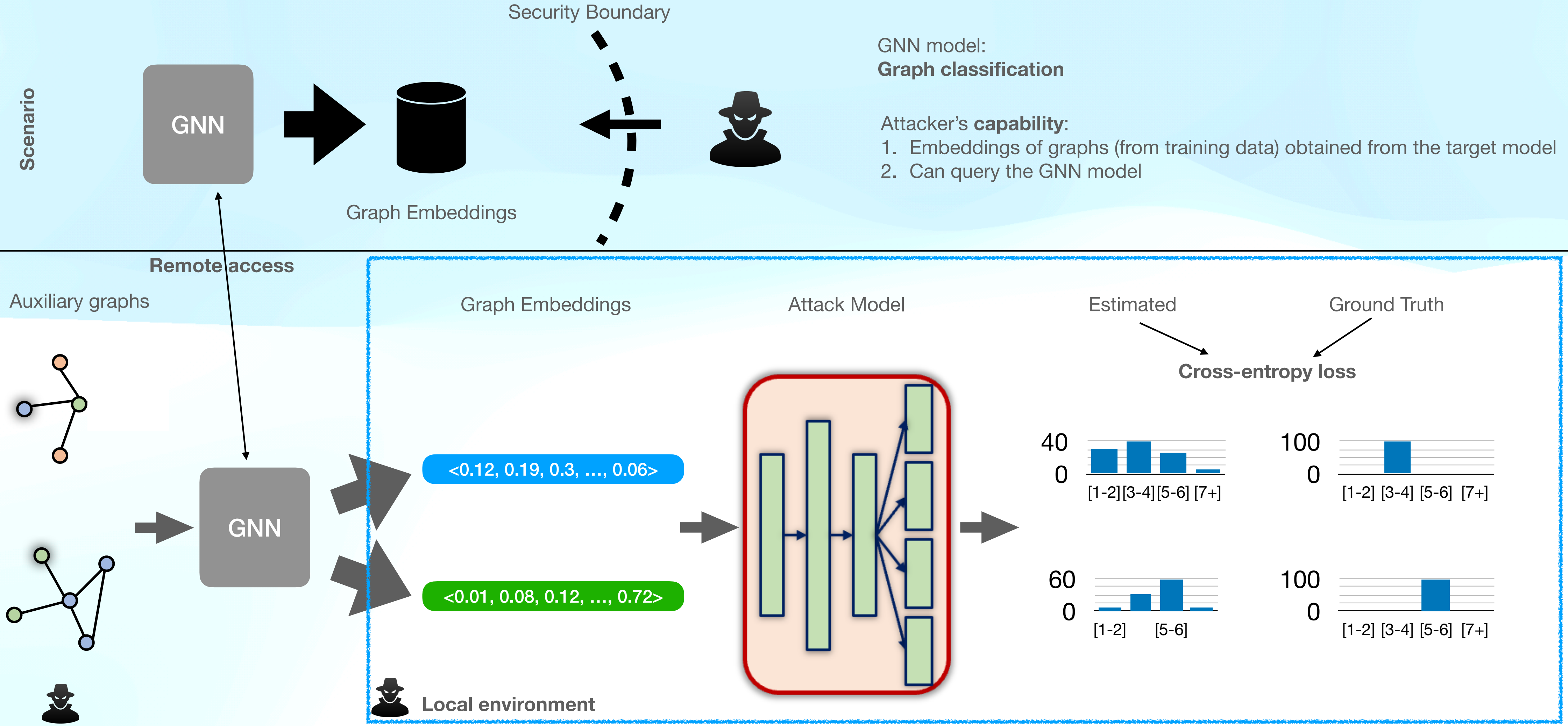


Property Inference Attack

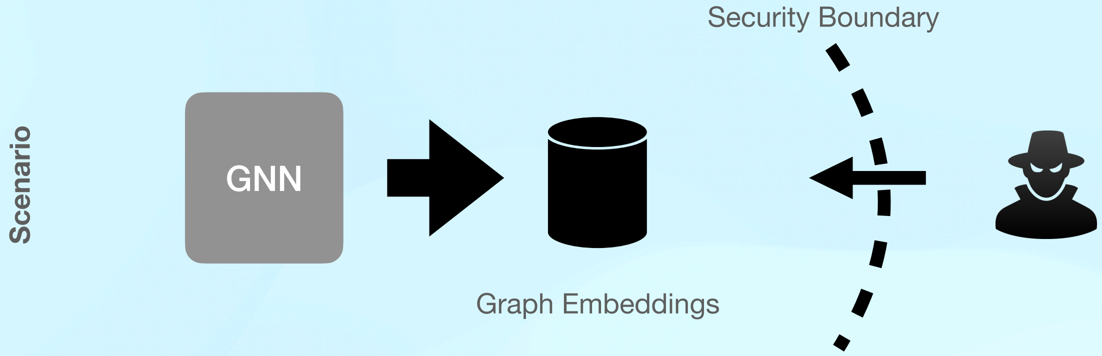
GNN model:
Graph classification

Attacker's **capability**:

1. Embeddings of graphs (from training data) obtained from the target model
2. Can query the GNN model



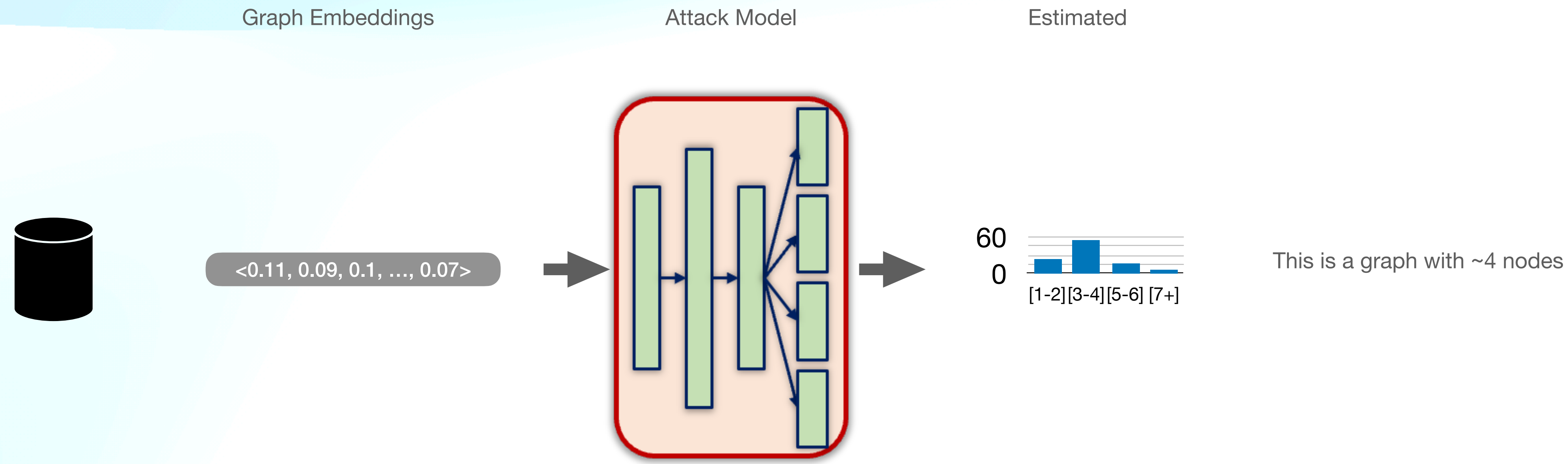
Property Inference Attack



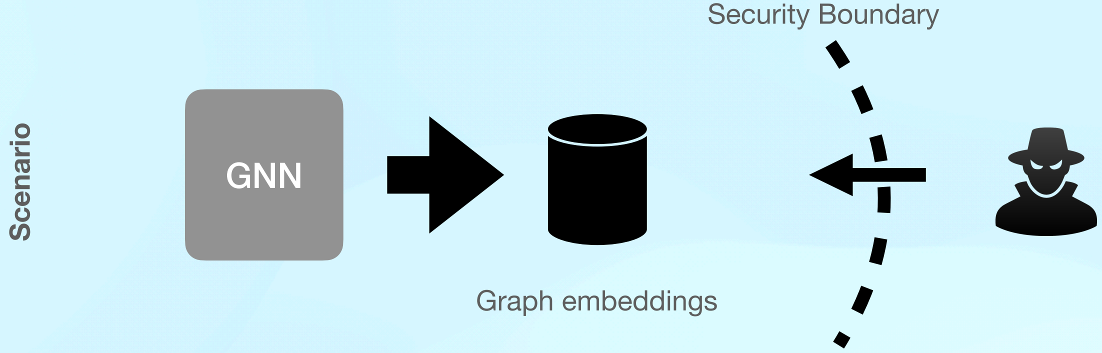
GNN model:
Graph classification

Attacker's **capability**:

1. Embeddings of graphs (from training data) obtained from the target model
2. Can query the GNN model



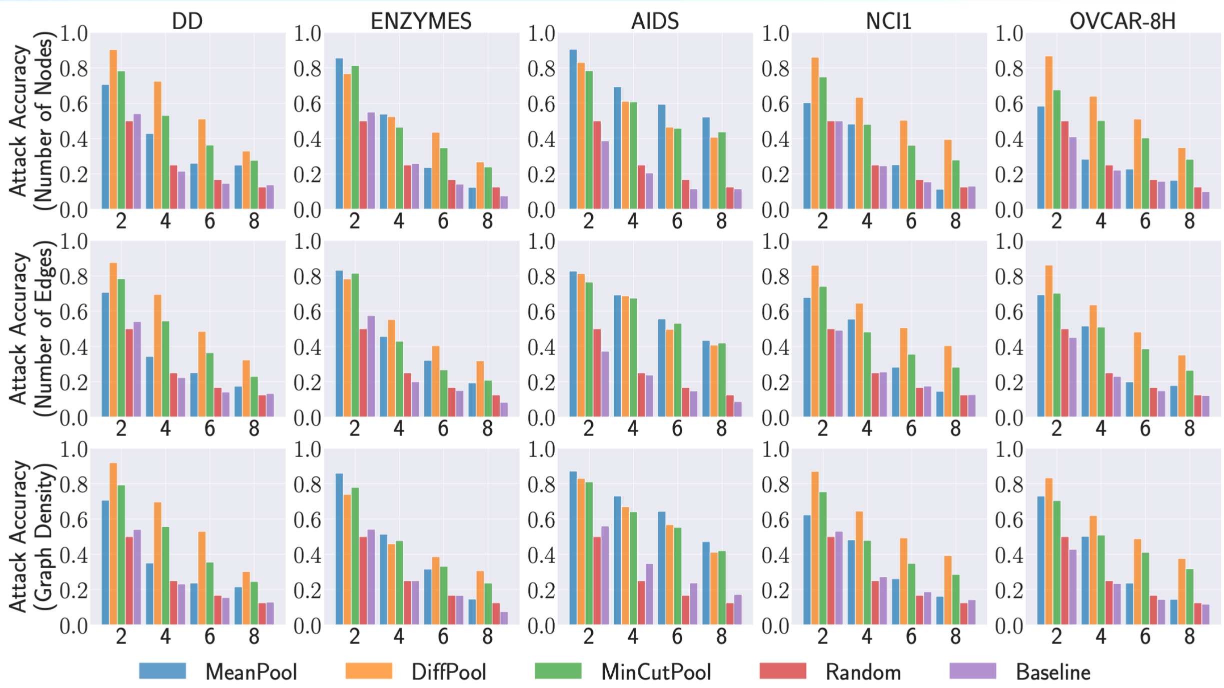
Property Inference Attack



GNN model:
Graph classification

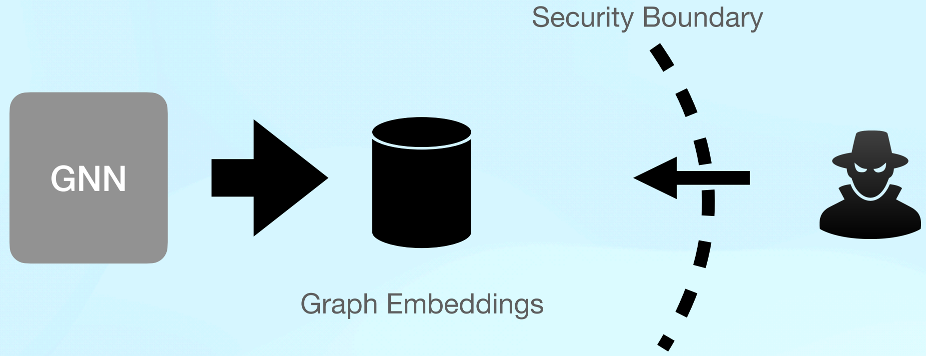
Attacker's **capability**:

1. Embeddings of graphs (from training data) obtained from the target model
2. Can query the GNN model



Subgraph Inference Attack

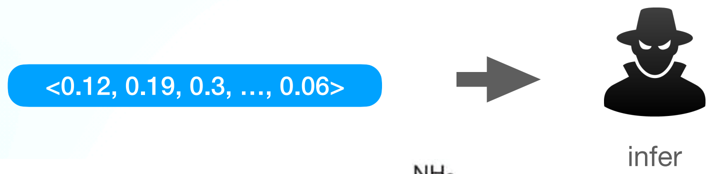
Scenario



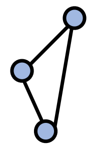
GNN model:
Graph classification

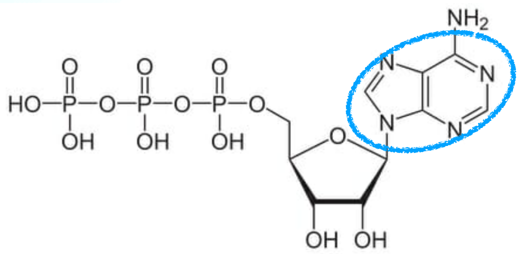
Attacker's **capability**:

1. Embeddings of graphs (from training data) obtained from the target model
2. Can query the GNN model

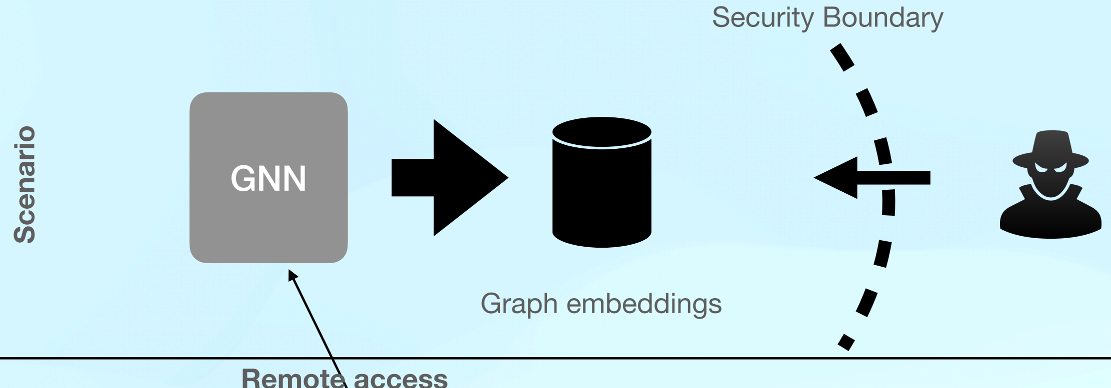


This graph contains at least one



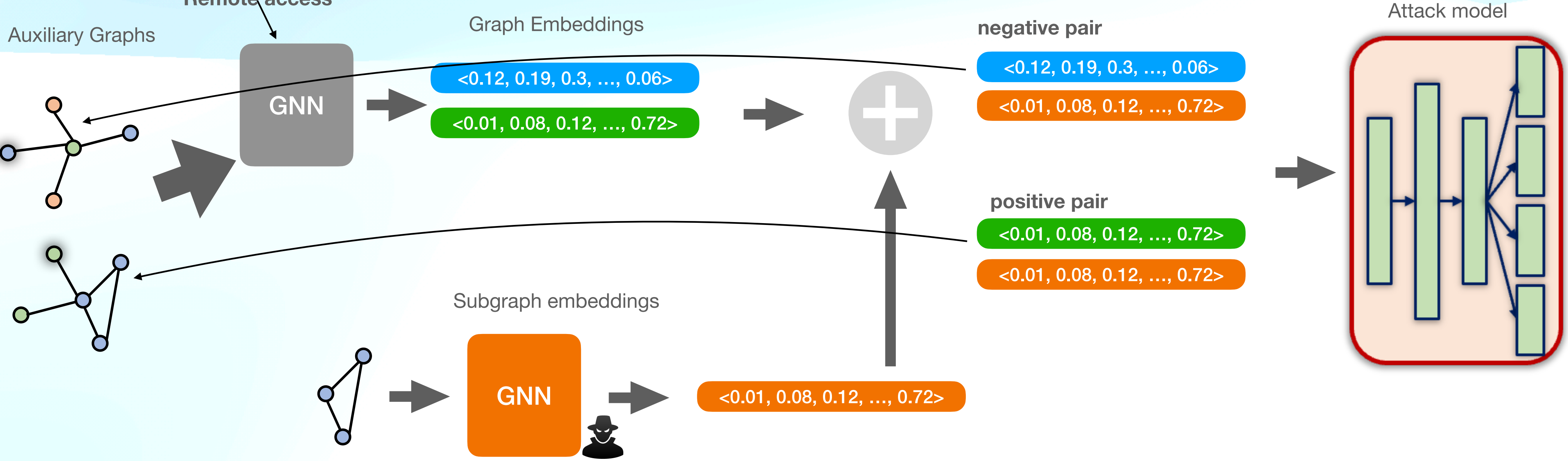


Subgraph Inference Attack



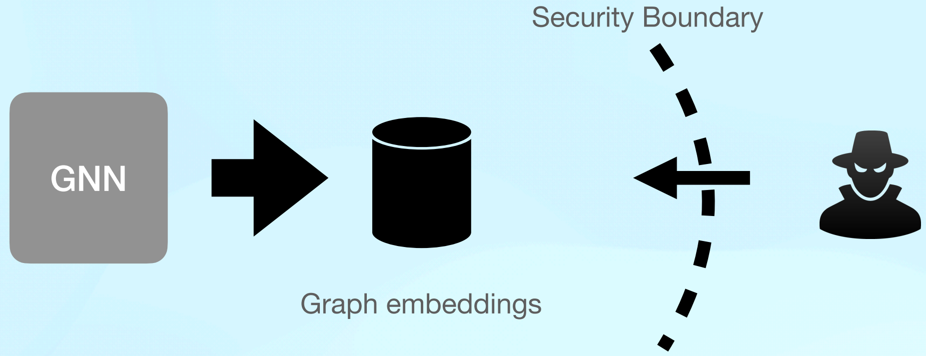
GNN model:
Graph classification

Attacker's **capability**:
1. Embeddings of graphs (from training data) obtained from the target model
2. Can query the GNN model



Subgraph Inference Attack

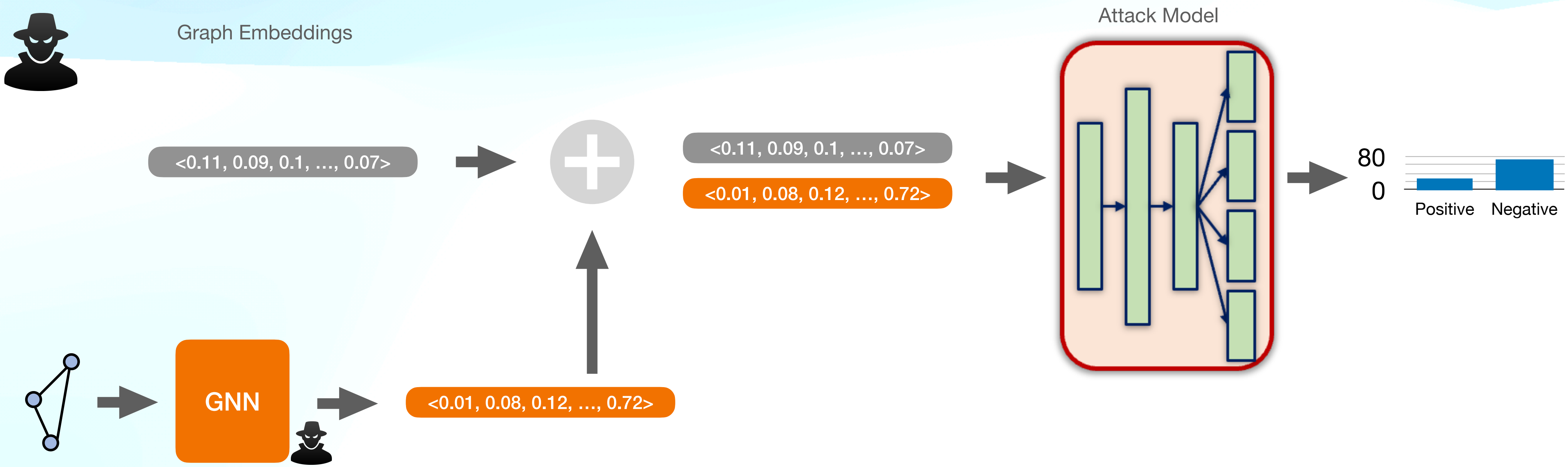
Scenario



GNN model:
Graph classification

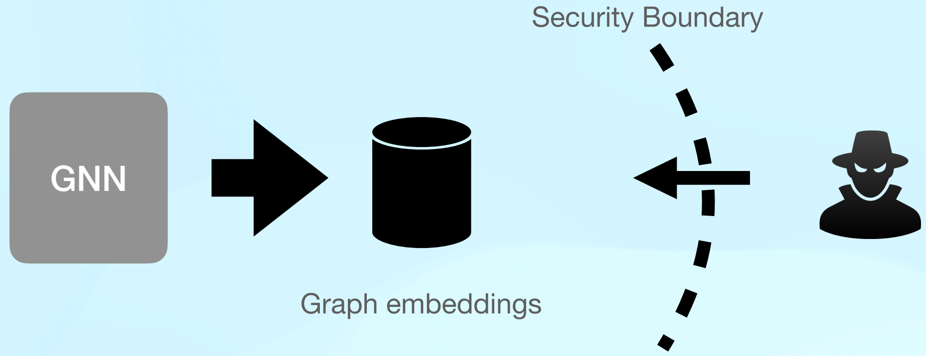
Attacker's **capability**:

1. Embeddings of graphs (from training data) obtained from the target model
2. Can query the GNN model



Subgraph Inference Attack

Scenario

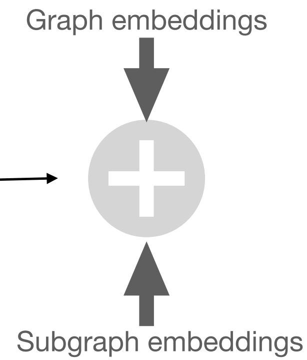


GNN model:
Graph classification

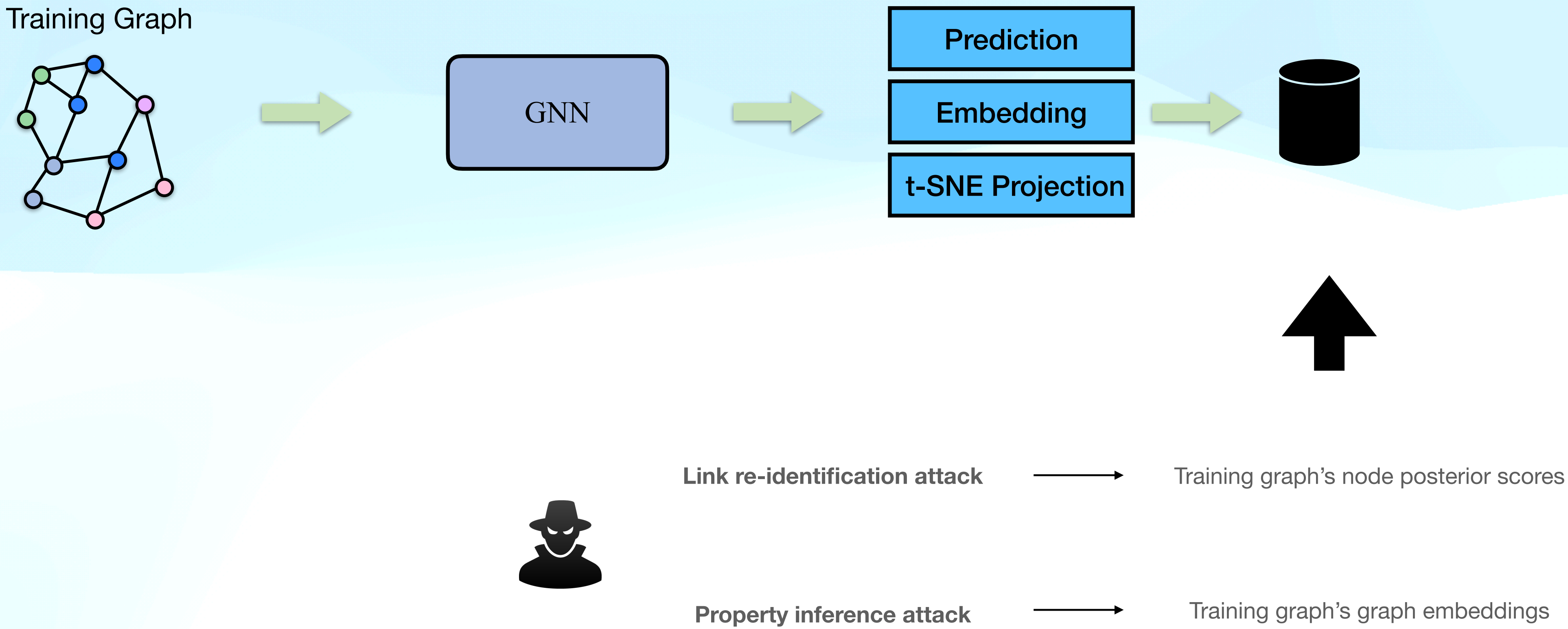
Attacker's **capability**:
1. Embeddings of graphs (from training data) obtained from the target model
2. Can query the GNN model

AUC

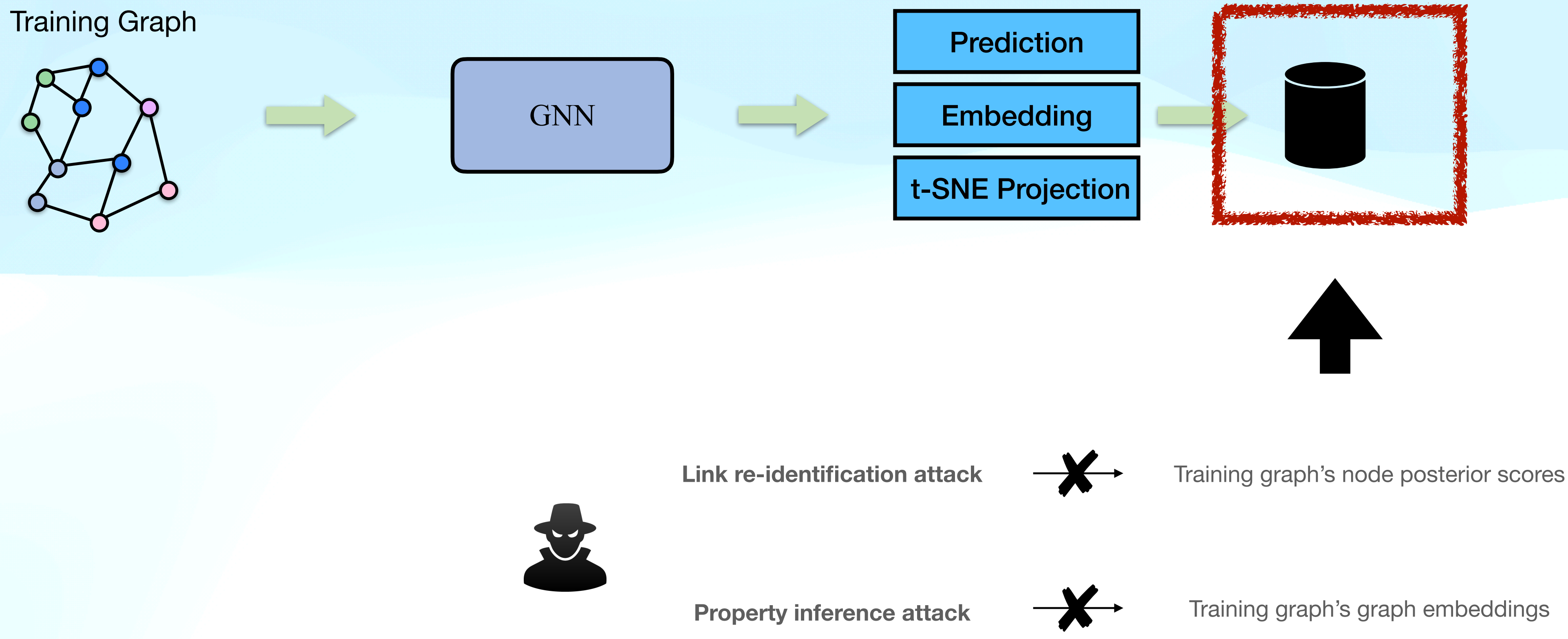
Dataset		0.8			0.6			0.4			0.2	
	Concat	EDist	EDiff	Concat	EDist	EDiff	Concat	EDist	EDiff	Concat	EDist	EDiff
DD	0.53 ± 0.01	0.81 ± 0.06	**0.88 ± 0.01**	0.51 ± 0.01	0.79 ± 0.04	**0.87 ± 0.01**	0.52 ± 0.01	0.79 ± 0.02	**0.85 ± 0.01**	0.50 ± 0.02	0.71 ± 0.08	**0.80 ± 0.00**
ENZYMES	0.49 ± 0.02	0.63 ± 0.10	**0.88 ± 0.03**	0.52 ± 0.03	0.71 ± 0.10	**0.88 ± 0.03**	0.54 ± 0.02	0.56 ± 0.07	**0.86 ± 0.01**	0.48 ± 0.02	0.53 ± 0.03	**0.78 ± 0.01**
AIDS	0.51 ± 0.01	0.53 ± 0.04	**0.78 ± 0.04**	0.55 ± 0.01	0.51 ± 0.02	**0.76 ± 0.05**	0.54 ± 0.01	0.51 ± 0.03	**0.73 ± 0.06**	0.56 ± 0.02	0.50 ± 0.00	**0.76 ± 0.05**
NCI1	0.51 ± 0.00	0.51 ± 0.02	**0.70 ± 0.06**	0.49 ± 0.02	0.52 ± 0.01	**0.67 ± 0.06**	0.50 ± 0.01	0.51 ± 0.01	**0.64 ± 0.03**	0.49 ± 0.01	0.51 ± 0.01	**0.64 ± 0.00**
OVCAR-8H	0.54 ± 0.01	0.63 ± 0.12	**0.89 ± 0.02**	0.50 ± 0.04	0.69 ± 0.09	**0.88 ± 0.02**	0.51 ± 0.03	0.74 ± 0.02	**0.84 ± 0.01**	0.54 ± 0.01	0.60 ± 0.13	**0.82 ± 0.02**



Analysis


Training Graph
GNN
Prediction
Embedding
t-SNE Projection
Link re-identification attack
Training graph's node posterior scores
Property inference attack
Training graph's graph embeddings

Analysis

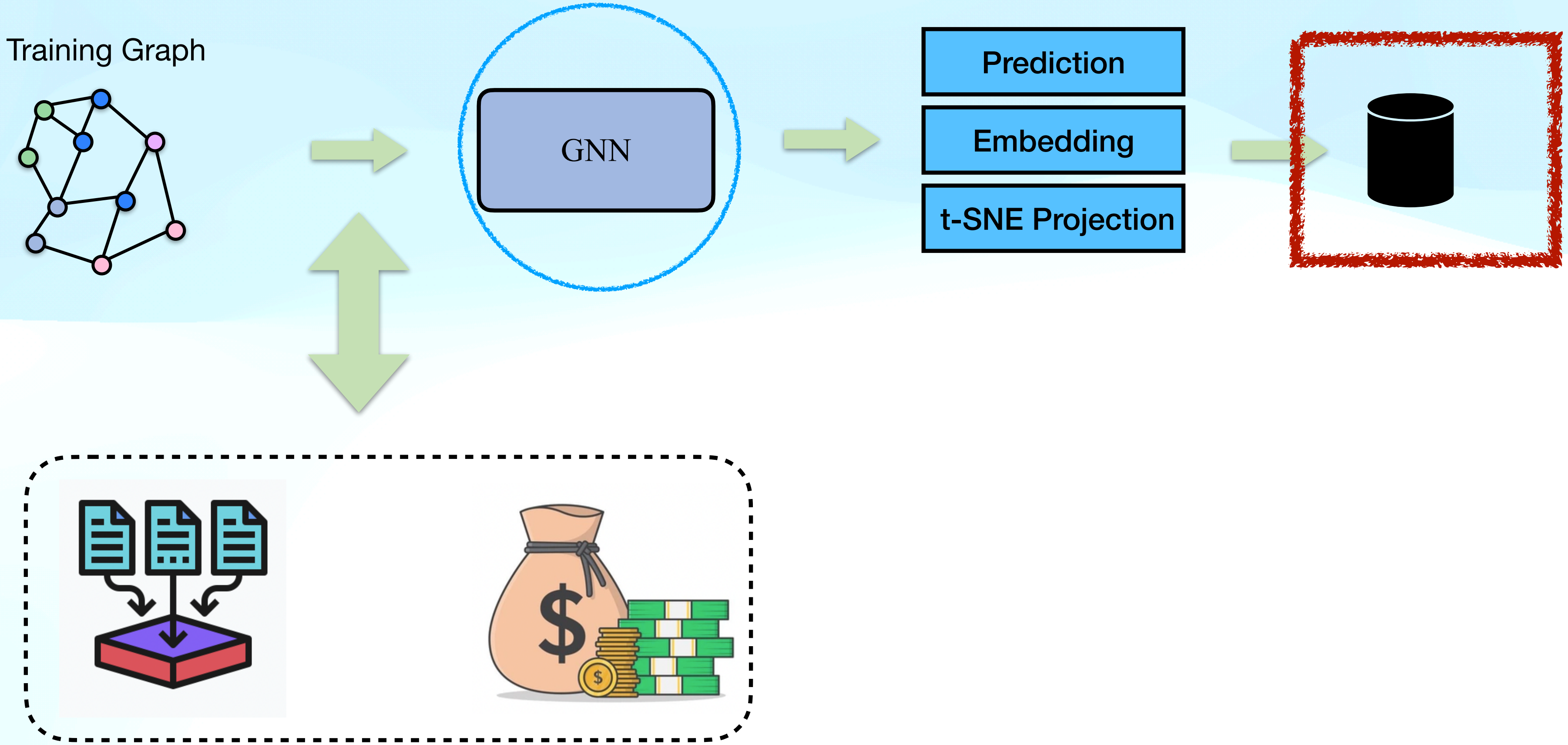


Training Graph
GNN
Prediction
Embedding
t-SNE Projection
Link re-identification attack
Training graph's node posterior scores
Property inference attack
Training graph's graph embeddings

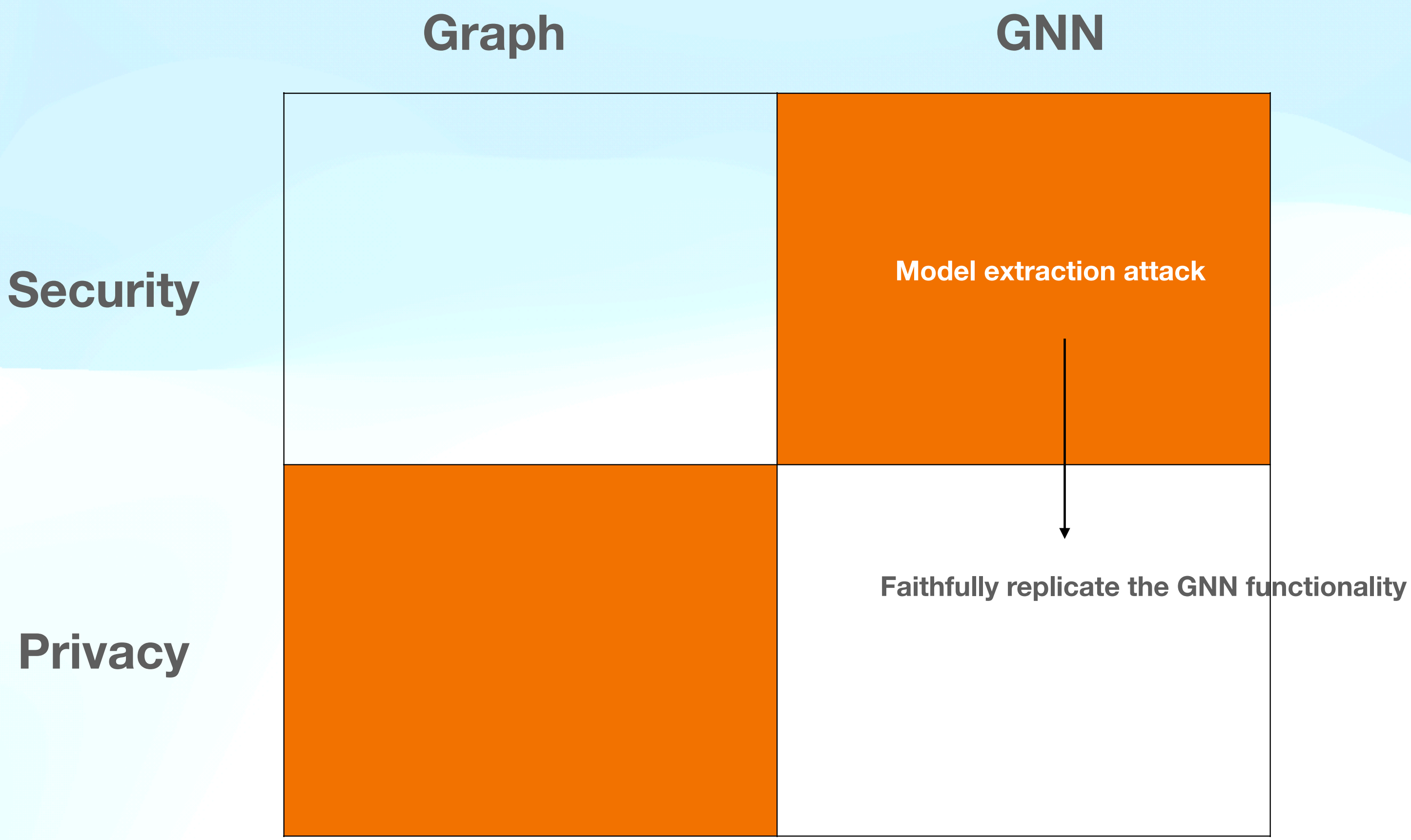
Takeaways (1)

- **Secure** your infrastructure
- **Audit** your GNN-based machine learning pipeline

What Is Next?

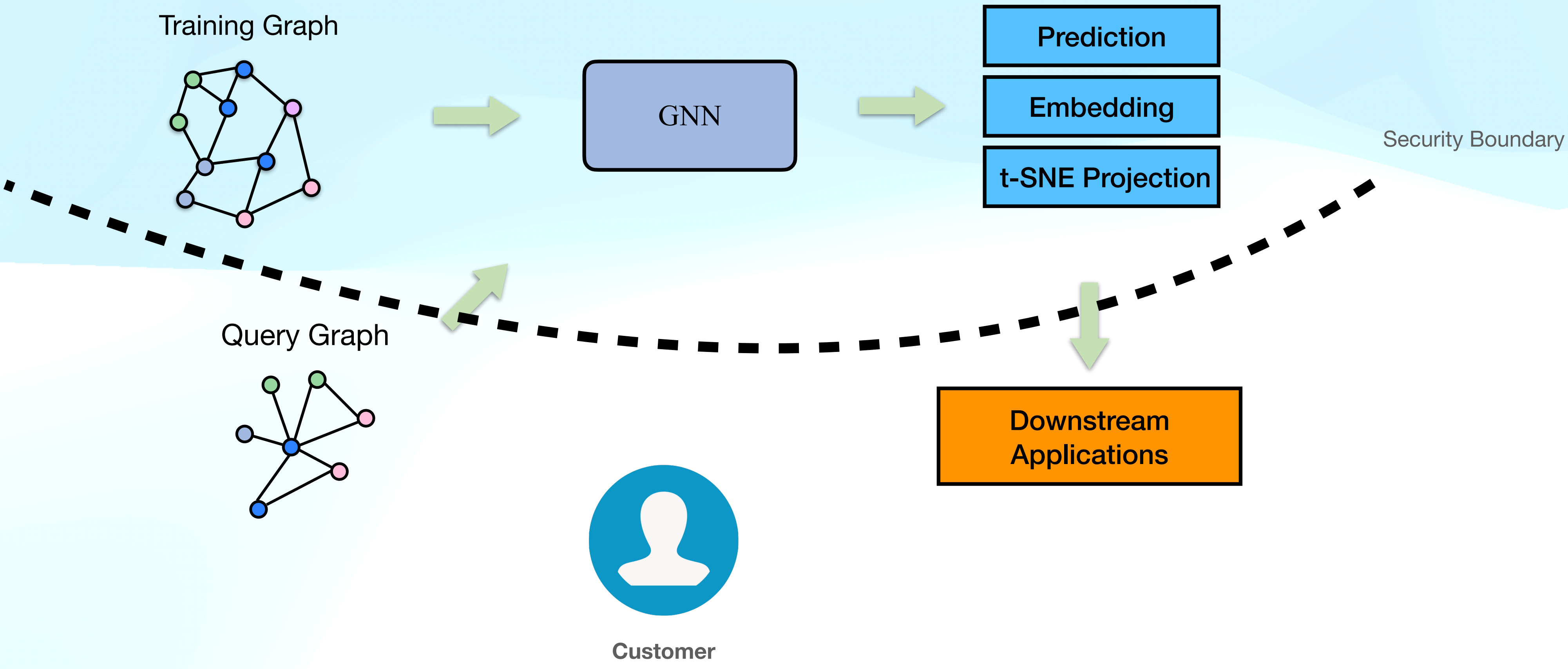


Overview*

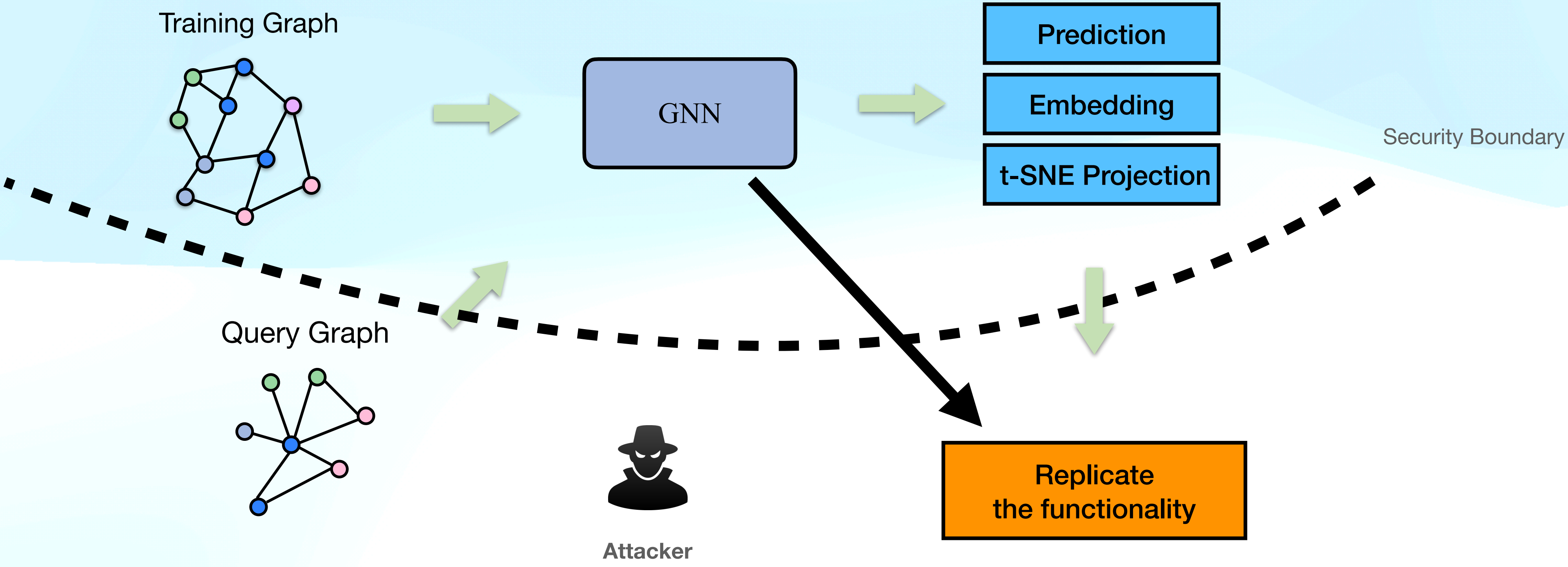


*All attacks discussed in this talk are simulated in the lab environment.

Model Stealing Attack

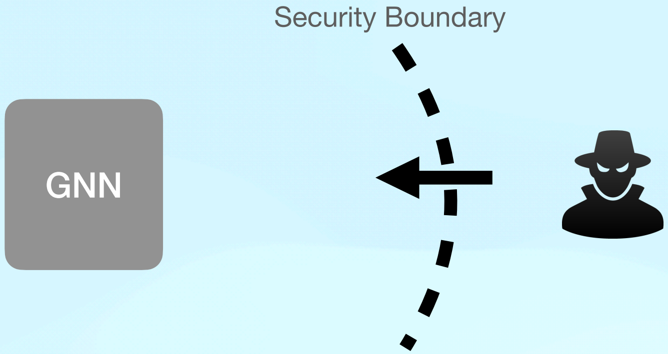


Model Stealing Attack



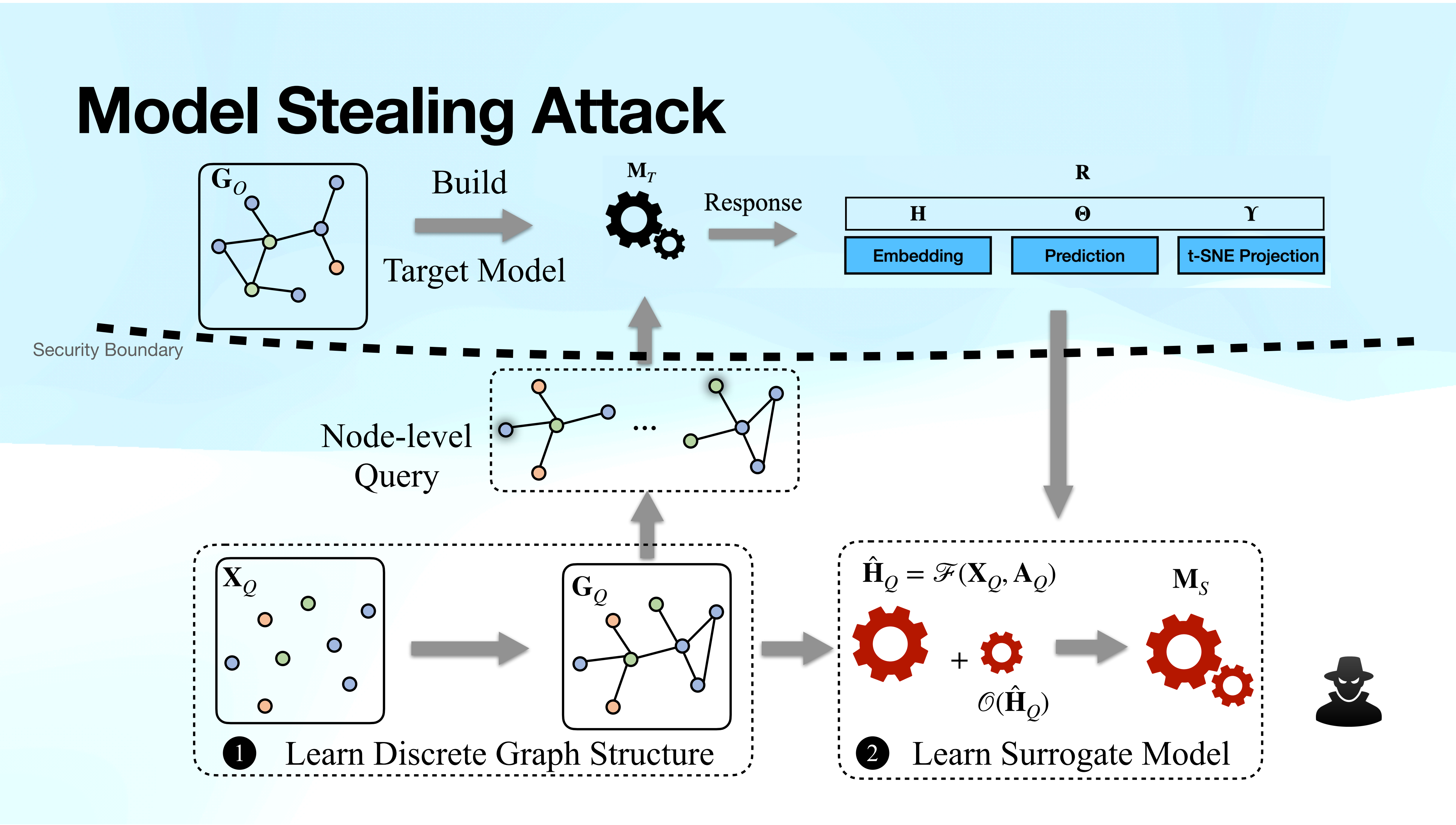
Model Stealing Attack

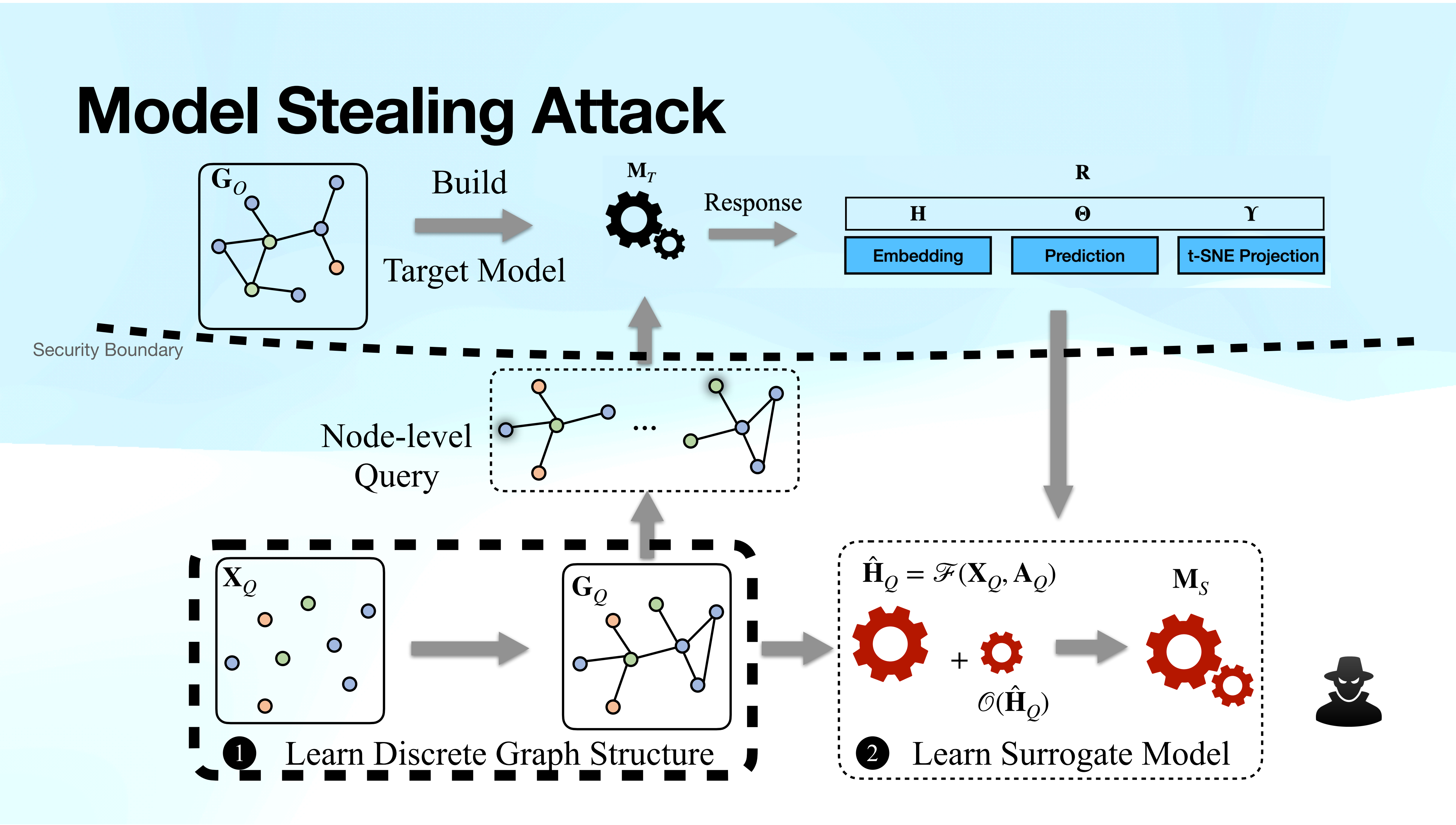
Scenario



GNN model:
Node classification

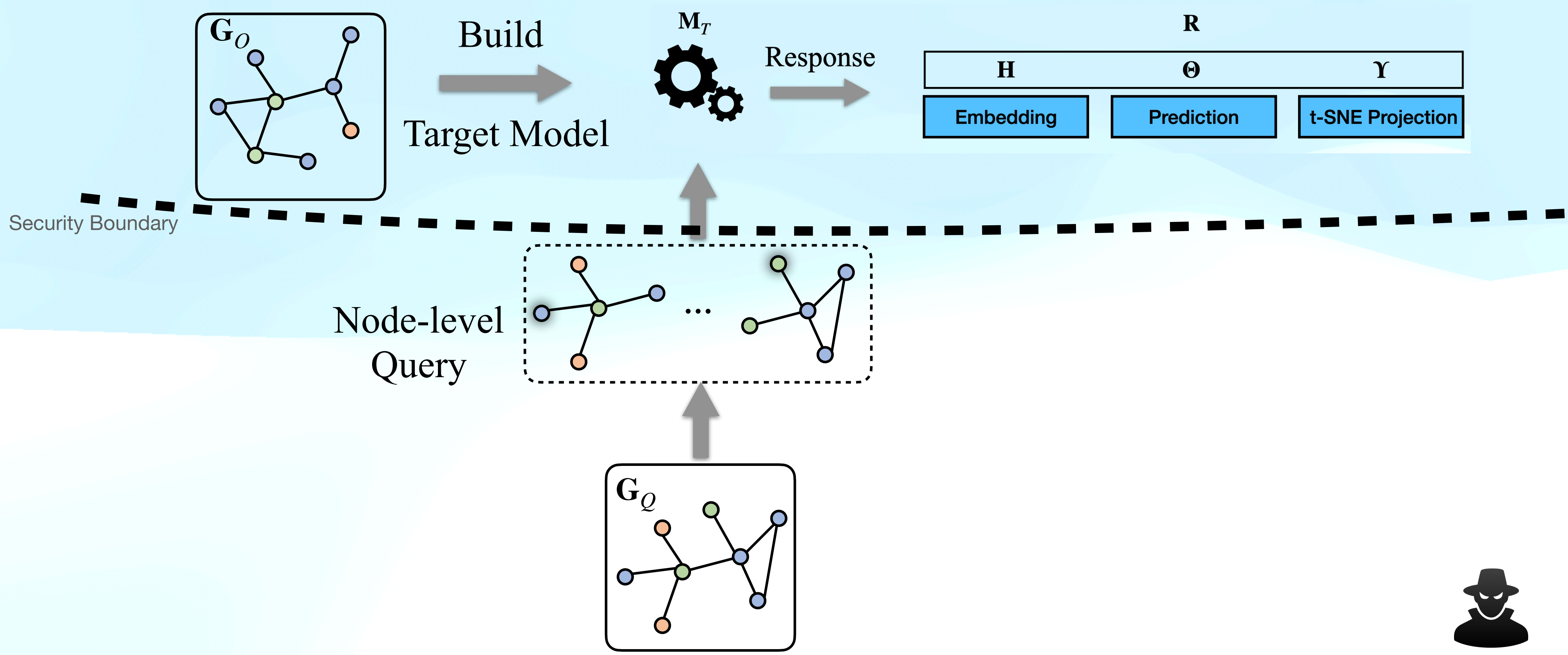
Attacker's **capability**:
1. Can query the GNN model via publicly accessible API


Model Stealing Attack
$\mathbf{G}_O$
Build
Target Model
$\mathbf{M}_T$
Response
$\mathbf{R}$
$\mathbf{H}$
$\mathbf{\Theta}$
$\mathbf{\Upsilon}$
Embedding
Prediction
t-SNE Projection
Security Boundary
Node-level Query
...
$\mathbf{X}_Q$
$\mathbf{G}_Q$
1 Learn Discrete Graph Structure
$\hat{\mathbf{H}}_Q = \mathscr{F}(\mathbf{X}_Q, \mathbf{A}_Q)$
+
$\mathscr{O}(\hat{\mathbf{H}}_Q)$
$\mathbf{M}_S$
2 Learn Surrogate Model

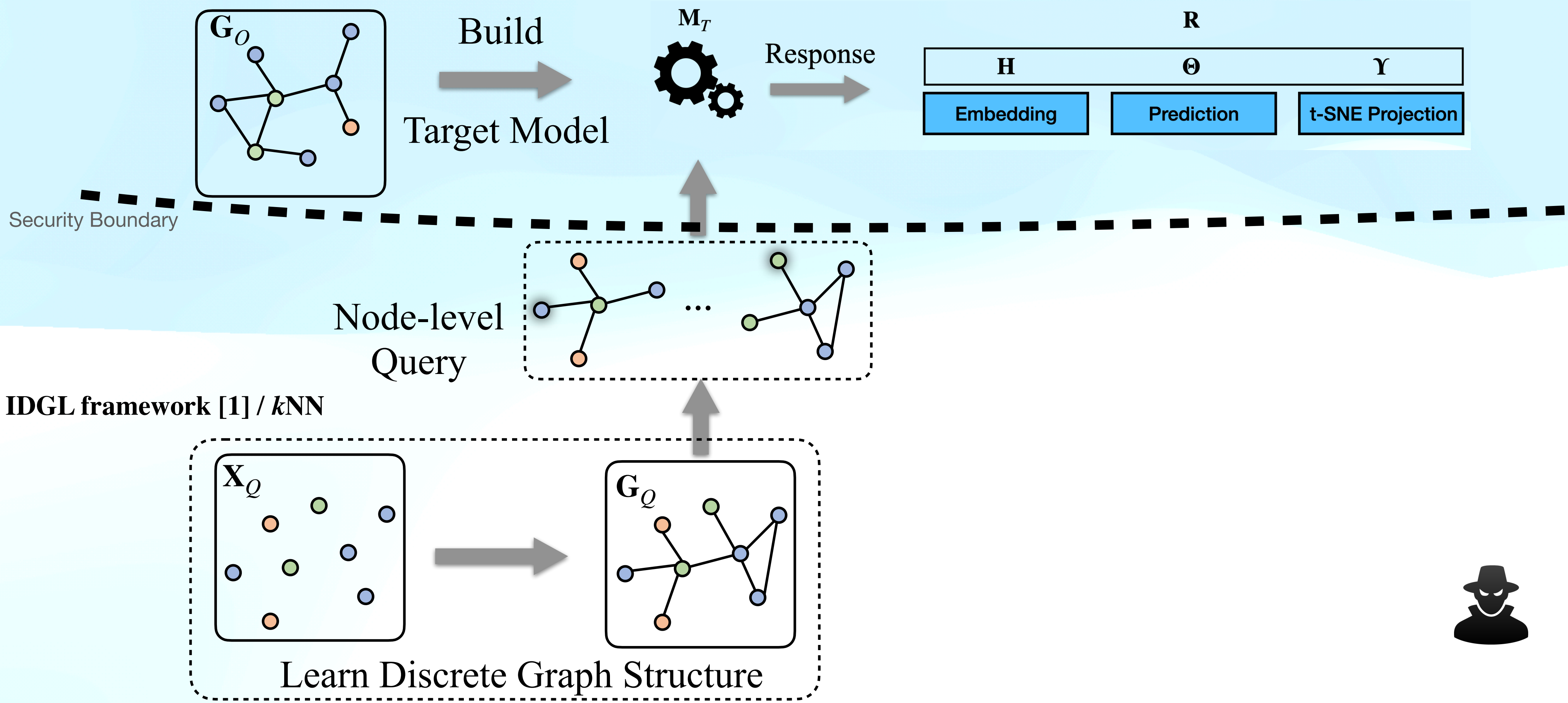


Model Stealing Attack
$\mathbf{G}_O$
Build
Target Model
$\mathbf{M}_T$
Response
$\mathbf{R}$
$\mathbf{H}$
$\mathbf{\Theta}$
$\mathbf{\Upsilon}$
Embedding
Prediction
t-SNE Projection
Security Boundary
Node-level Query
$\mathbf{X}_Q$
$\mathbf{G}_Q$
1 Learn Discrete Graph Structure
$\hat{\mathbf{H}}_Q = \mathscr{F}(\mathbf{X}_Q, \mathbf{A}_Q)$
$\mathbf{M}_S$
$\mathscr{O}(\hat{\mathbf{H}}_Q)$
2 Learn Surrogate Model

Model Stealing Attack

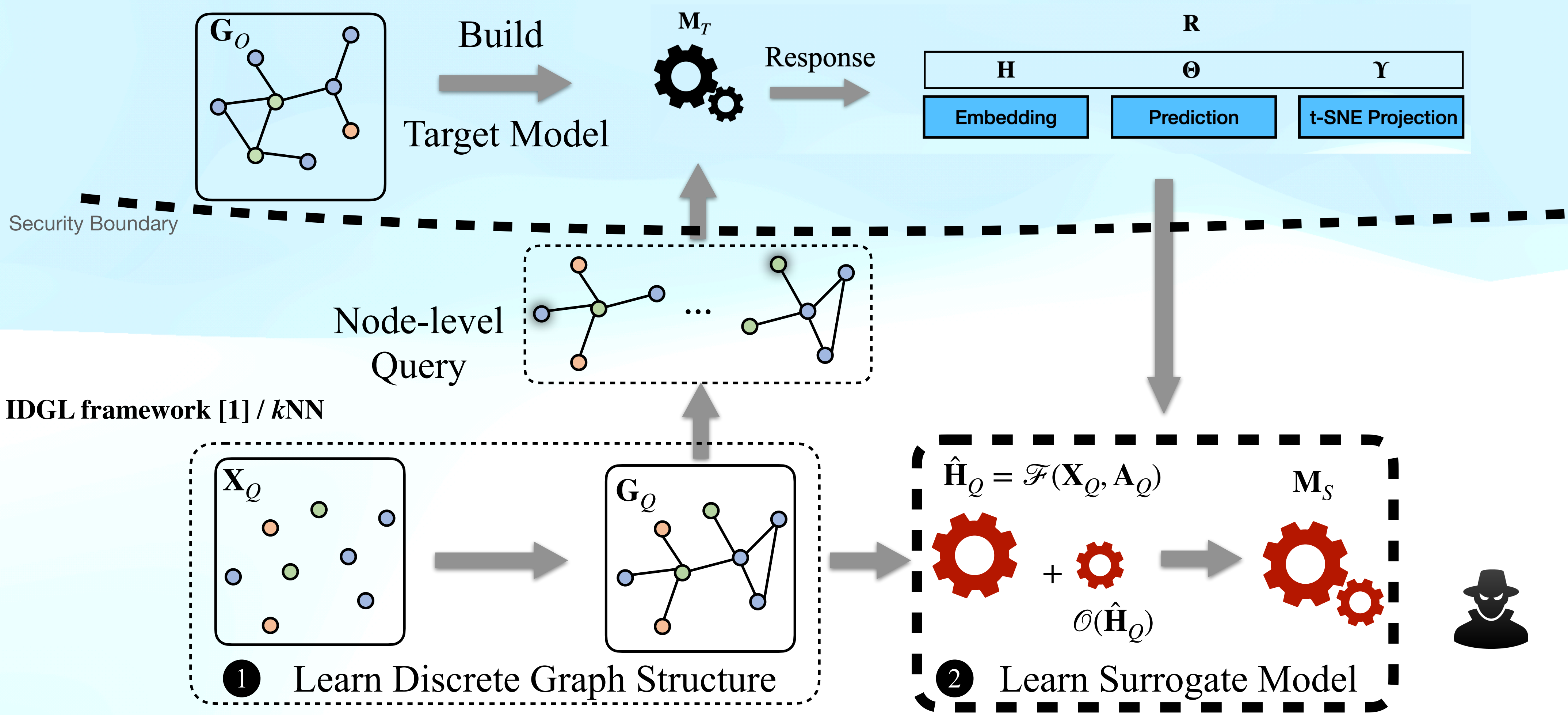


Model Stealing Attack



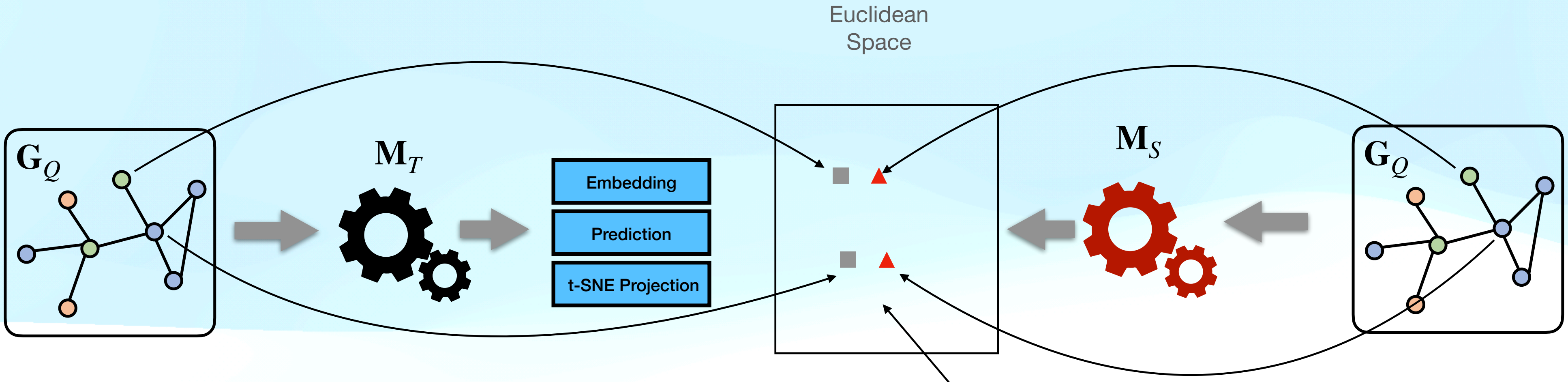
[1] Chen, Yu, Lingfei Wu, and Mohammed Zaki. "Iterative deep graph learning for graph neural networks: Better and robust node embeddings." *Advances in neural information processing systems* 33 (2020)

Model Stealing Attack

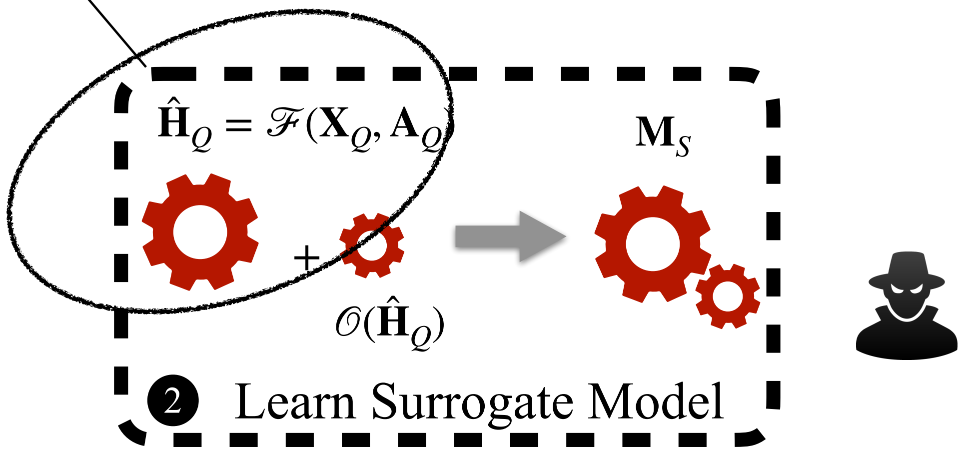


[1] Chen, Yu, Lingfei Wu, and Mohammed Zaki. "Iterative deep graph learning for graph neural networks: Better and robust node embeddings." *Advances in neural information processing systems* 33 (2020)

Model Stealing Attack

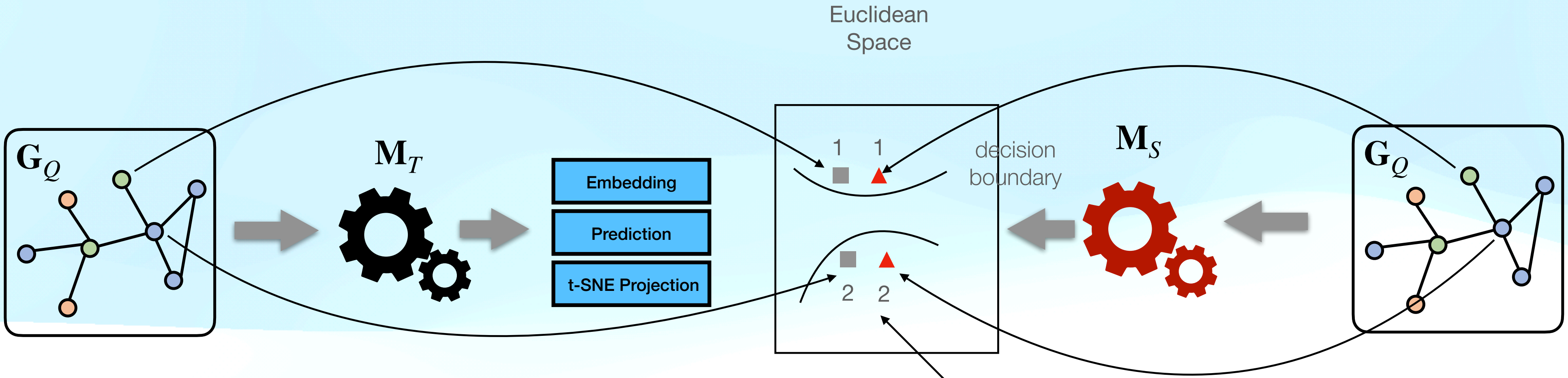


$$
\begin{aligned}
\hat{\mathbf{H}}_Q &= \mathcal{F}(\mathbf{X}_Q, \mathbf{A}_Q) \\
\mathcal{L}_R &= \frac{1}{n_Q} \|\hat{\mathbf{H}}_Q - \mathbf{R}\|_2
\end{aligned}
$$

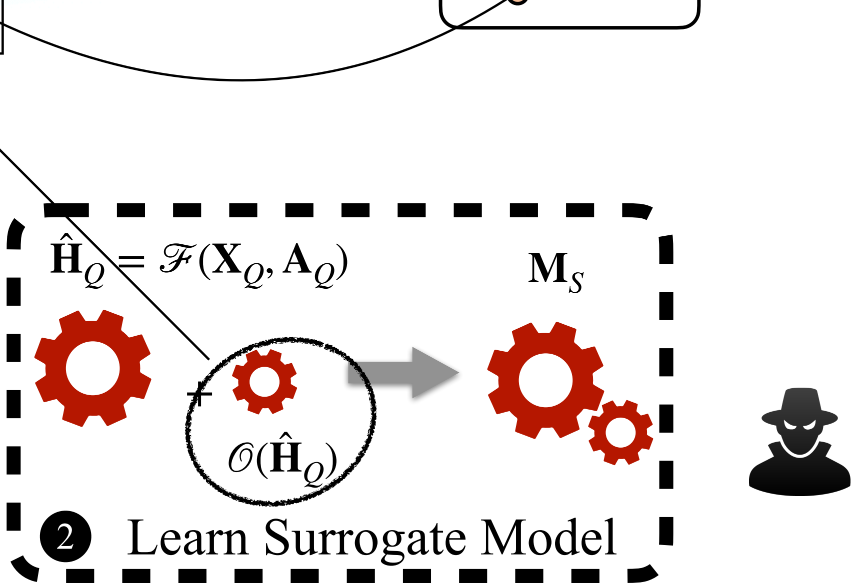


[1] Chen, Yu, Lingfei Wu, and Mohammed Zaki. "Iterative deep graph learning for graph neural networks: Better and robust node embeddings." *Advances in neural information processing systems* 33 (2020)

Model Stealing Attack

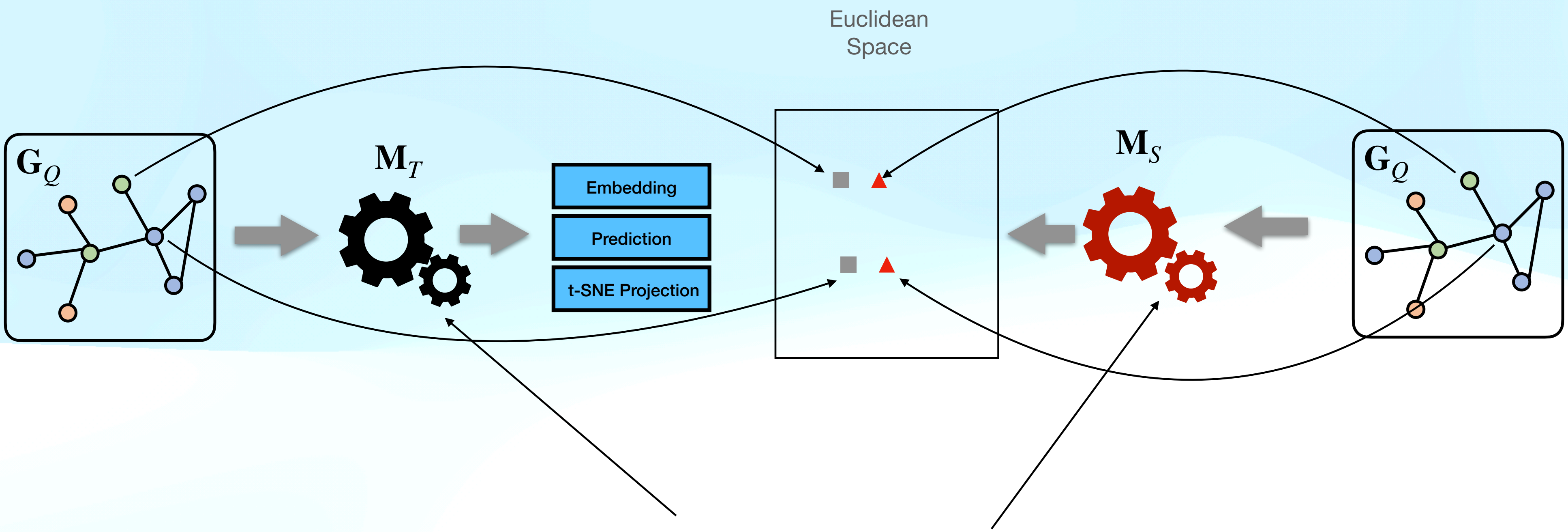


$$\mathcal{L}_P = -\frac{1}{n_Q} \sum_{v \in \mathbf{G}_Q} \sum_{i \in |\mathbf{C}_Q|} c_i log[O(\mathbf{h}_v)_i]$$



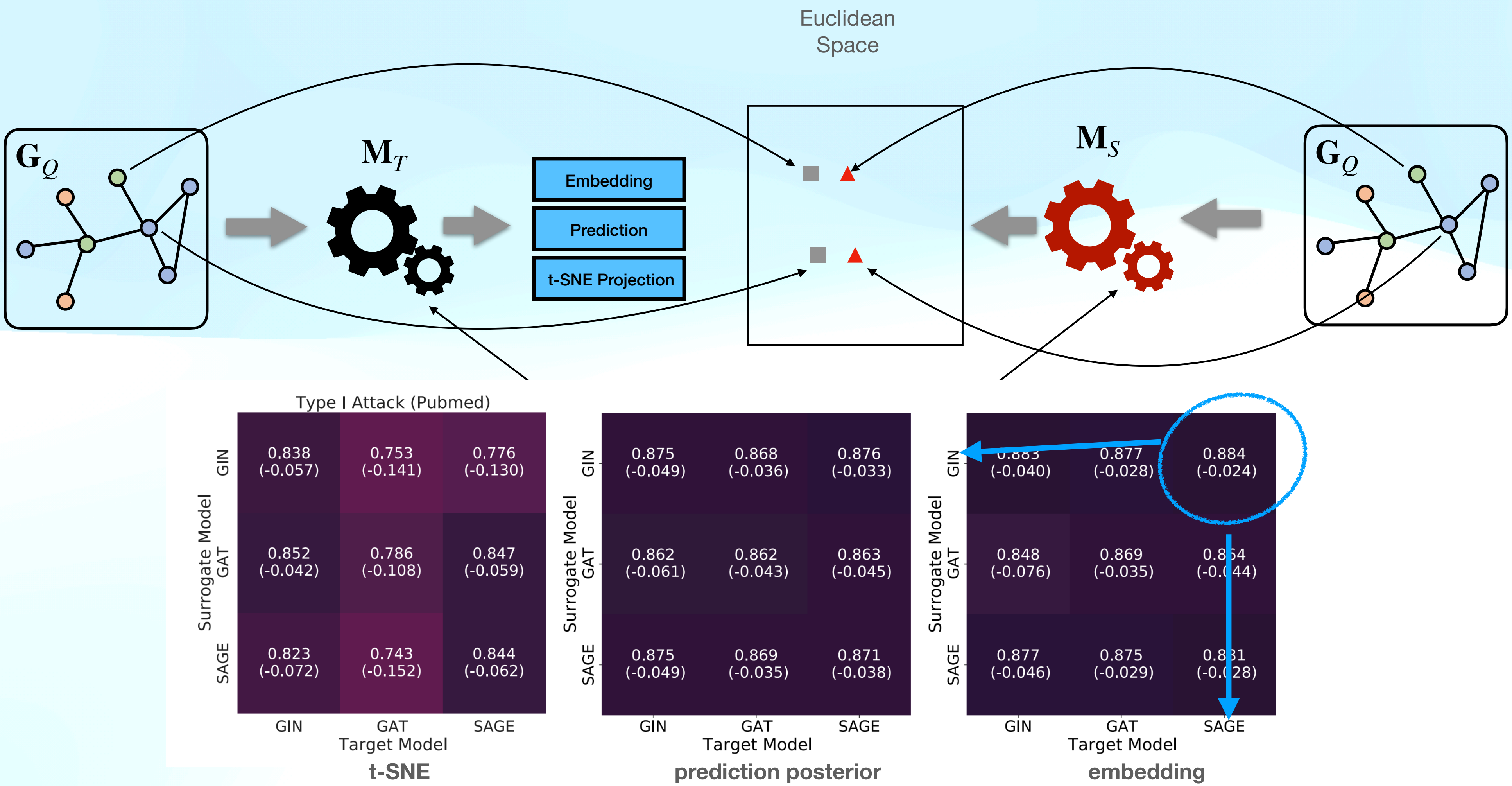
[1] Chen, Yu, Lingfei Wu, and Mohammed Zaki. "Iterative deep graph learning for graph neural networks: Better and robust node embeddings." *Advances in neural information processing systems* 33 (2020)

Model Stealing Attack

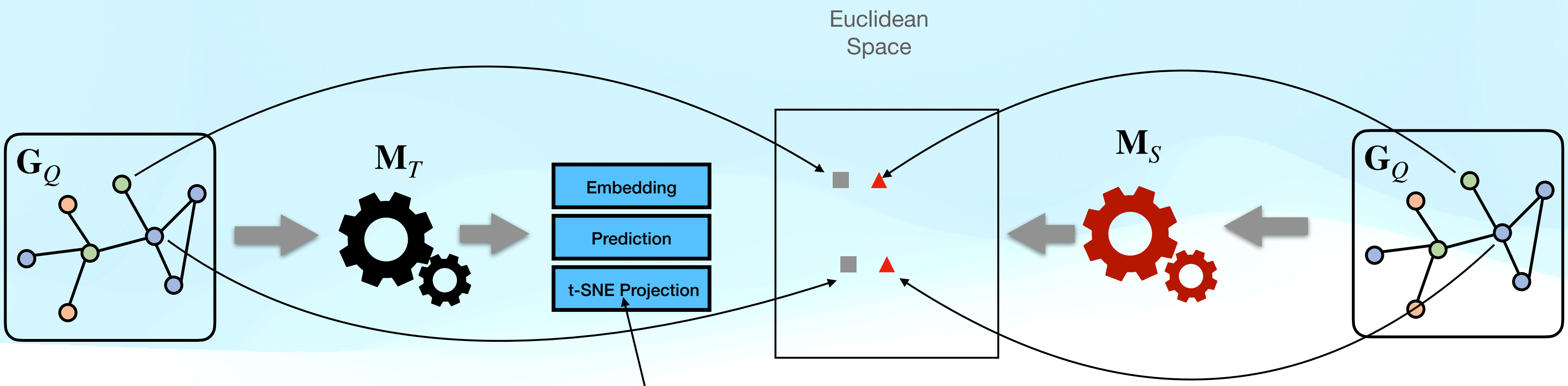


conduct the attack without knowing the target model's architecture

Model Stealing Attack

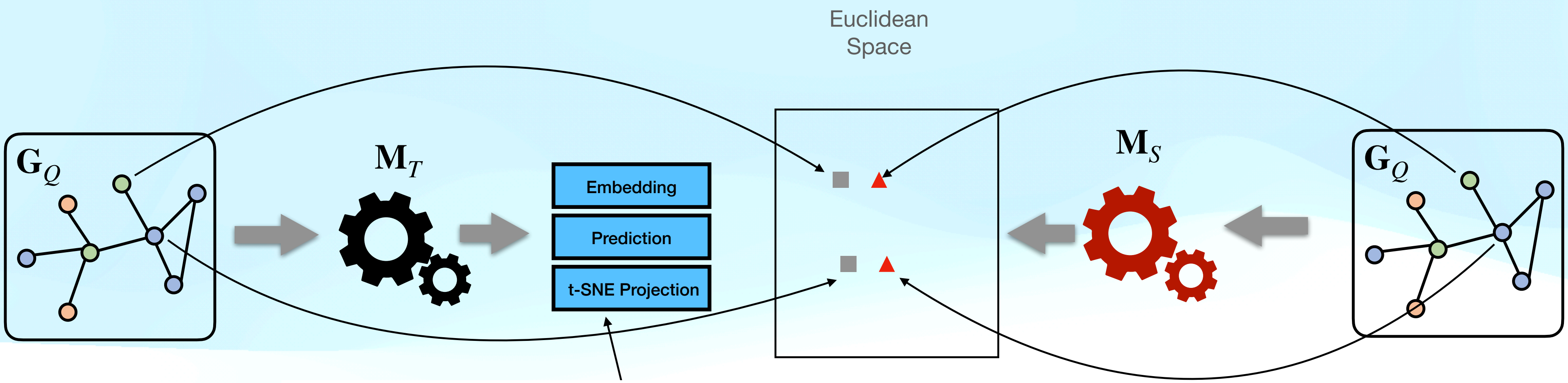


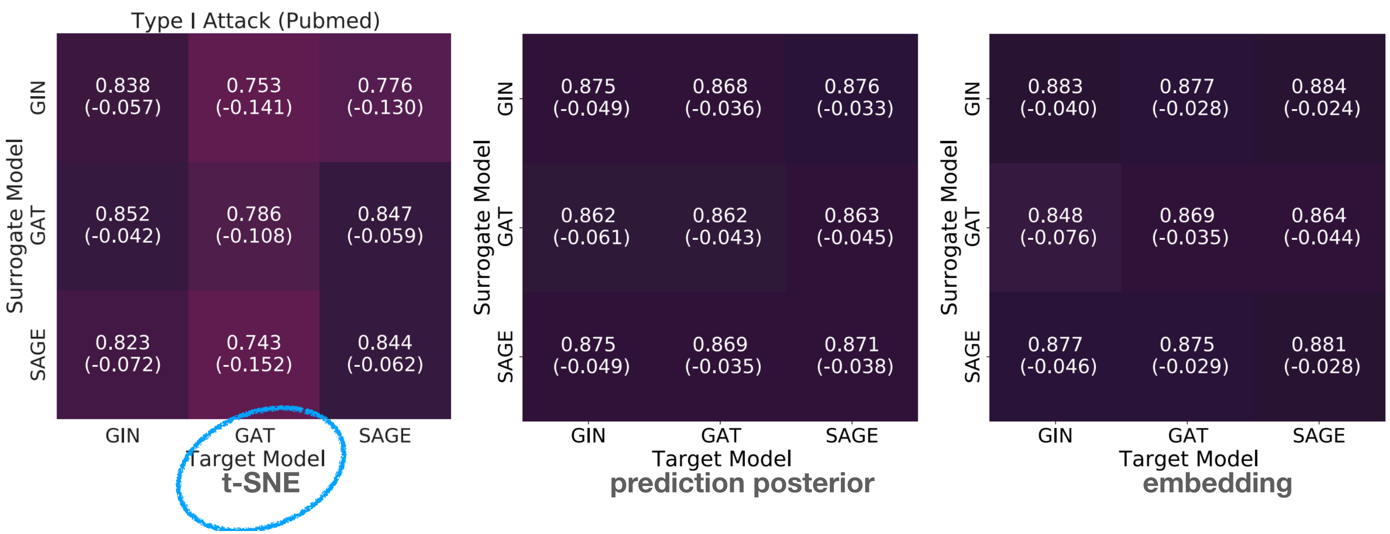
Model Stealing Attack



2 dimensional t-SNE projection can be the new attack surface

Model Stealing Attack





Takeaways (2)

- <u>**Secure**</u> your infrastructure
- <u>**Audit**</u> your GNN-based machine learning pipeline
- <u>**Monitor**</u> your model logs for anomalies
- <u>**Evaluate the security and privacy posture**</u> of your Graph Neural Network (GNN) models

Code

- **Link re-identification attack**

  https://github.com/xinleihe/link_stealing_attack

- **Property/Subgraph inference attack**

  https://github.com/Zhangzhk0819/GNN-Embedding-Leaks

- **Model stealing attack**

  https://github.com/xinleihe/GNNStealing

Thank You

Yang Zhang and his research group
CISPA Helmholtz Center for Information Security
zhang@cispa.de

Azzedine Benameur and Yun Shen
Spot by NetApp
{Azzedine.Benameur, Yun.Shen}@netapp.com