



重庆大学
CHONGQING UNIVERSITY



THE UNIVERSITY
OF QUEENSLAND
AUSTRALIA



Towards Secure and Robust Recommender Systems: A Data-Centric Perspective

Zongwei Wang¹, Junliang Yu², Tong Chen², Hongzhi Yin², Shazia Sadiq², Min Gao¹

¹Chongqing University, China,

²The University of Queensland, Australia

Presenters



Zongwei Wang

- A second-year PhD student
School of Big Data and Software
Engineering, CQU

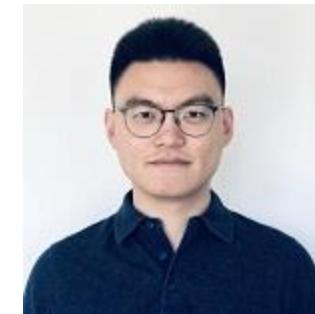
 zongwei@cqu.edu.cn



Dr. Junliang Yu

- Lecturer
School of Information Technology and
Electrical Engineering, UQ

 jl.yu@uq.edu.au



Dr. Tong Chen

- Senior Lecturer
School of Information Technology and
Electrical Engineering, UQ

 tong.chen@uq.edu.au

1. Introduction

2. Securing Data Integrity

3. Preserving Data Privacy

4. Managing Data Noise

5. Limitations and Opportunities

6. Toolkit

1. Introduction



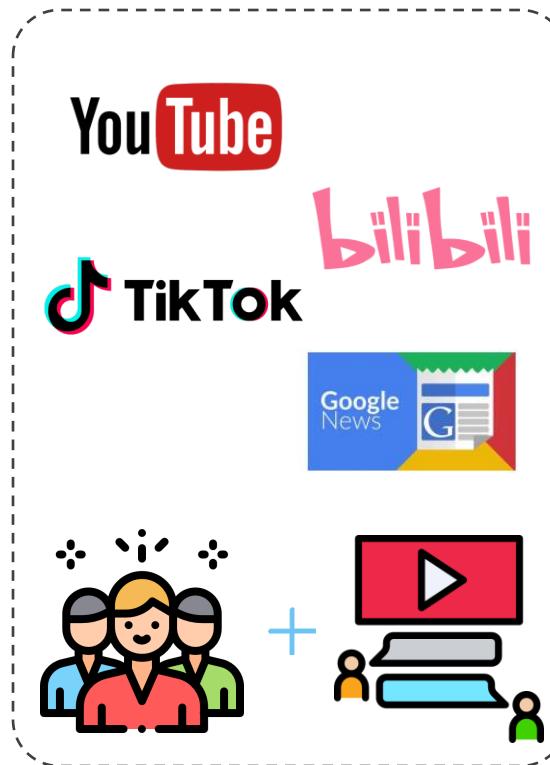
Different Platforms Use Recommender System

1. Introduction

Recommender Systems Find Widespread Application in the Following Fields:



E-Commerce



Video Streaming



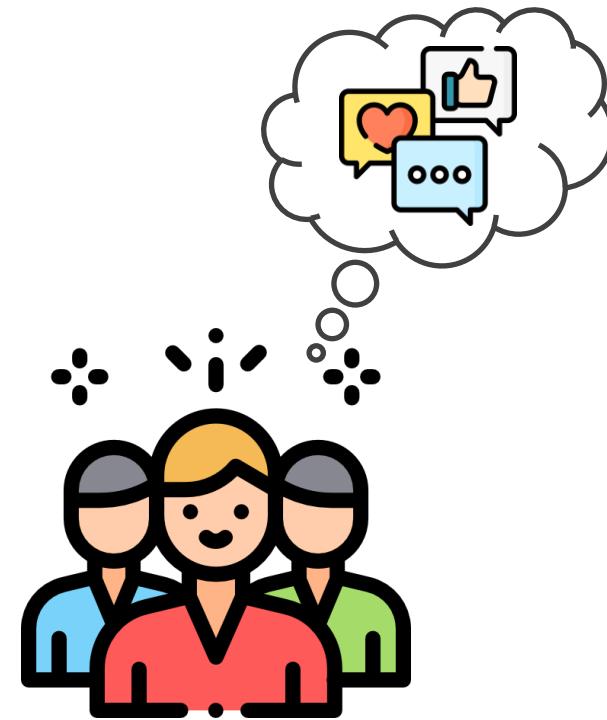
Social Media

1. Introduction



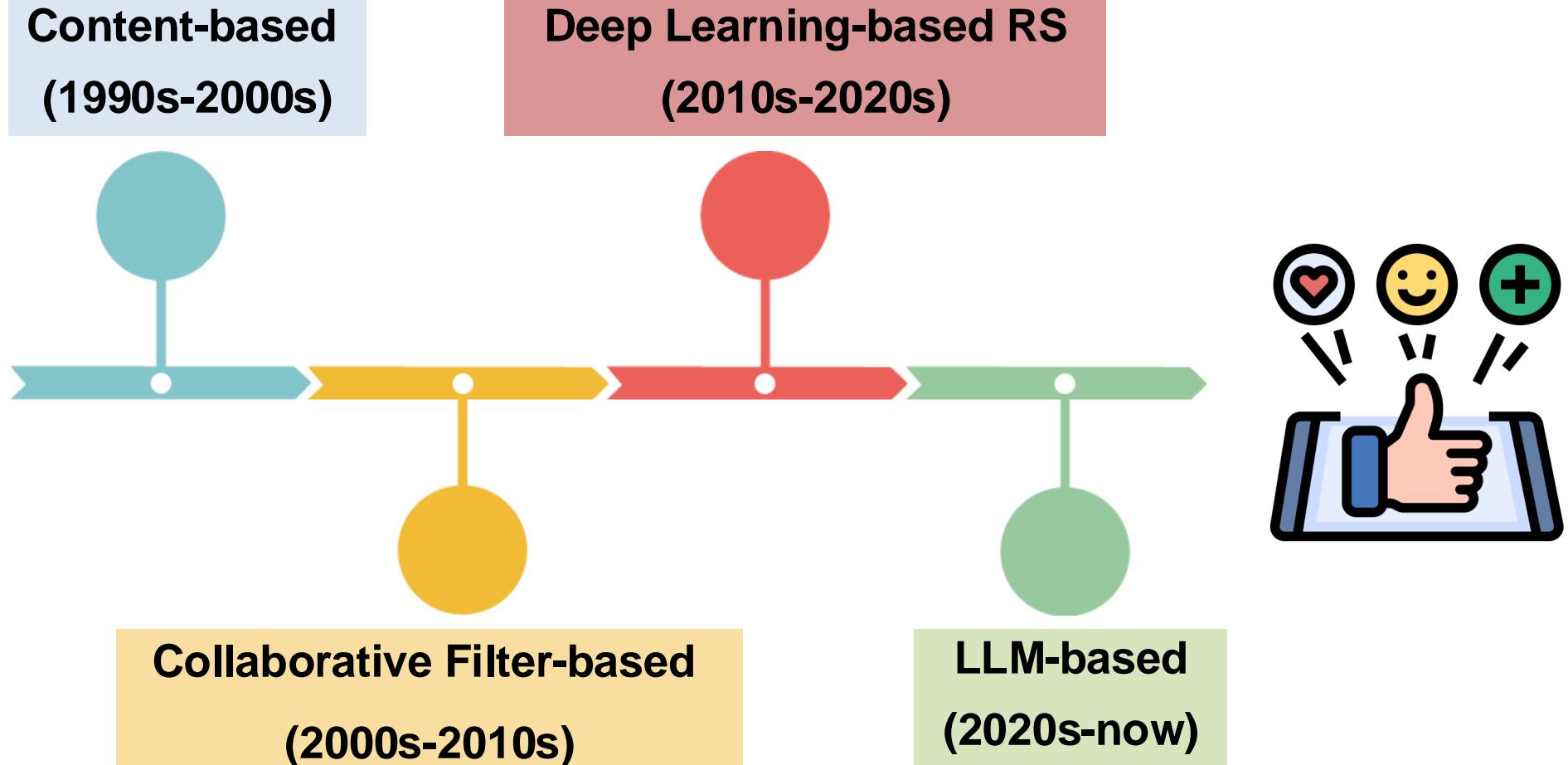
Items: Products, Movies,
Friends, Videos, ...

Recommend Item X
→
to Users



**Large Number of
Users**

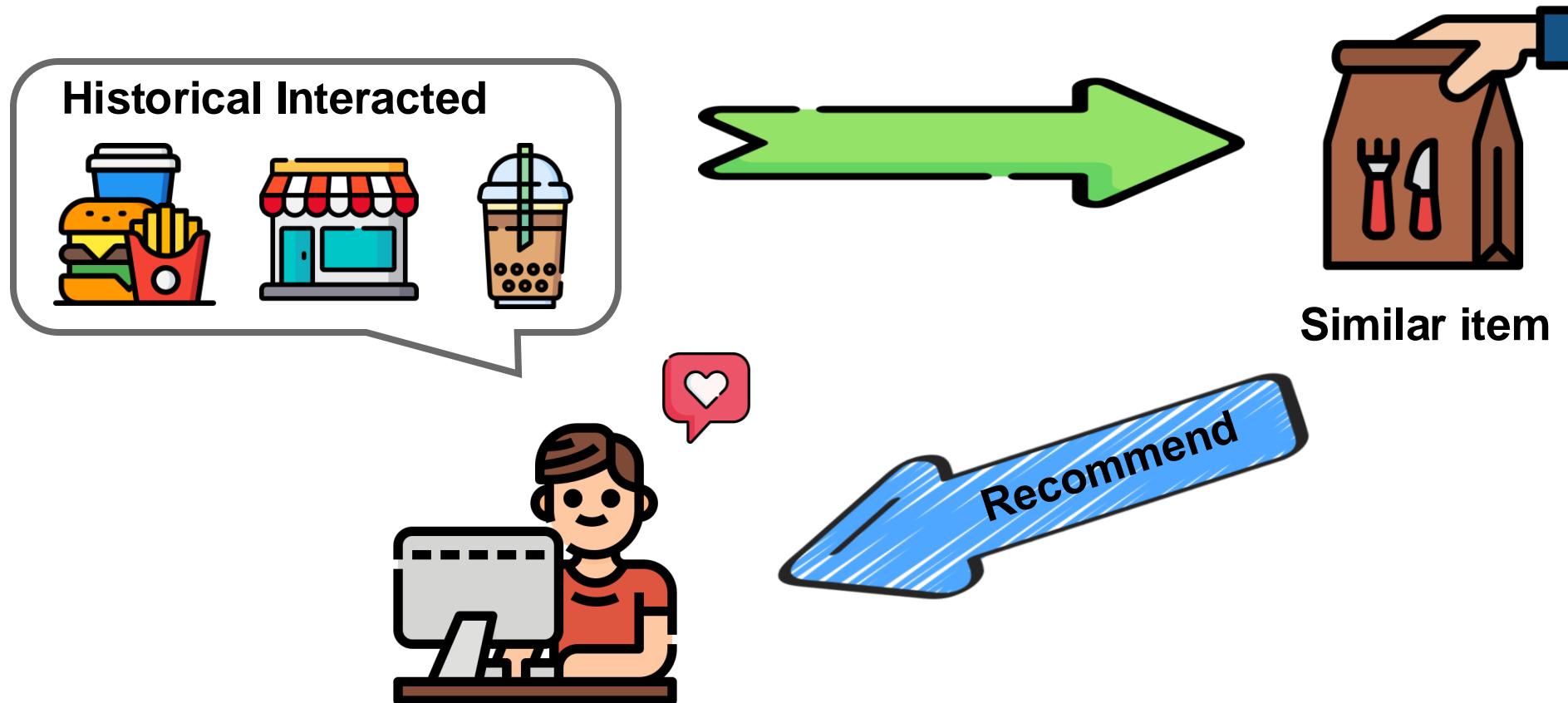
1. Introduction



1. Introduction

Content-based RS (1990s-2000s)

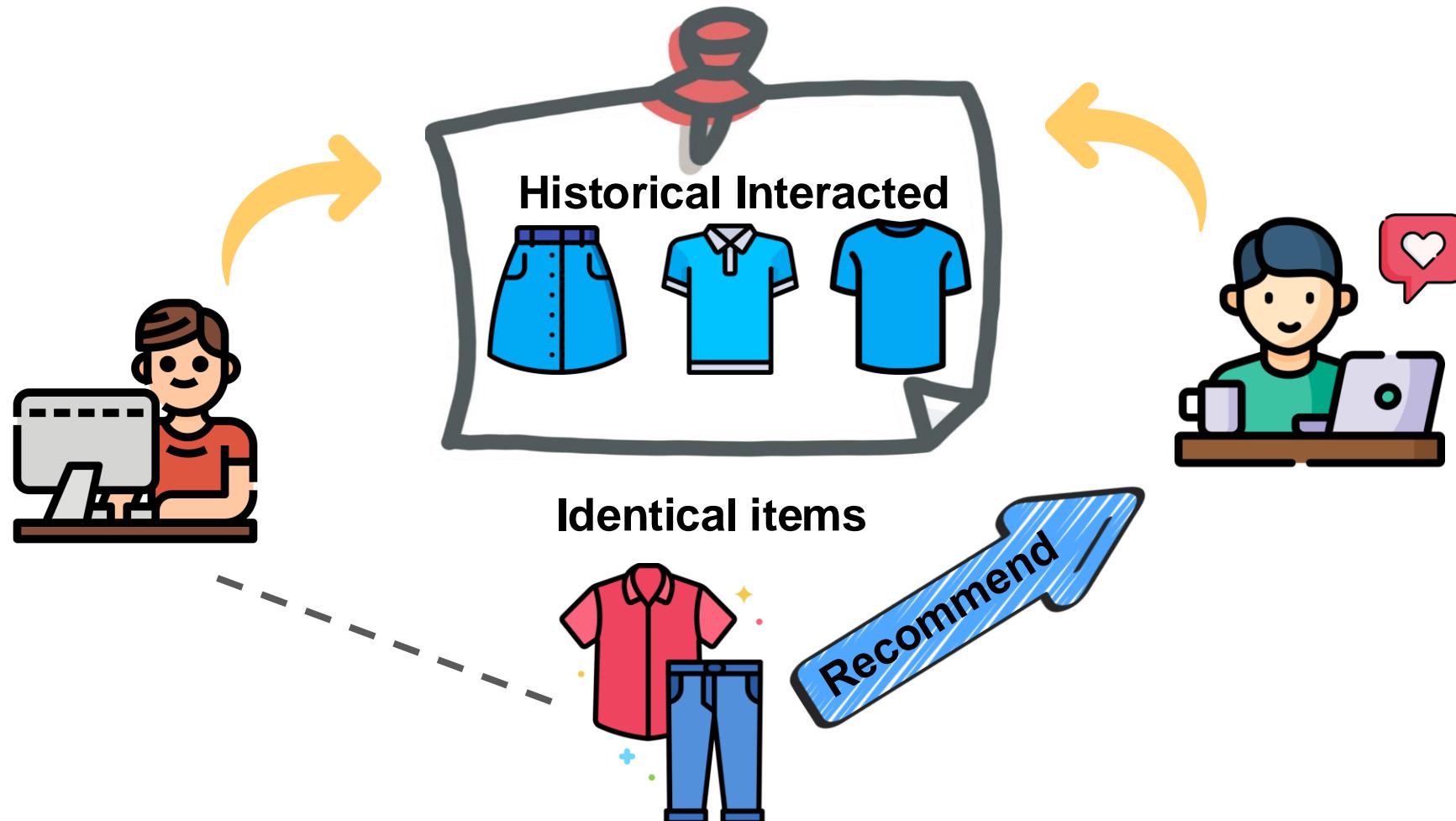
Users are likely to prefer items whose **content is similar to** those they have interacted with before.



1. Introduction

Collaborative Filter-based RS (2000s-2010s)

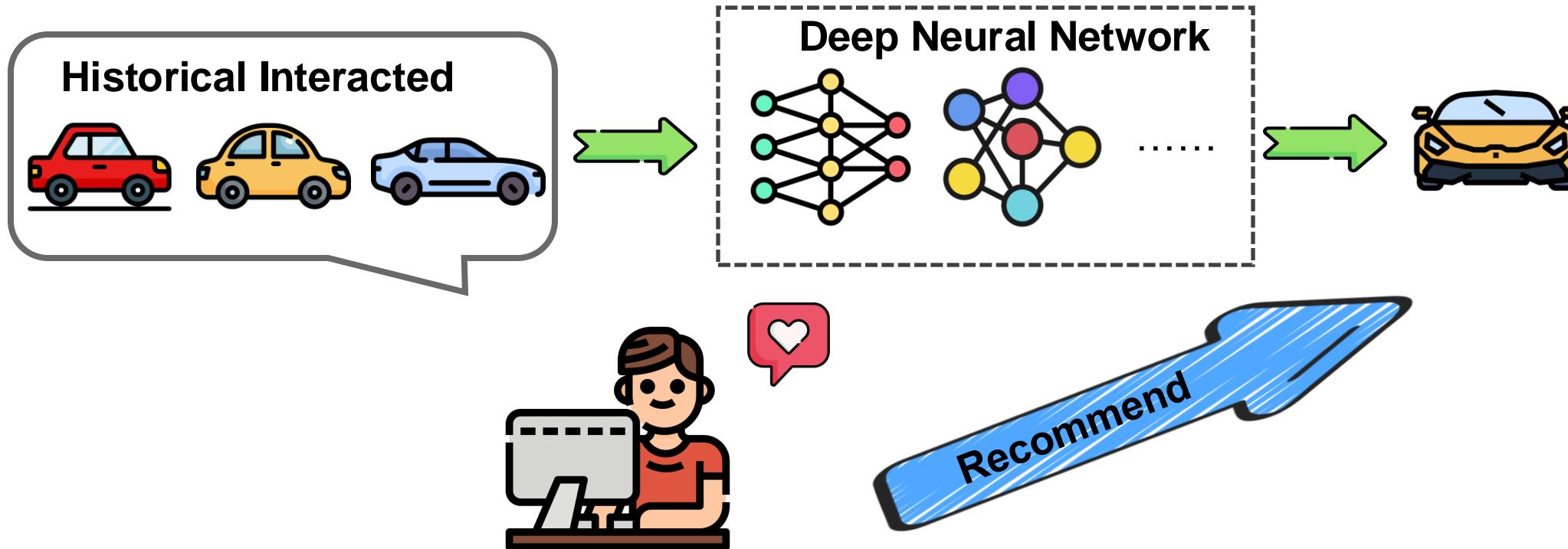
Users with **similar behaviours** tend to have similar preferences.



1. Introduction

Deep Learning-based RS (2010s-2020s)

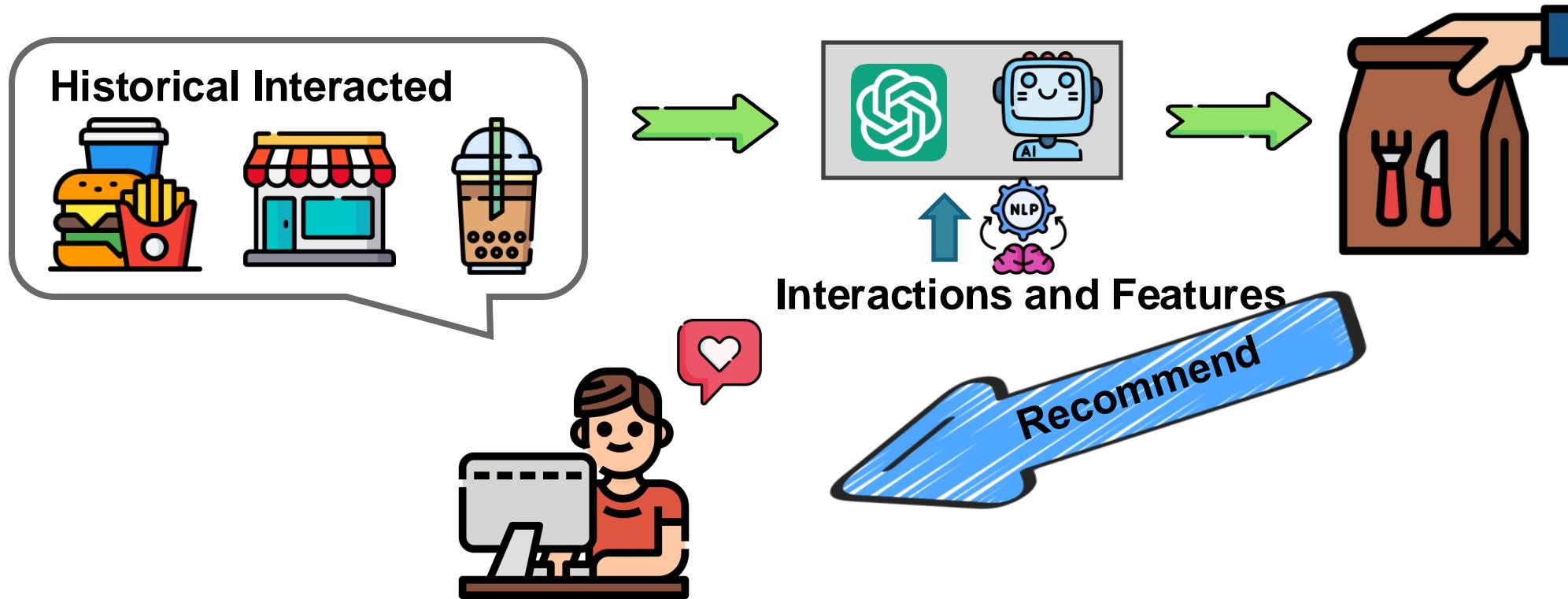
Applying **deep learning techniques** to uncover user preferences.



1. Introduction

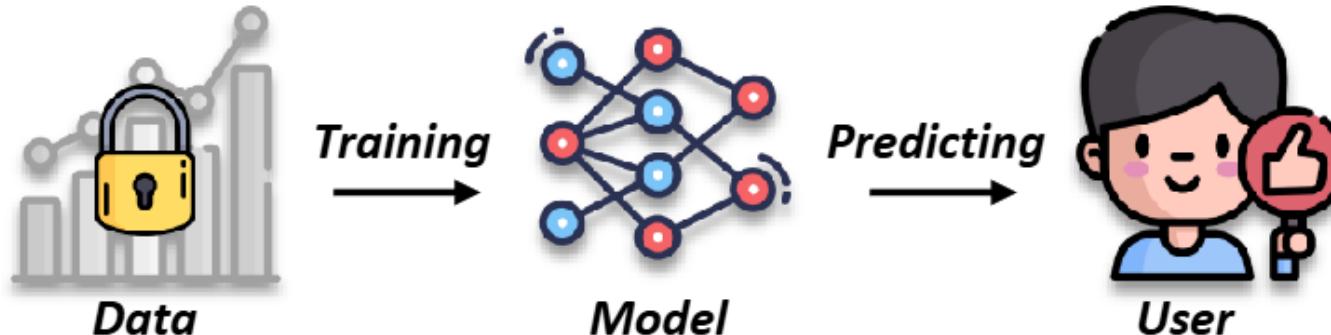
LLM-based RS (2020s-now)

Leveraging the powerful reasoning capabilities of **large language models** for recommendations.

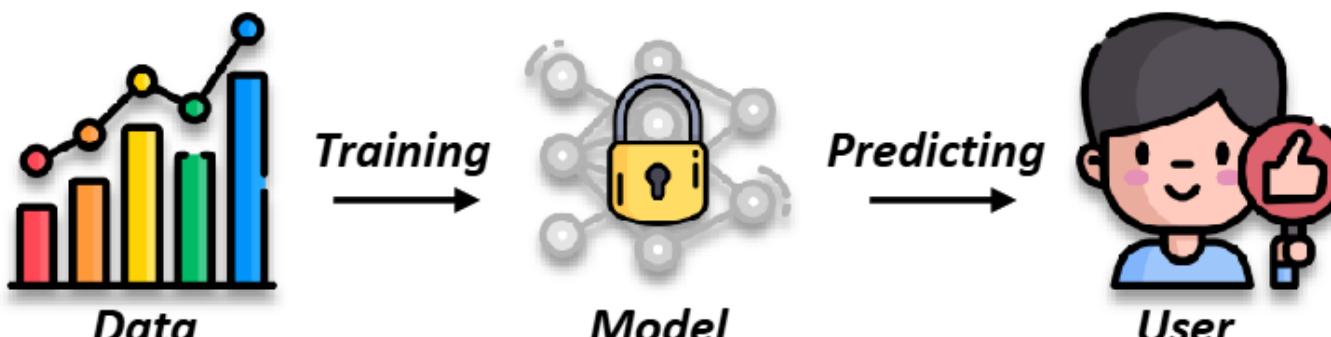


1. Introduction

Model-Centric RS to Data-Centric RS



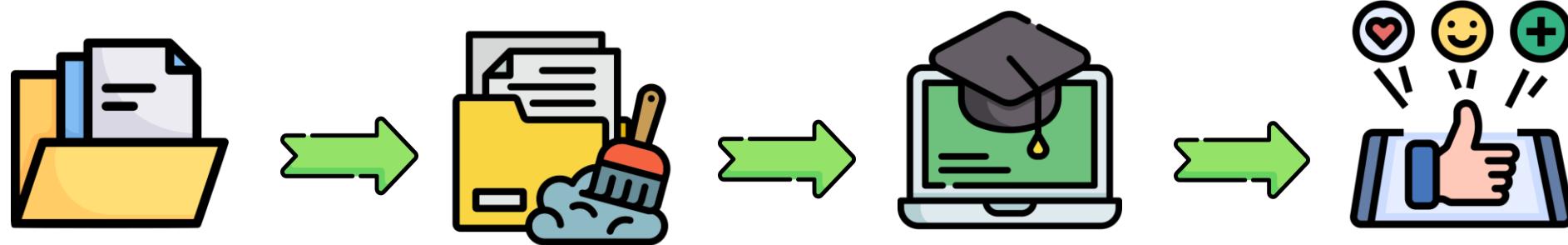
Model-Centric Recommender System: Refine Recommendation Model



Data-Centric Recommender System: Enhance Recommendation Data

1. Introduction

Data Issues Appear in Data-Centric RS Pipeline



Data Collection

Data Preprocessing

Data Training

Data Monitoring



Data Integrity



Data Privacy

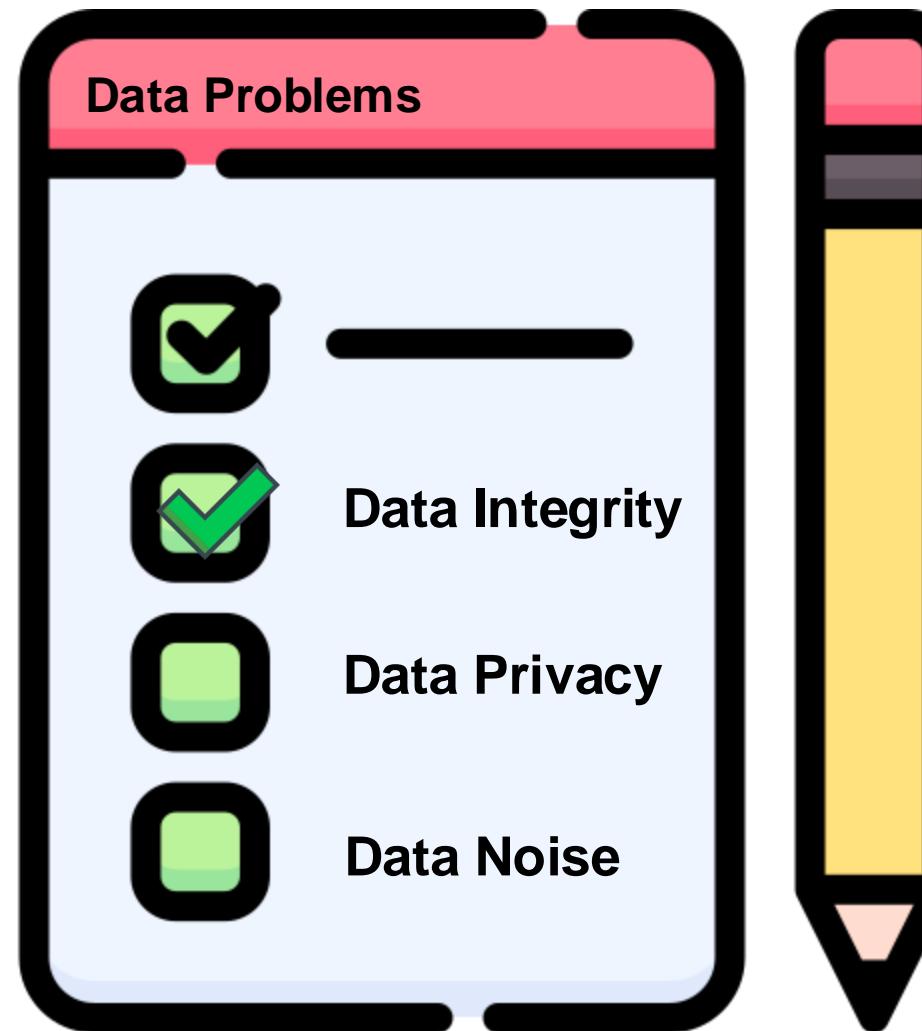


Data Noise

1. Introduction



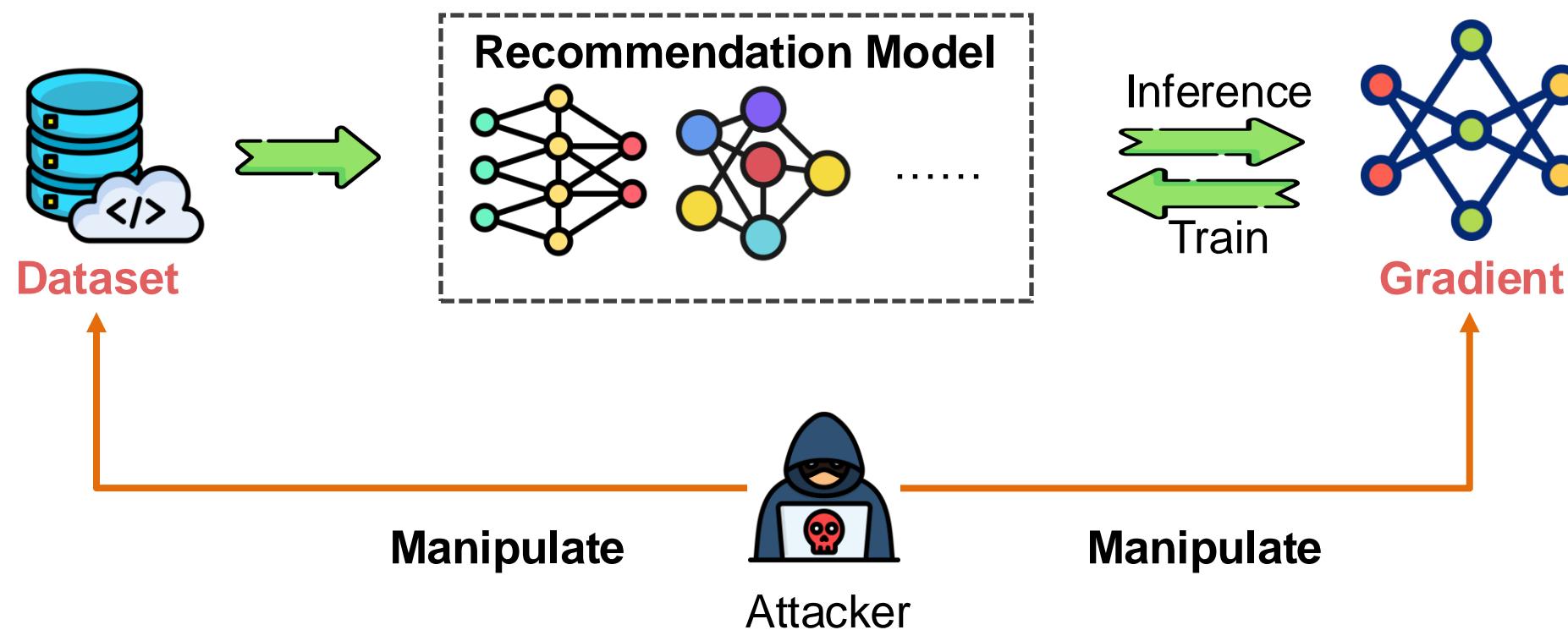
Data Integrity



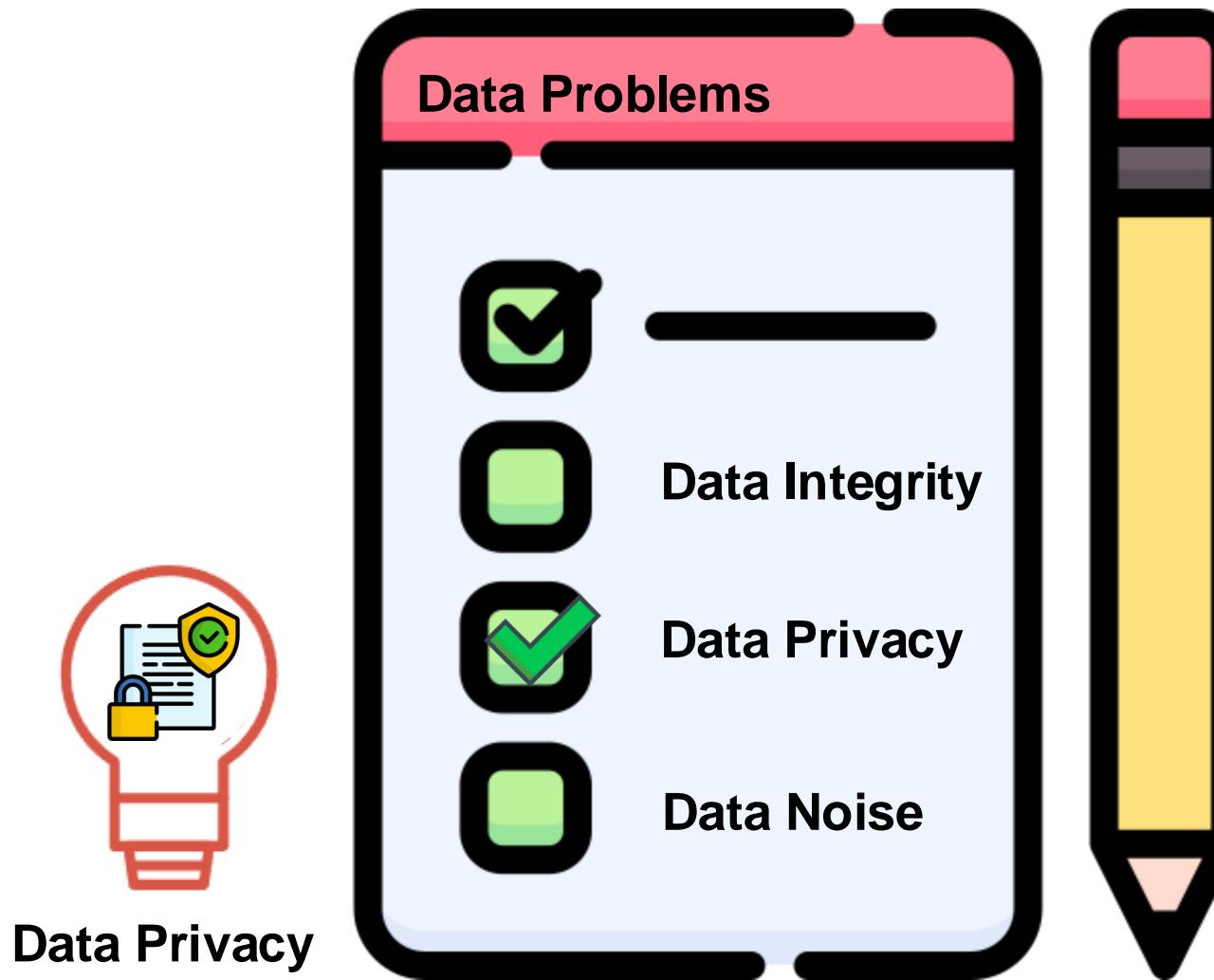
1. Introduction

Data Integrity Problem

Attackers target this data by corrupting training datasets or manipulating model gradients, ultimately distorting recommendation results.



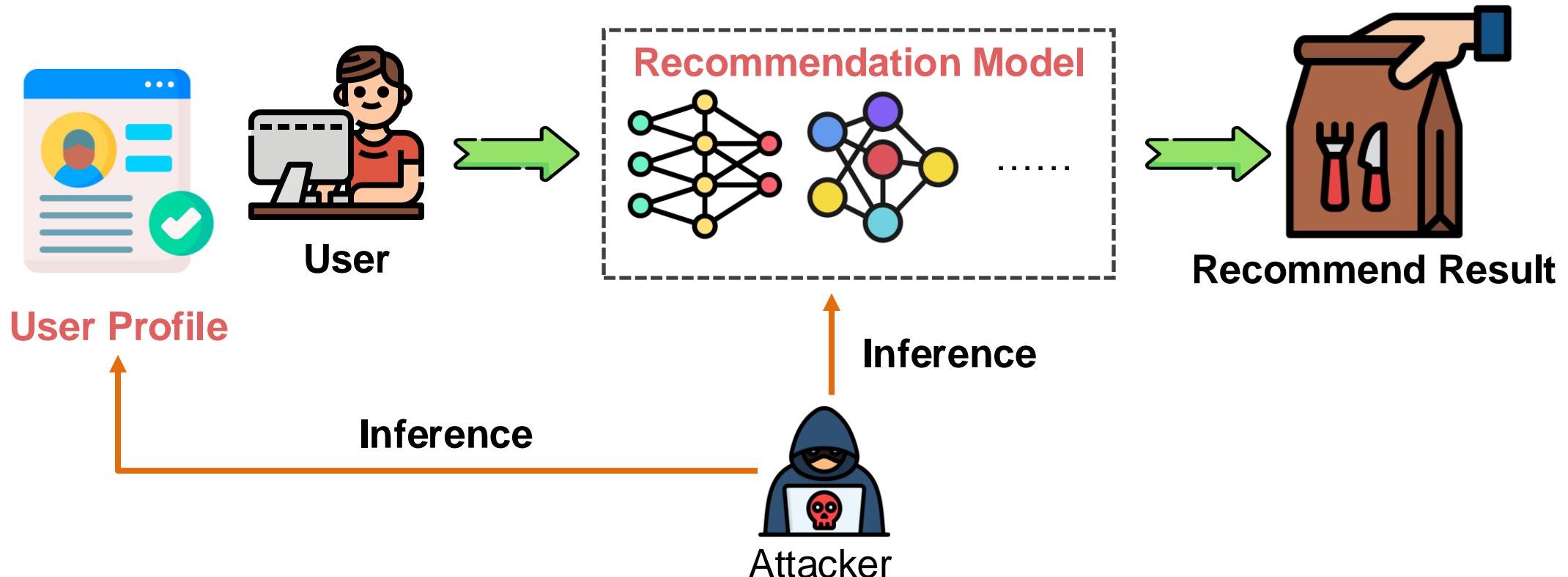
1. Introduction



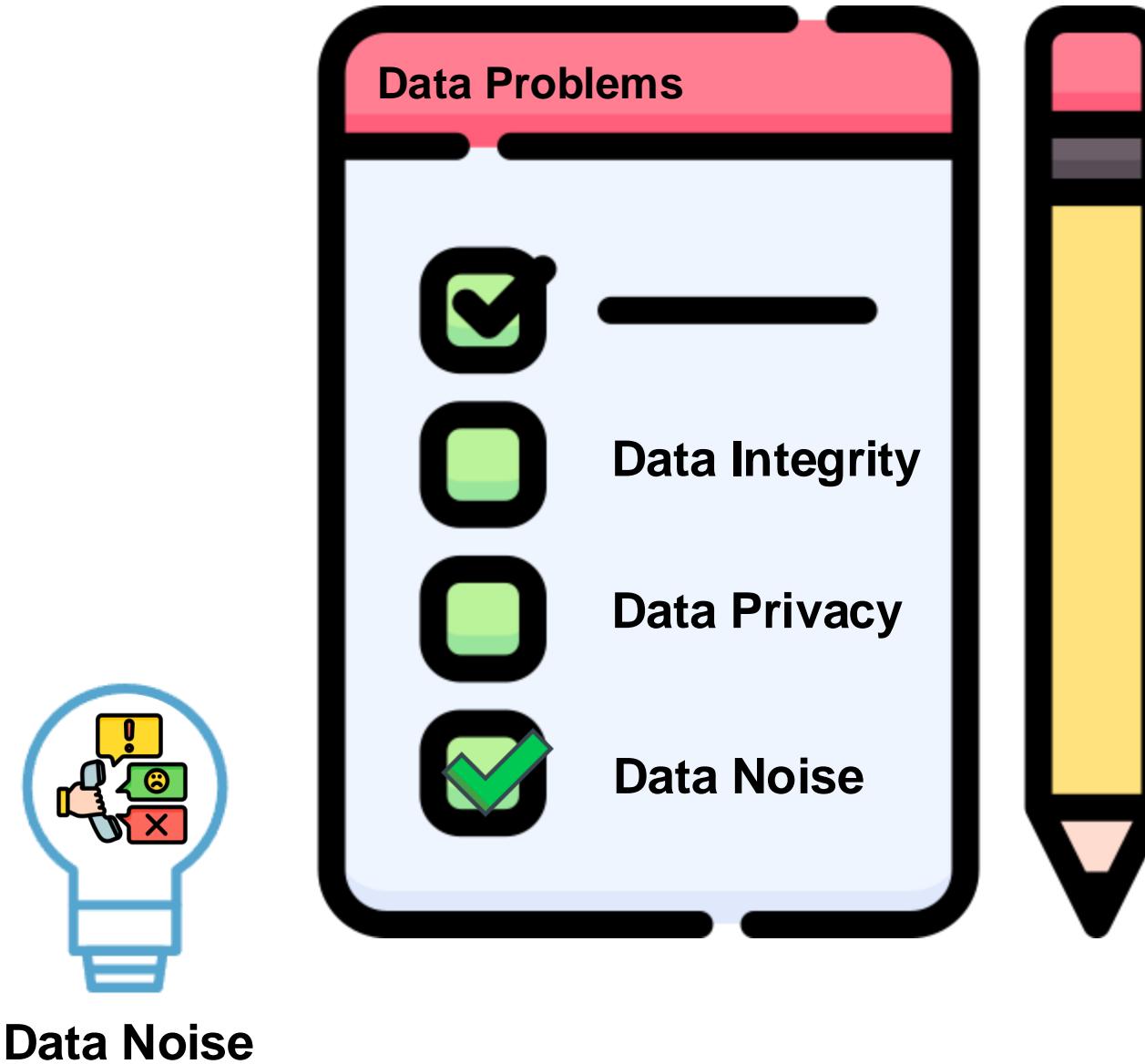
1. Introduction

Data Privacy Problem

Attackers leverage the extensive data generated by recommender systems to uncover sensitive information.



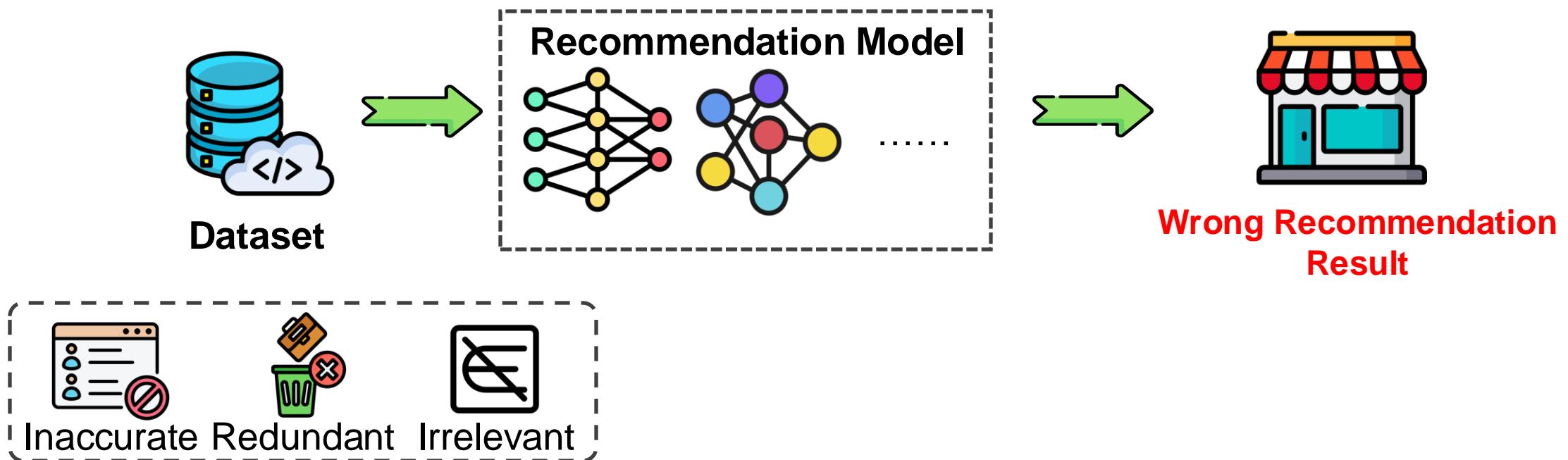
1. Introduction



1. Introduction

Data Noise Problem

Noise in data, caused by inaccurate, redundant, or irrelevant information, can severely degrade system performance.



1. Introduction

2. Securing Data Integrity

3. Preserving Data Privacy

4. Managing Data Noise

5. Limitations and Opportunities

6. Toolkit

Key Dimensions of How Attackers Manipulate Data Integrity

How Defender Protect Recommender System

2.1 Manipulating Data Integrity

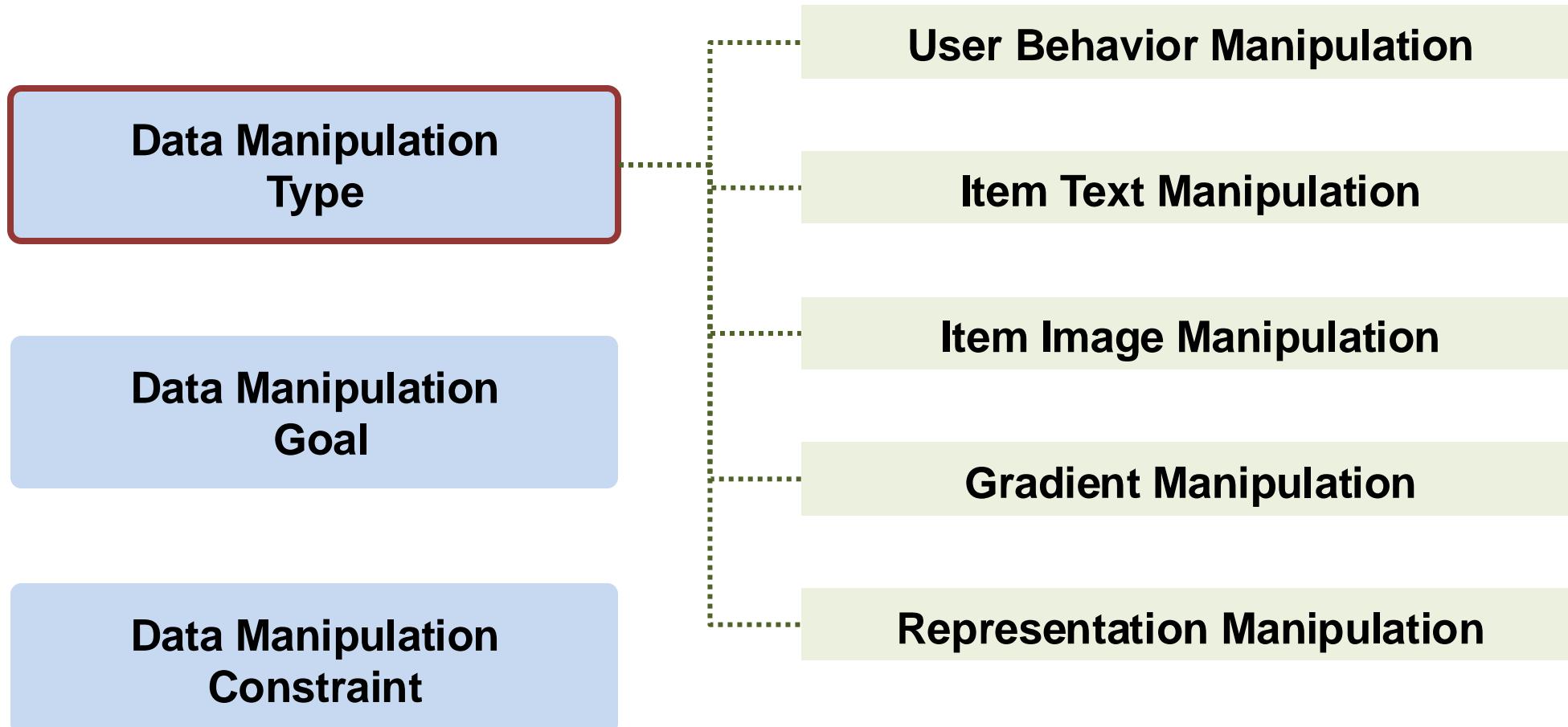
Key Dimensions of How Attackers Manipulate Data Integrity

**Data Manipulation
Type**

**Data Manipulation
Goal**

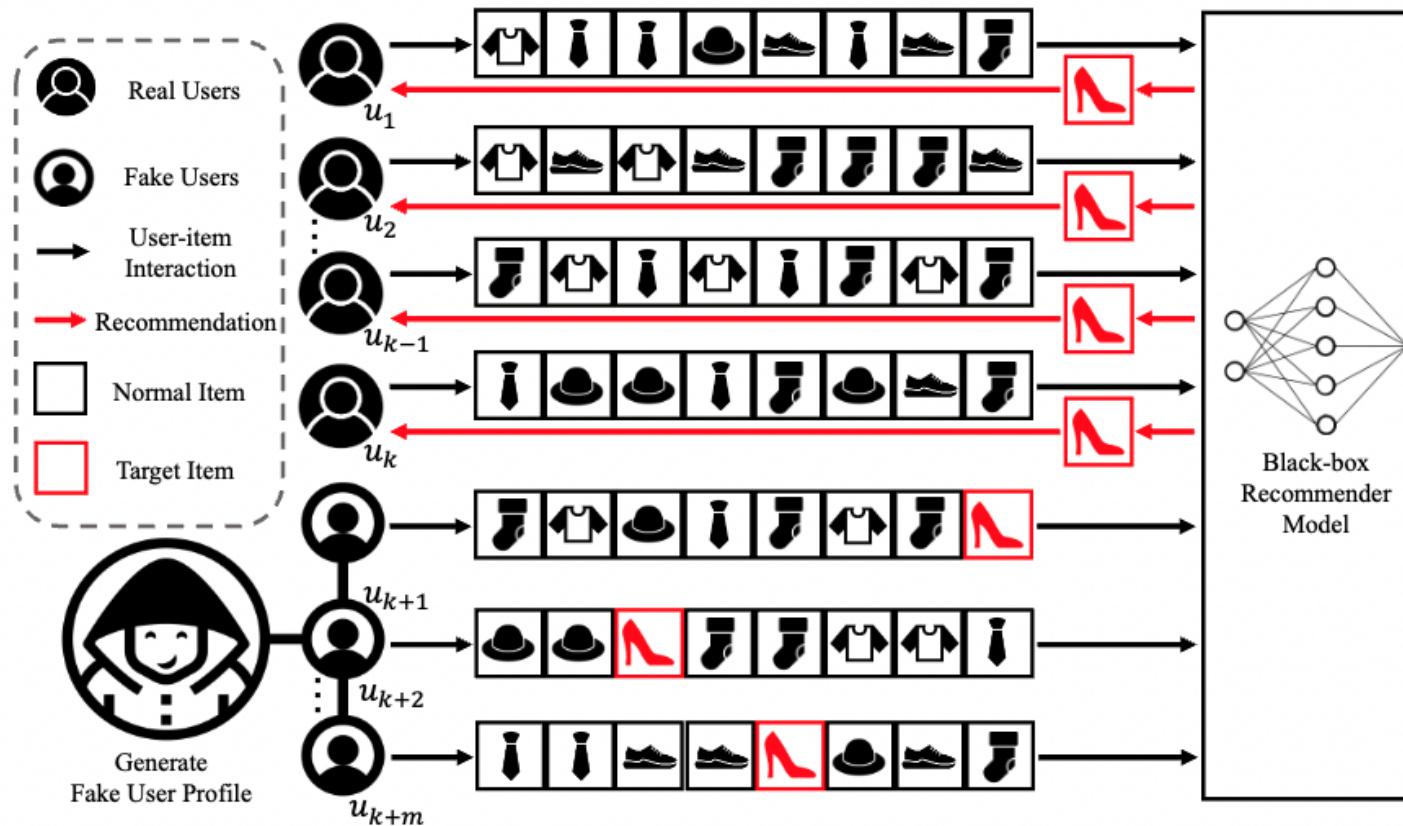
**Data Manipulation
Constraint**

2.1 Manipulating Data Integrity



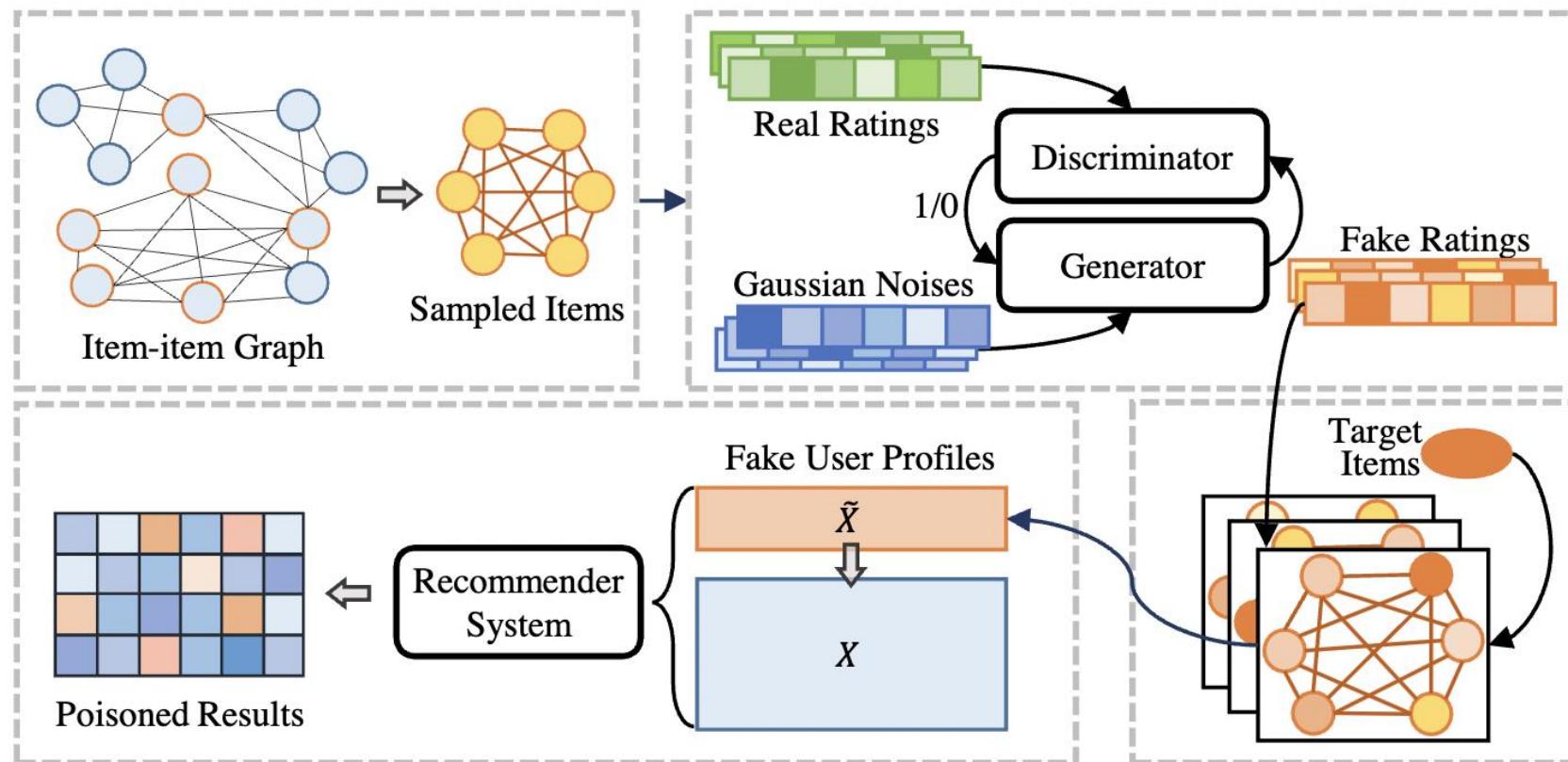
2.1.1 User Behavior Manipulation

User Behavior Manipulation: Attackers manipulates real user accounts and inject fake behaviors (clicks, purchases, ratings).



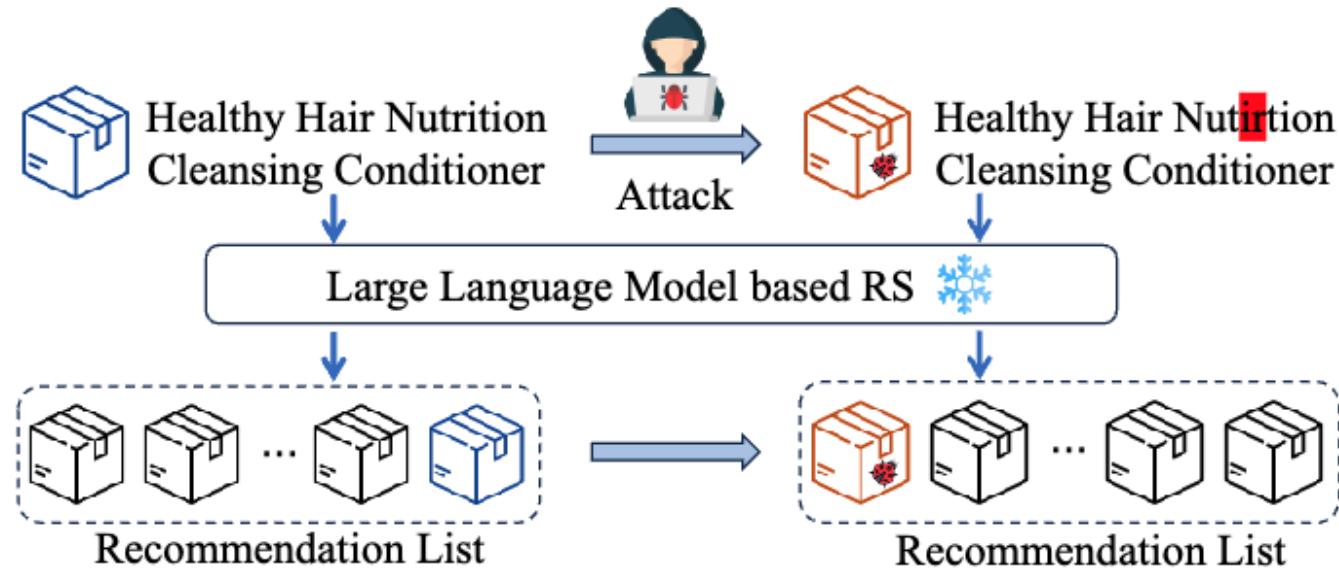
2.1.1 User Behavior Manipulation

Example (Rating): GOAT performs poisoning attacks on the ratings of graph data.



2.1.1 Item Text Manipulation

Item Text Manipulation: Attackers modify chars or sentences of the description of items.



2.1.1 Item Text Manipulation

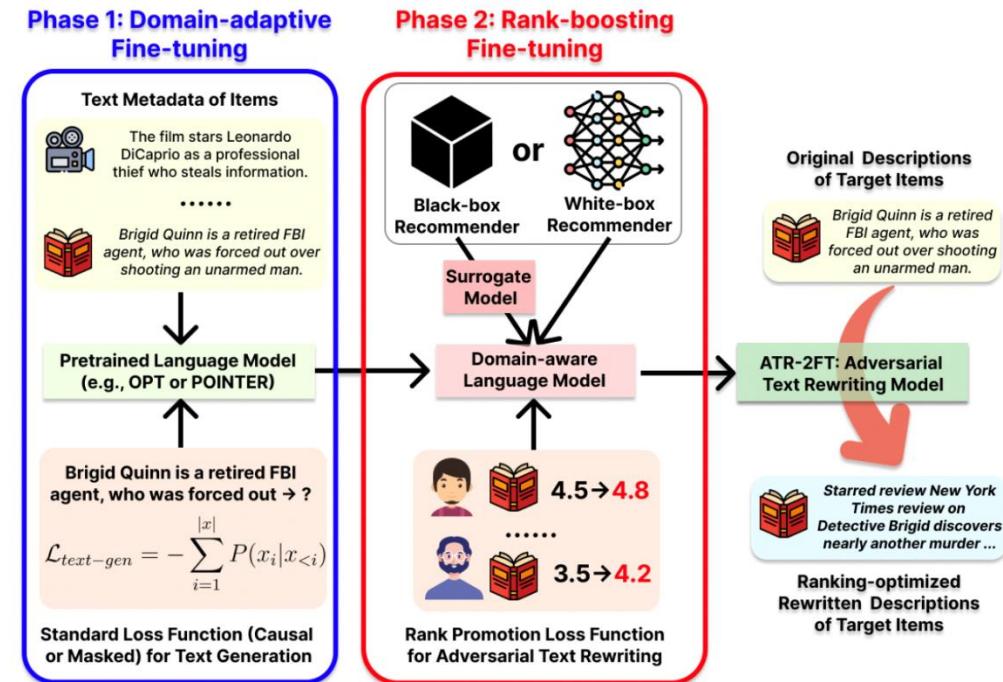
Example (Rewriting): TextRecAttack and ATR2FT employ text rewriting attack on text-aware recommendation.

Prompt 1: You are a marketing expert that helps to promote the product selling. Rewrite the product title in <MaxLen> words to keep its body the same but more attractive to customers: <ItemTitle>.

Prompt 2: Here is a basic title of a product. Use your creativity to transform it into a catchy and unique title in <MaxLen> words that could attract more attention: <ItemTitle>.

Prompt 3: Rewrite this product's title by integrating positive and appealing words, making it more attractive to potential users without altering its original meaning (in <MaxLen> words): <ItemTitle>.

TextRecAttack



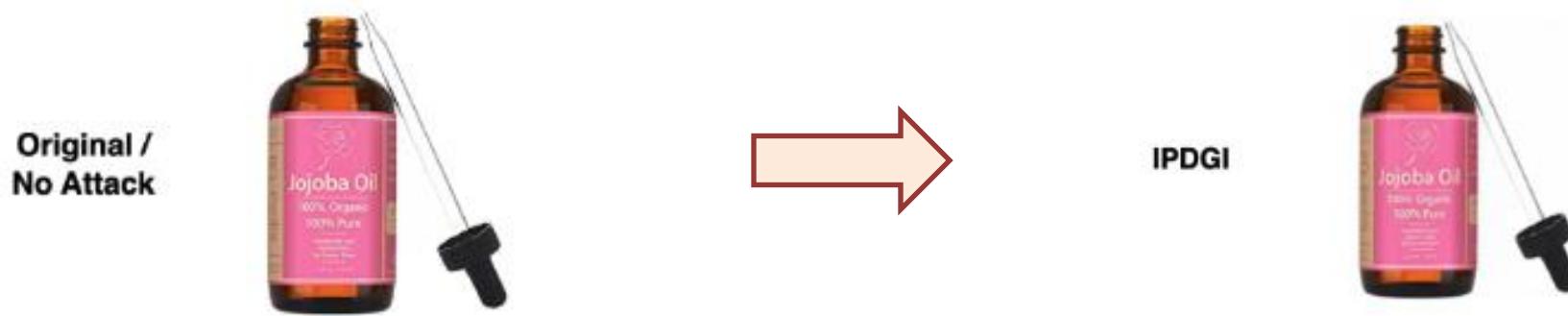
ATR2FT

Zhang, Jinghao, et al. "Stealthy attack on large language model based recommendation." ACL 2024.

Oh, Sejoon, et al. "Adversarial Text Rewriting for Text-aware Recommender Systems." Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. 2024.

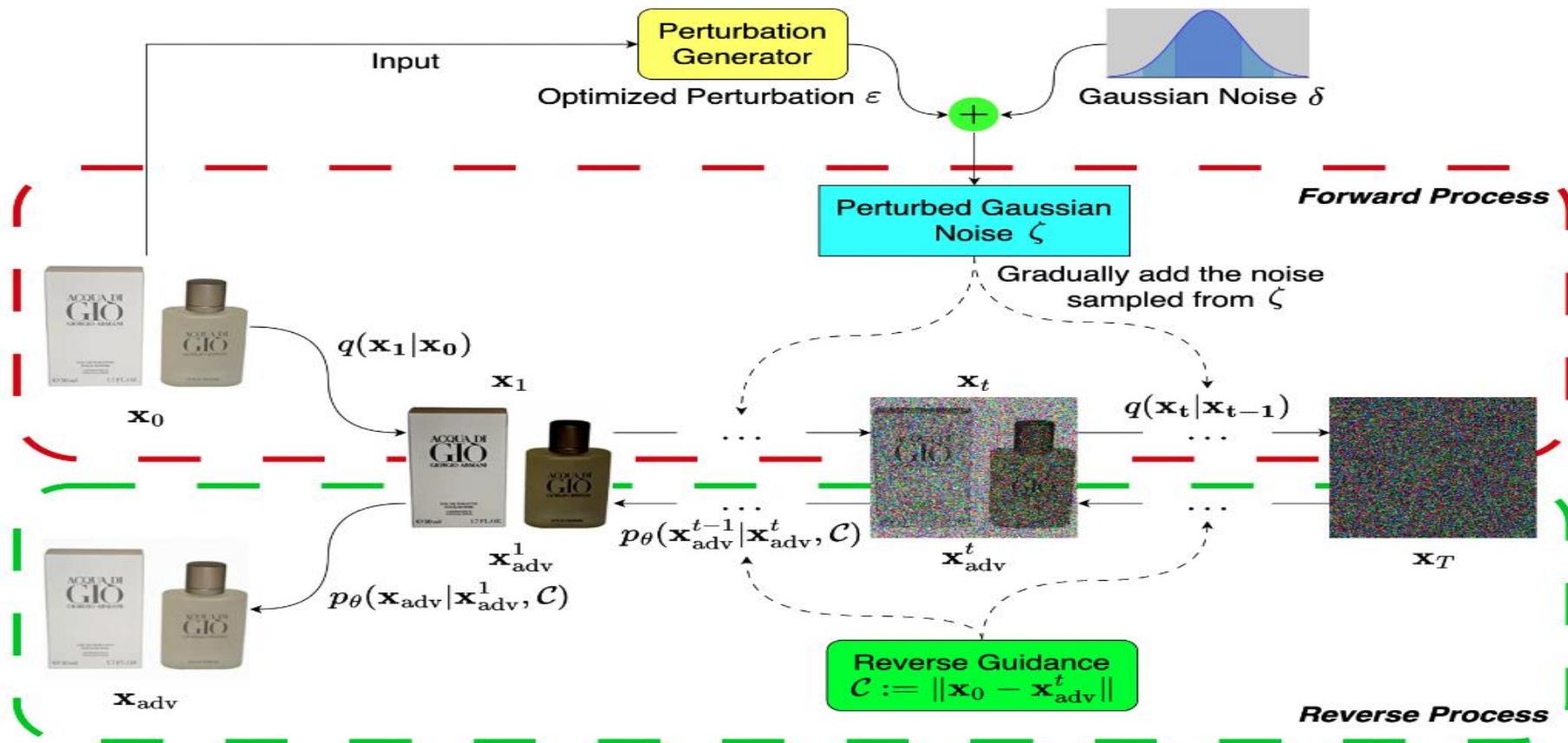
2.1.1 Item Image Manipulation

Item Image Manipulation: generates adversarial samples designed to promote the exposure rates of target items



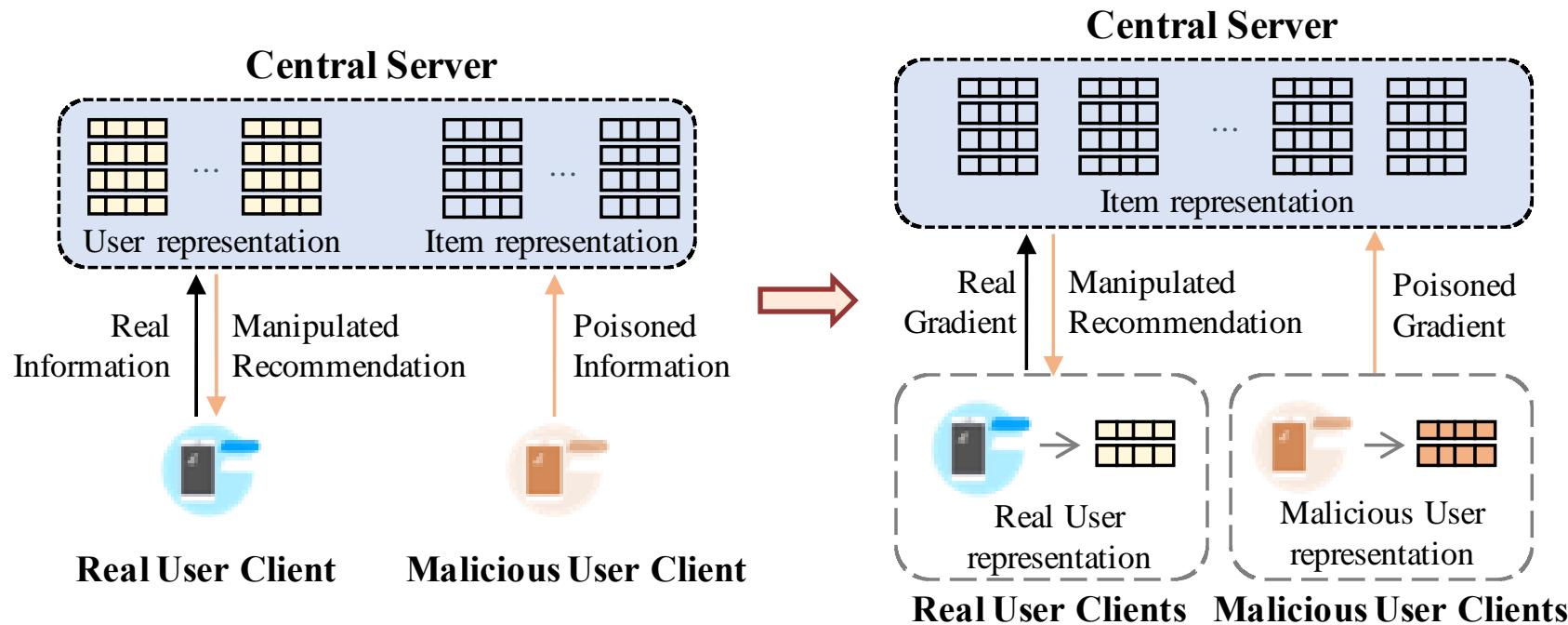
2.1.1 Item Image Manipulation

Example (Diffusion Model) : IPDGI generates adversarial samples designed to promote the exposure rates of target items



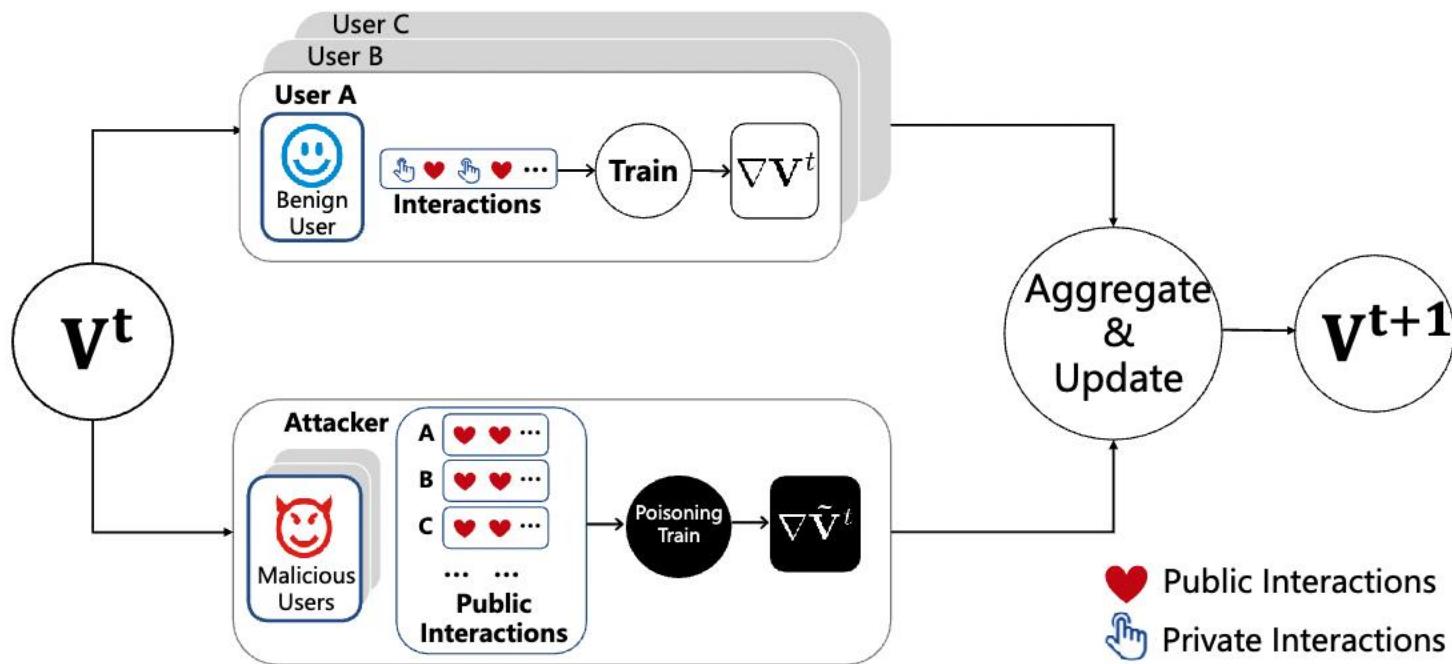
2.1.1 Gradient Manipulation

Gradient Manipulation: User Clients upload poisonous gradient into central server in decentralized recommendation (e.g., Federated RS) .



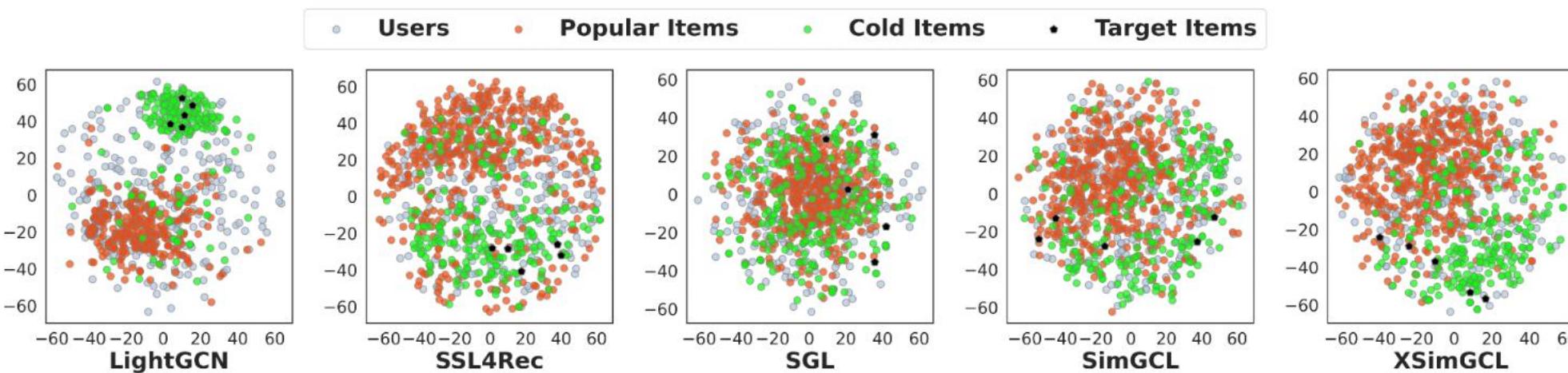
2.1.1 Data Manipulation Type

Example(Gradient Aggregation): FedRecAttack controls user clients uploads poisonous gradient into central server.



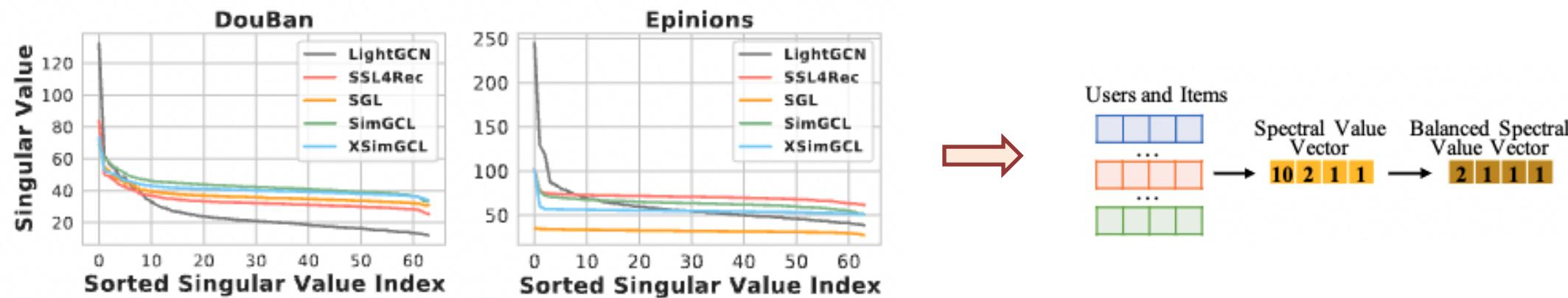
2.1.1 Representation Manipulation

Representation Manipulation: Attackers control the user/items embedding representation to achieve goal.

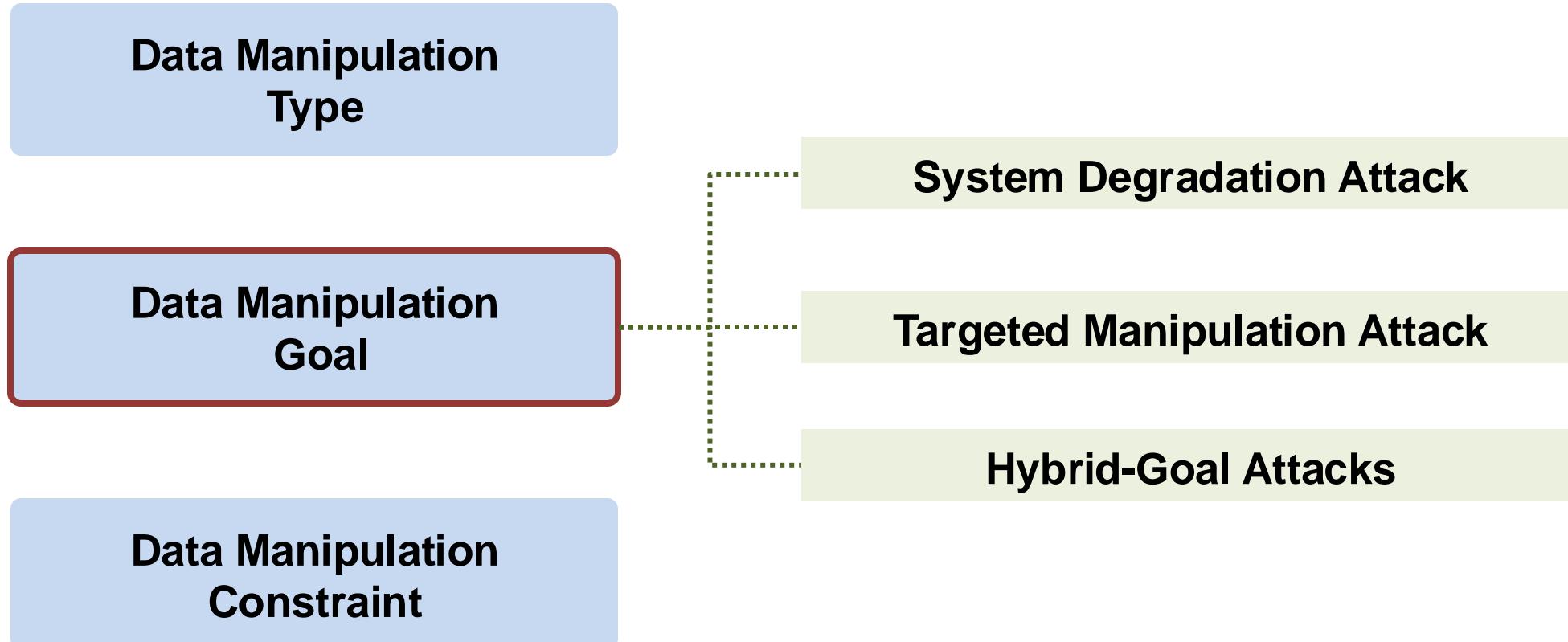


2.1.1 Representation Manipulation

Example (Representation Spectral Values): CLeaR controls the spectral value of user/items embedding representation into a smooth distribution.

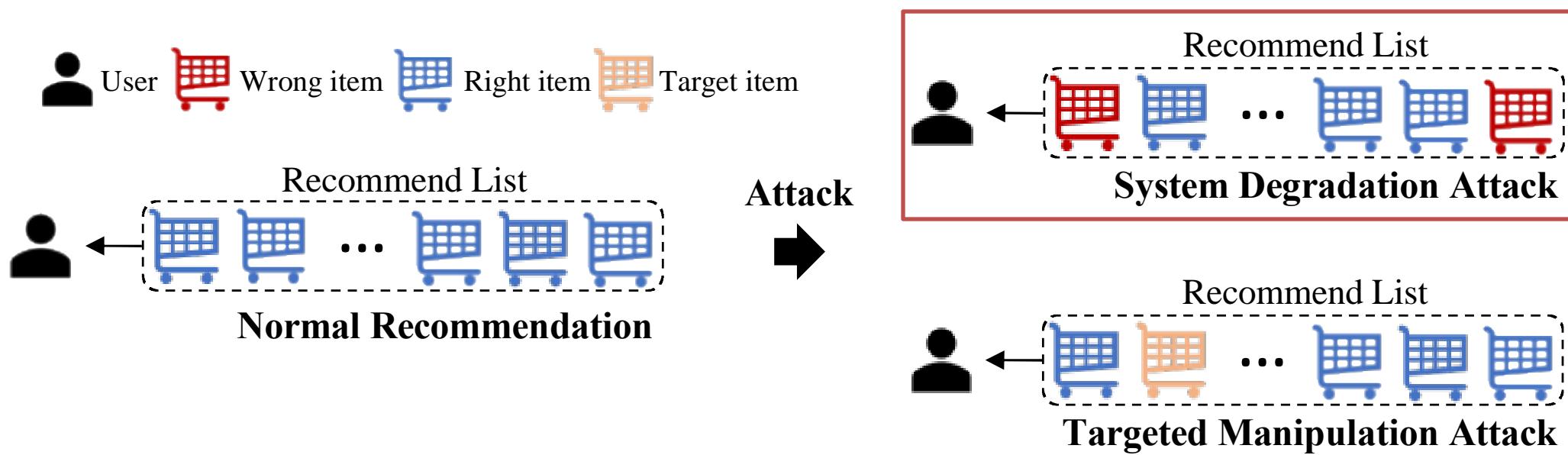


2.1 Manipulating Data Integrity



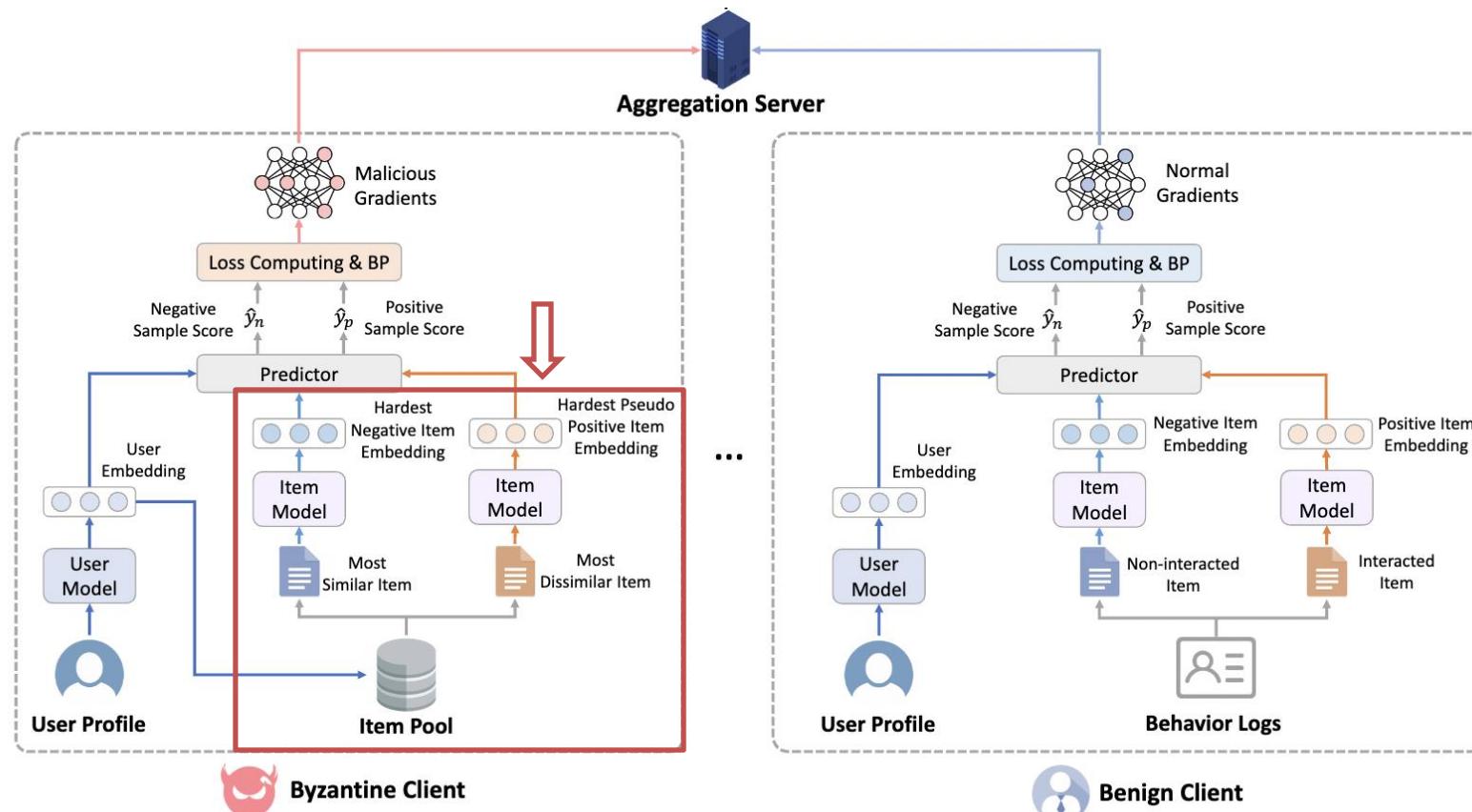
2.1.2 System Degradation Attack

System Degradation Attack (a.k.a untargeted attack), is crafted with the broad objective of undermining the entire system performance.



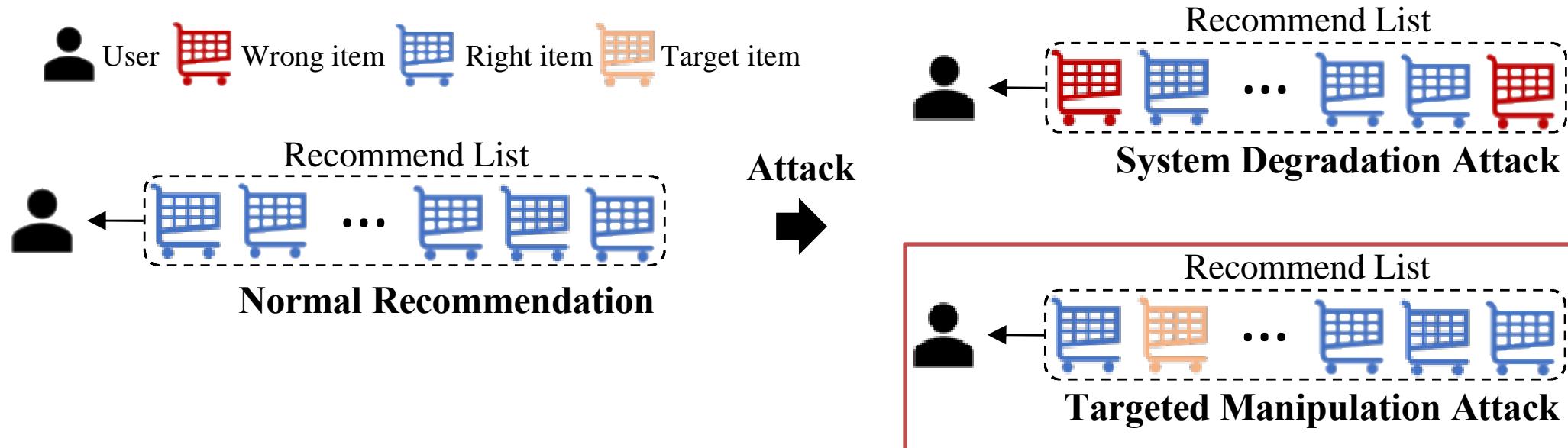
2.1.2 System Degradation Attack

Example (Hardest Sample) : FedAttack selects items most and least relevant to user embeddings as the hardest negative and positive samples.



2.1.2 Targeted Manipulation Attack

Targeted Manipulation Attack (a.k.a targeted attack) is elaborately crafted to either promote or demote specific items within distinct user groups or across all users.



2.1.2 Targeted Manipulation Attack

Two Common Ways of Targeted Manipulation Attacks

Example1 (Visibility of Target Items): **PoisonRec** focuses on maximizing the frequency of targeted items appearing in top-K recommendation lists.

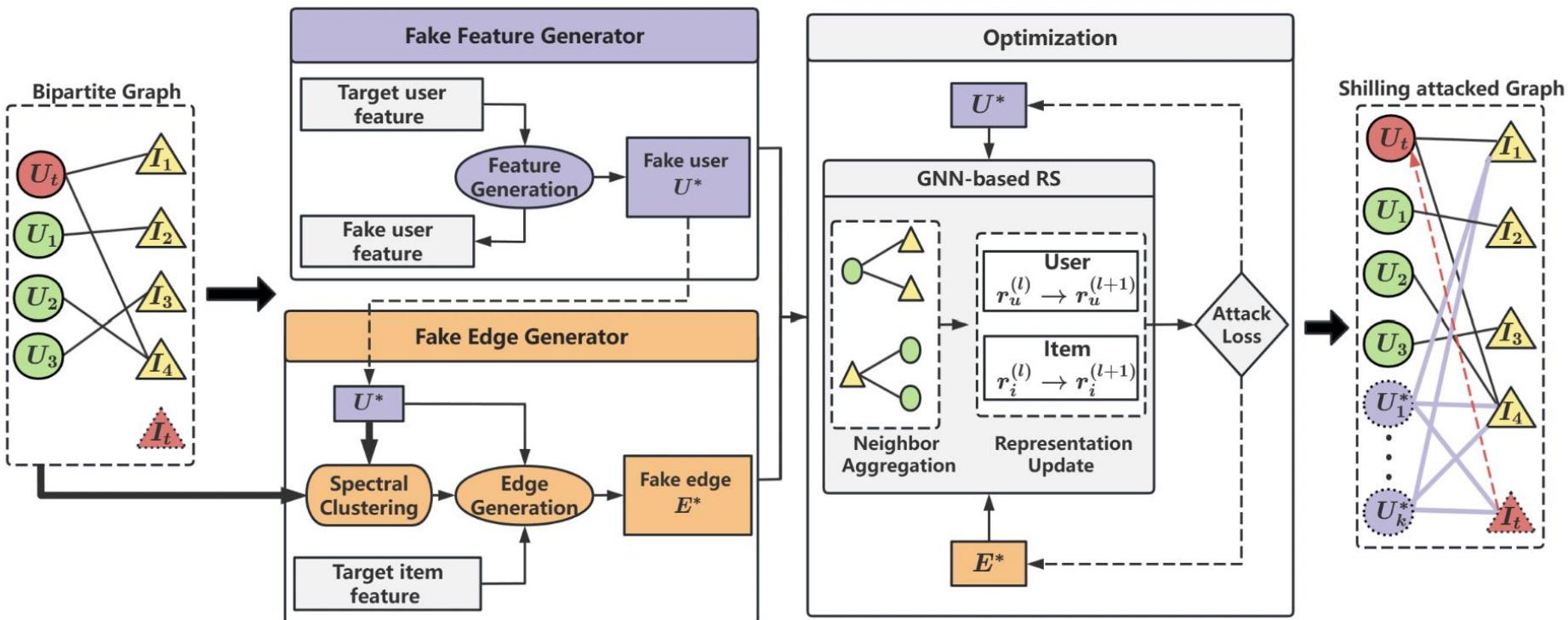
Example2 (Close to User Preferred Items): **GTA** evaluates user intent by predicting the user's most preferred item and then adjusts the target item to closely resemble these favored items.

Song, Junshuai, et al. "Poisonrec: an adaptive data poisoning framework for attacking black-box recommender systems." 2020 IEEE 36th international conference on data engineering (ICDE). IEEE, 2020.

Wang, Zhiye, et al. "Revisiting data poisoning attacks on deep learning based recommender systems." 2023 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2023.

2.1.2 Targeted Manipulation Attack

Example (Close to Targeted User Group): AutoAttack crafts malicious profiles that closely mimic the targeted user group. It enables an influence on the designated group while concurrently minimizing effects on other users.



2.1.2 Hybrid-Goal Attacks

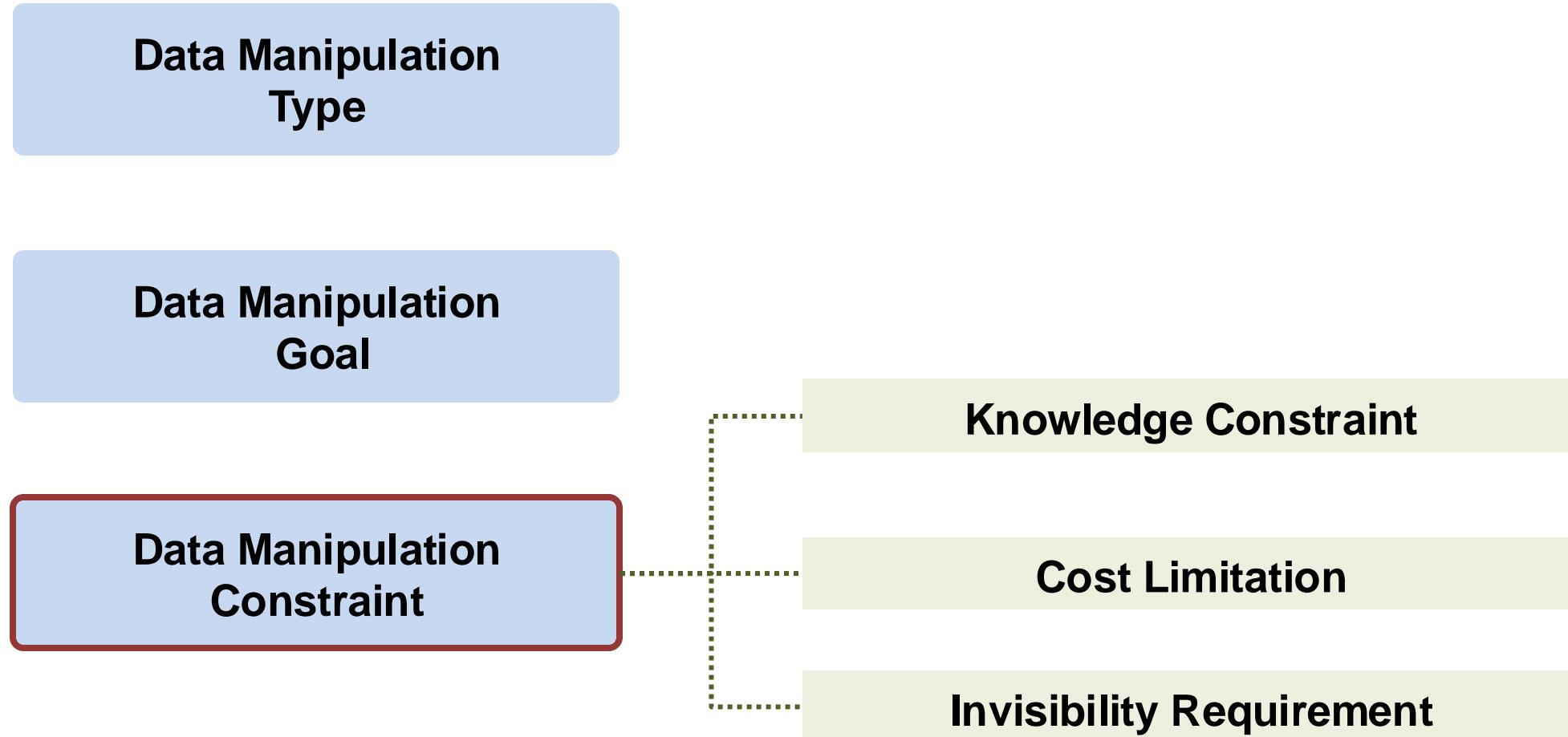
Hybrid-Goal Attack: Some attacks achieve a balance between these two goals using weighted coefficients, such as **SGLD**, **CD-Attack**.



$$R = \alpha R_{\text{target}} + \beta R_{\text{untarget}}$$

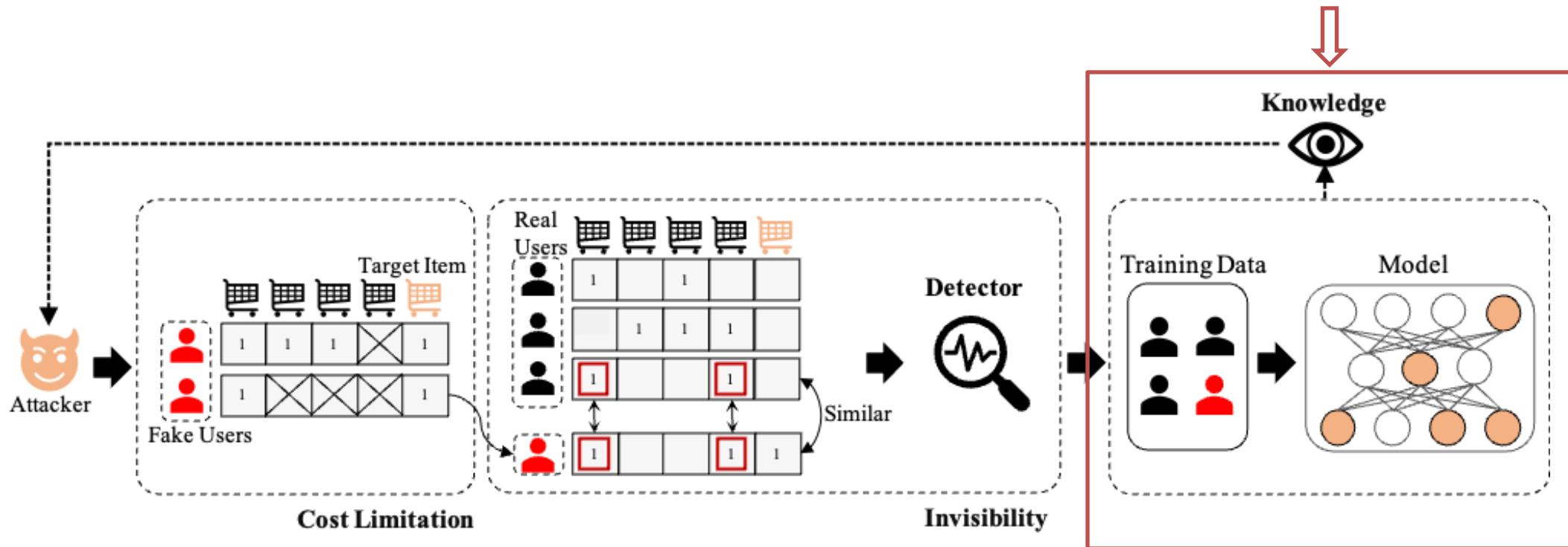
The potential for hybrid attacks has not been fully explored, which does not fully represent the diversity of real-world attackers' intentions.

2.1 Manipulating Data Integrity



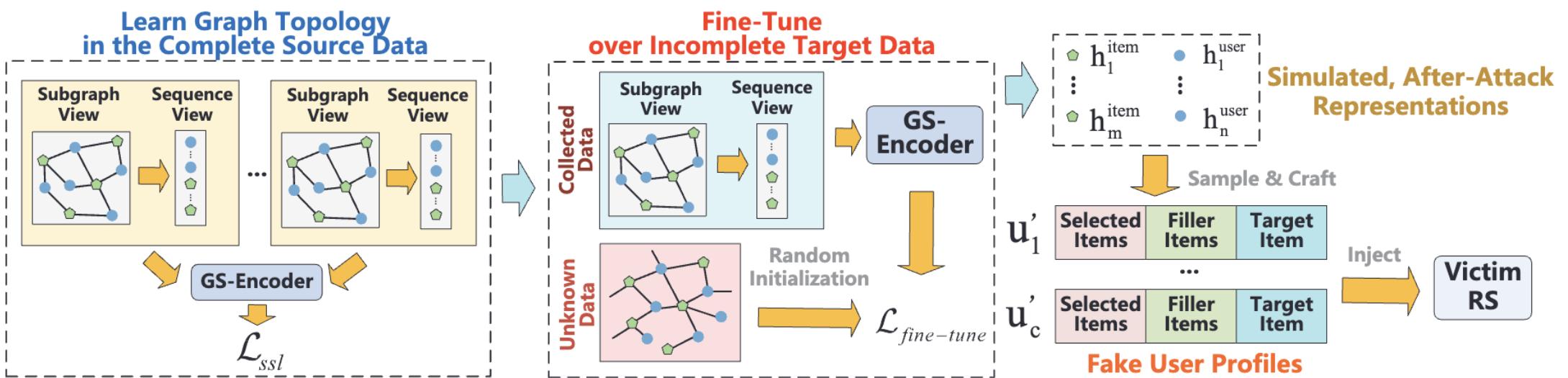
2.1.3 Knowledge Constraint

Knowledge Constraint: Attackers typically view RS as complex 'black boxes,' with limited transparency regarding the data used for training and the specifics of the underlying algorithms.



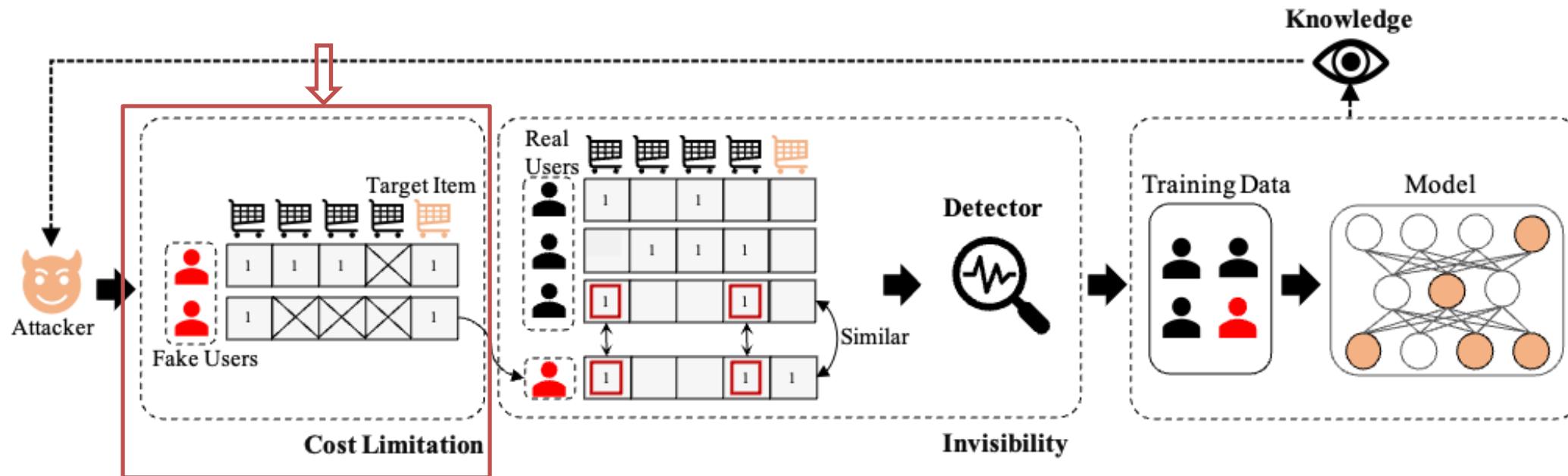
2.1.3 Knowledge Constraint

Example (Lack of Data Knowledge): PCAttack pre-trains a simulator model on multiple data sources. A small portion of the target data is then input into the simulator model for finetuning.



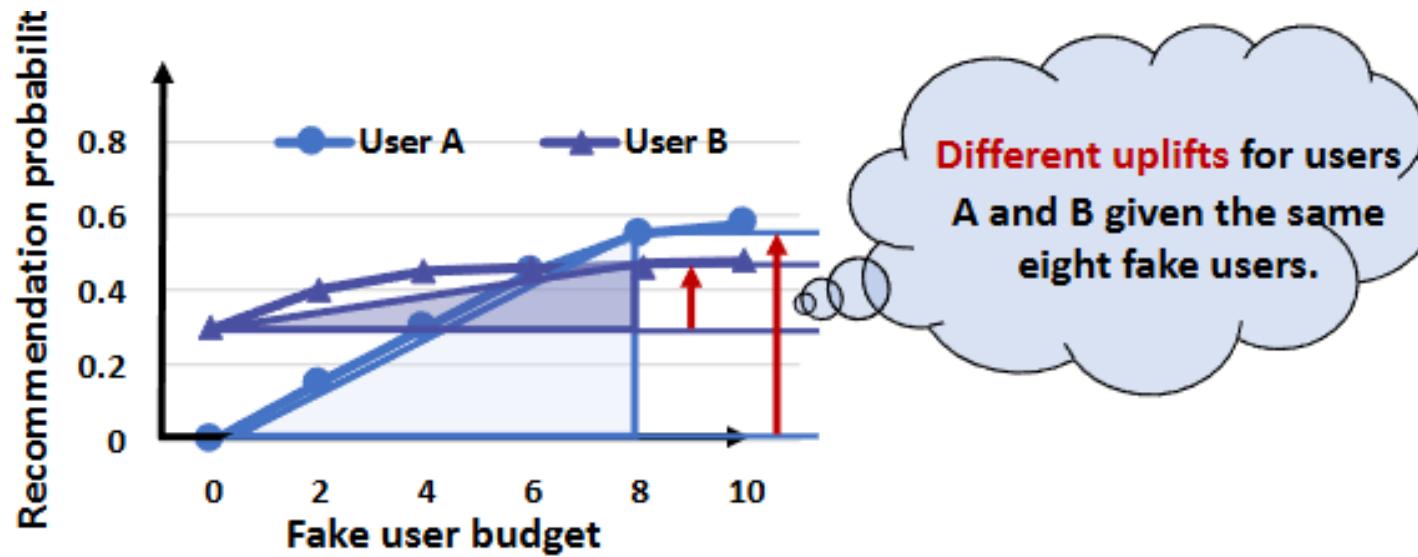
2.1.3 Cost Limitation

Cost Limitation: The insertion of poisoned data into the system requires a cost-effective analysis to ensure economic viability.



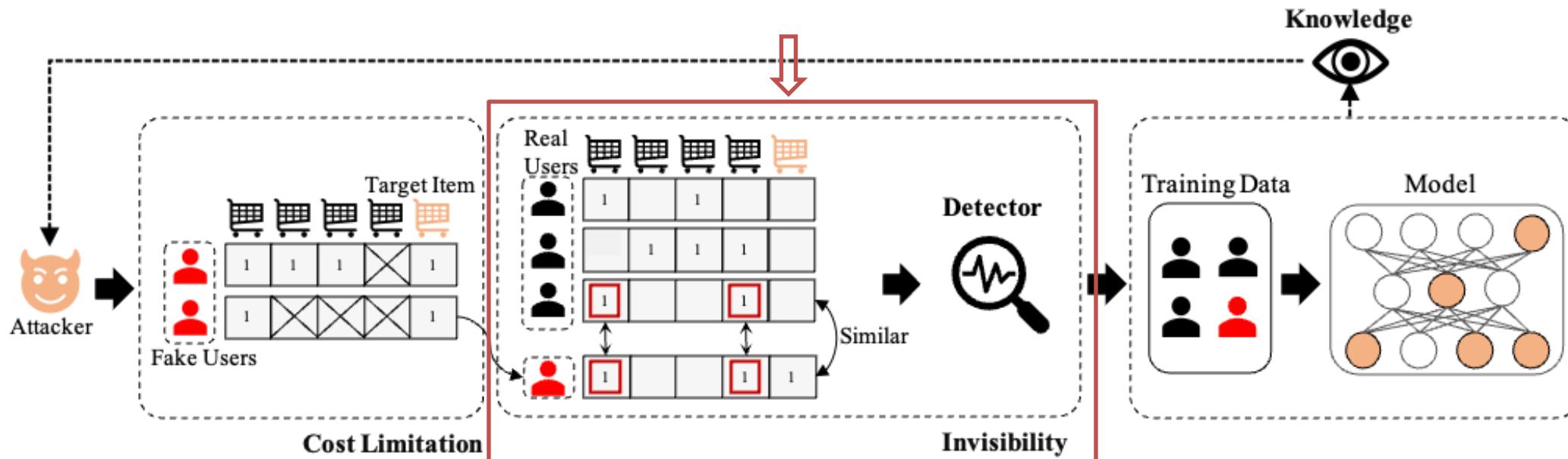
2.1.3 Cost Limitation

Example (Injection Budgets): UBA uses uplift model to estimate the treatment effect on each target user and optimizes the allocation of fake user budgets to maximize the attack performance.



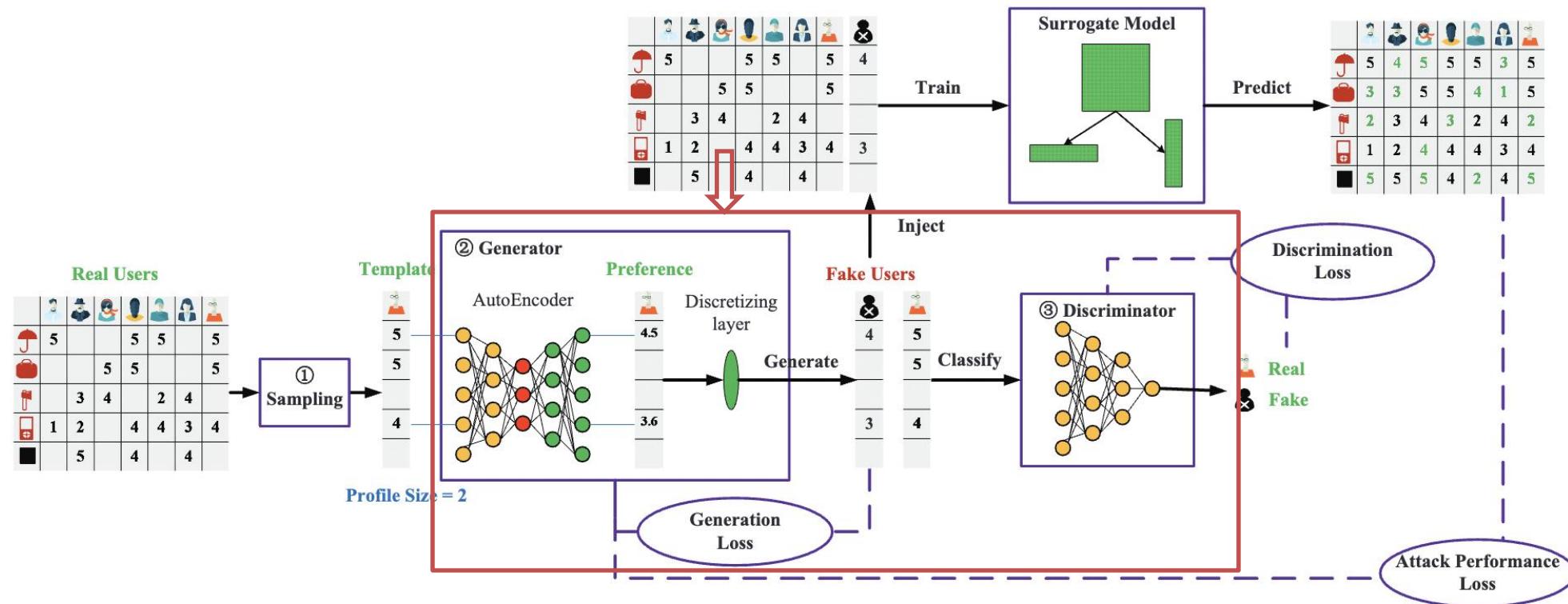
2.1.3 Invisibility Requirement

Invisibility Requirement: Poisoned data should be intricately crafted to blend seamlessly with legitimate data, making it challenging for the typical defense mechanisms within RS to detect the anomalies.



2.1.3 Invisibility Requirement

Example (GANS) : AUSH uses generative adversarial networks enhance the resemblance of generated user profiles to real user profiles.



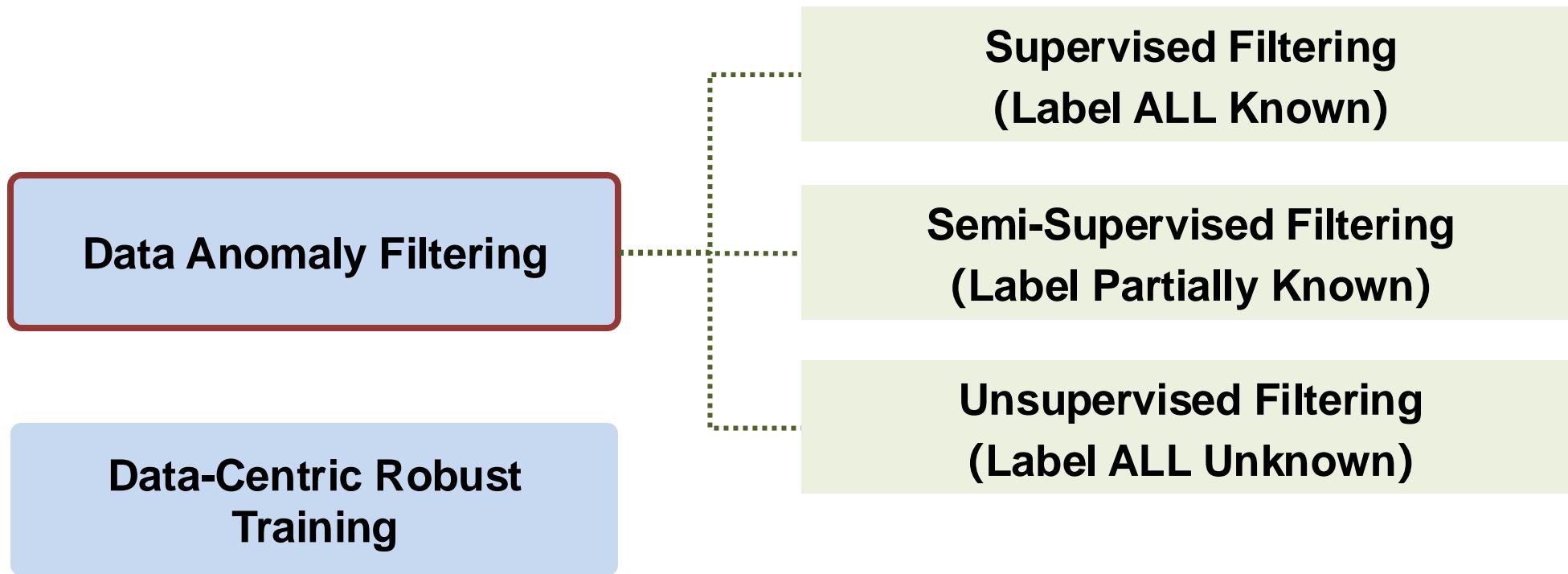
2.2 Data Integrity Protection

Two Stages of Data Integrity Protection

Data Anomaly Filtering

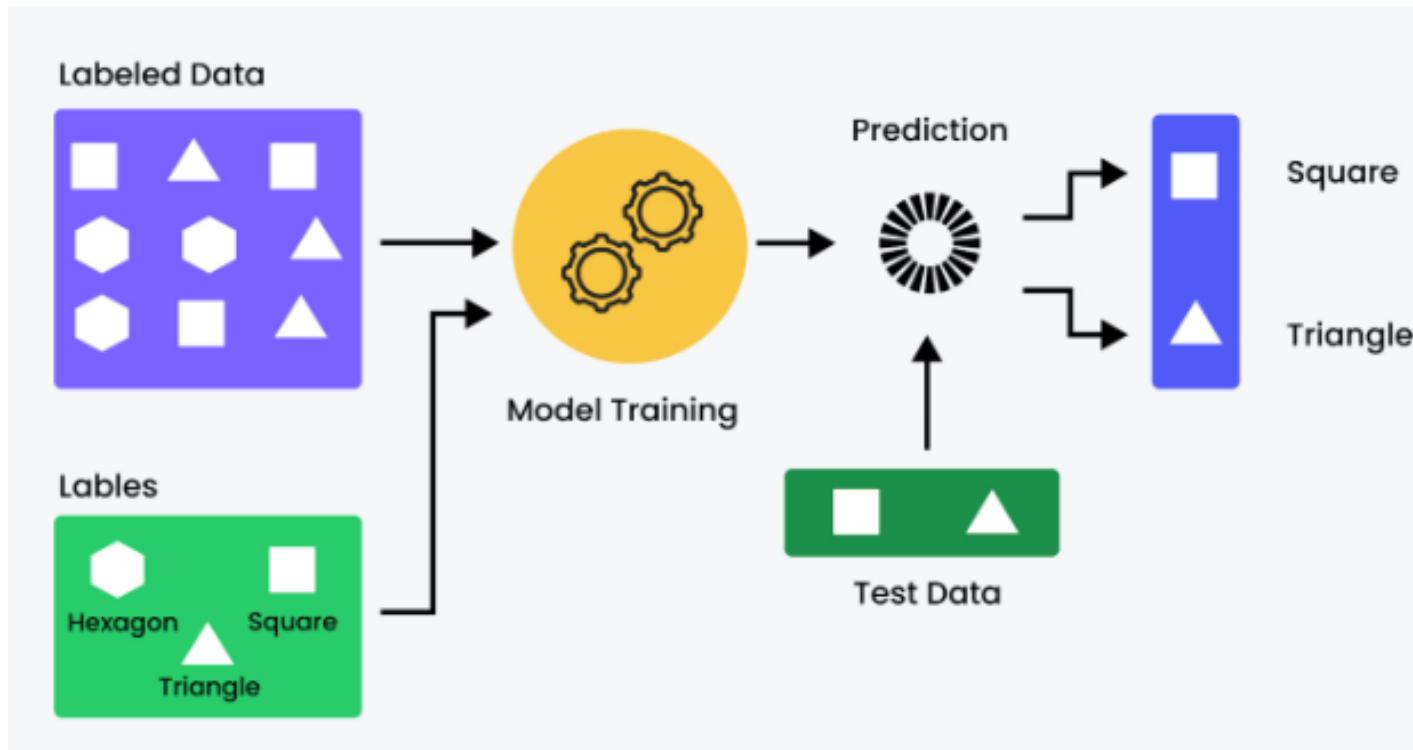
**Data-Centric Robust
Training**

2.2 Protecting Data Integrity



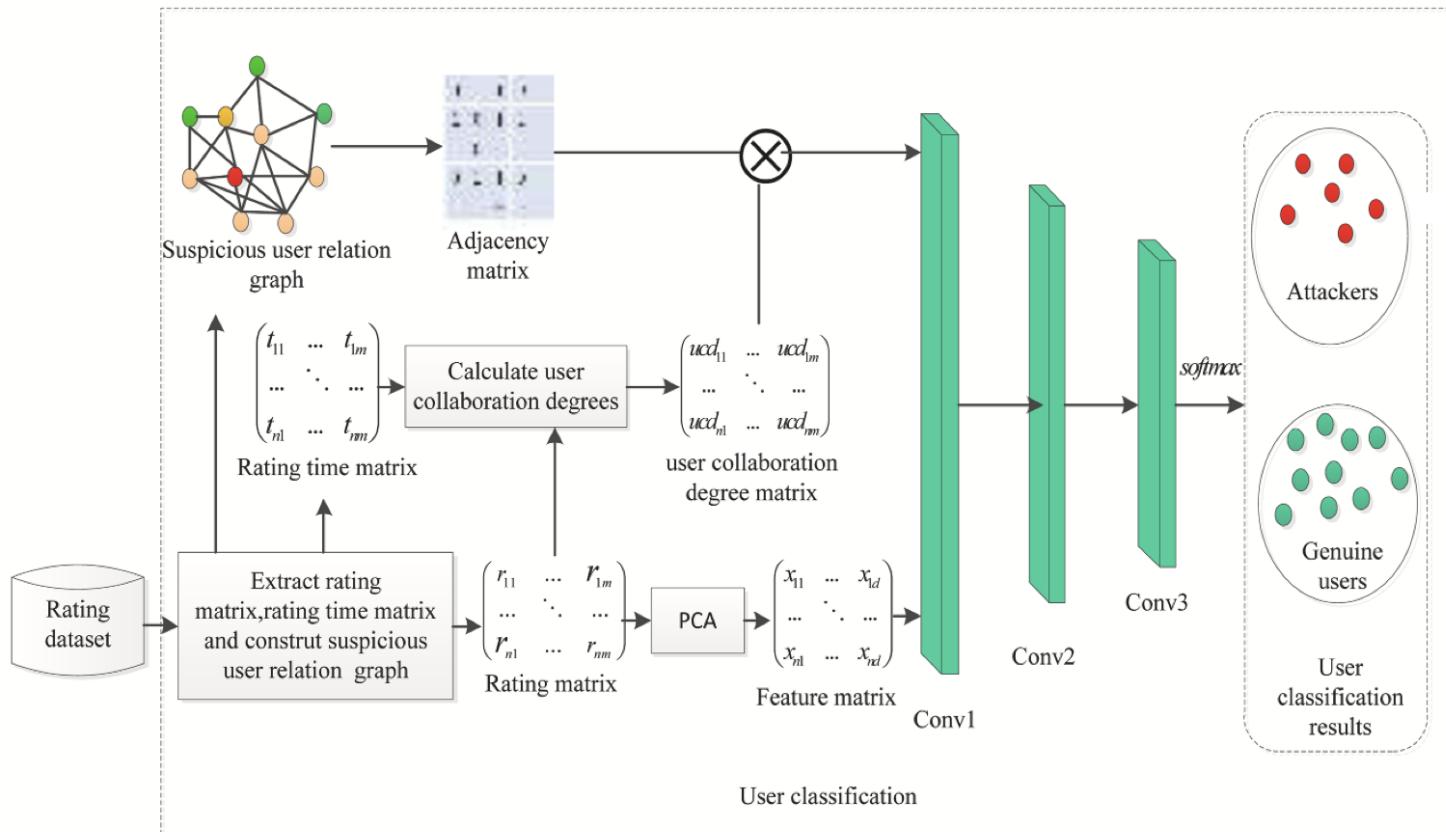
2.2.1 Supervised Filtering

Supervised Filtering (Label ALL Known): Defenders leverage labeled historical data to train models that can discern and isolate malicious entries based on learned patterns.



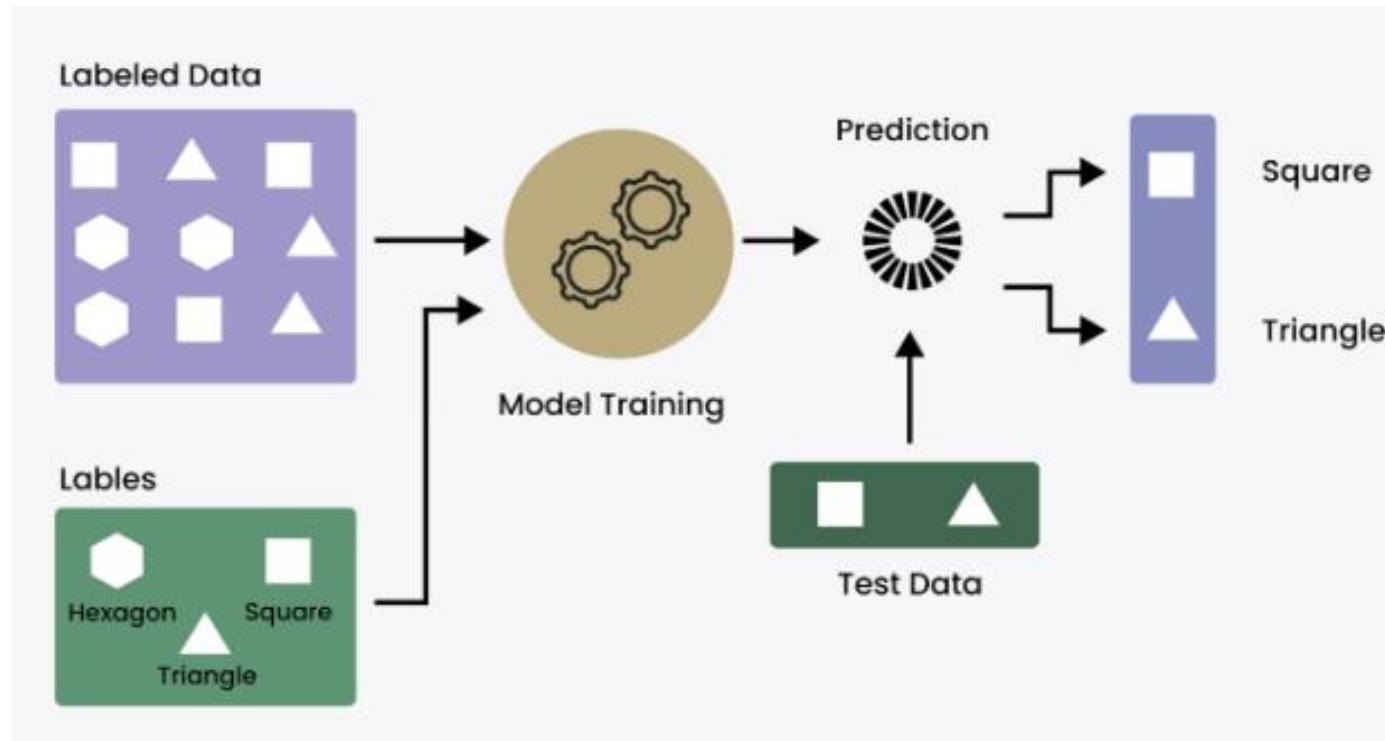
2.2.1 Supervised Filtering

Example (GCN): DHAGCN combines user behavior features with GCN to detect hybrids of model-generative and group attacks.



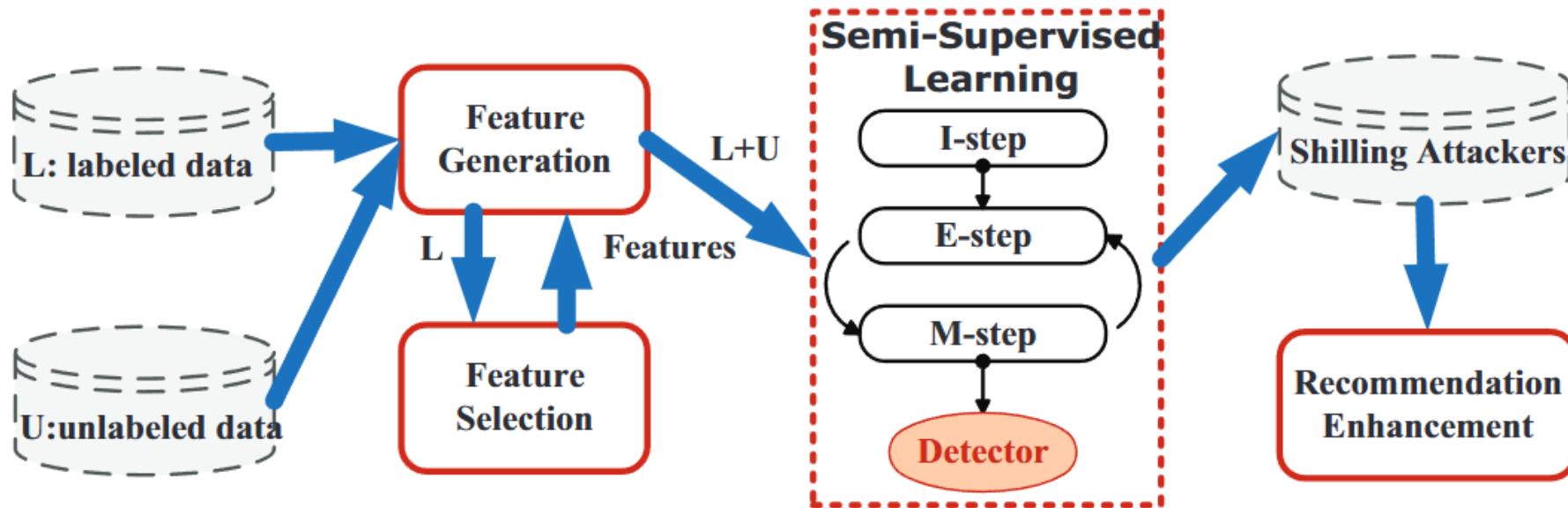
2.2.1 Semi-Supervised Filtering

Semi-Supervised Filtering (Label Partially Known): Defenders leverage labeled historical data to train models that can discern and isolate malicious entries based on learned patterns.



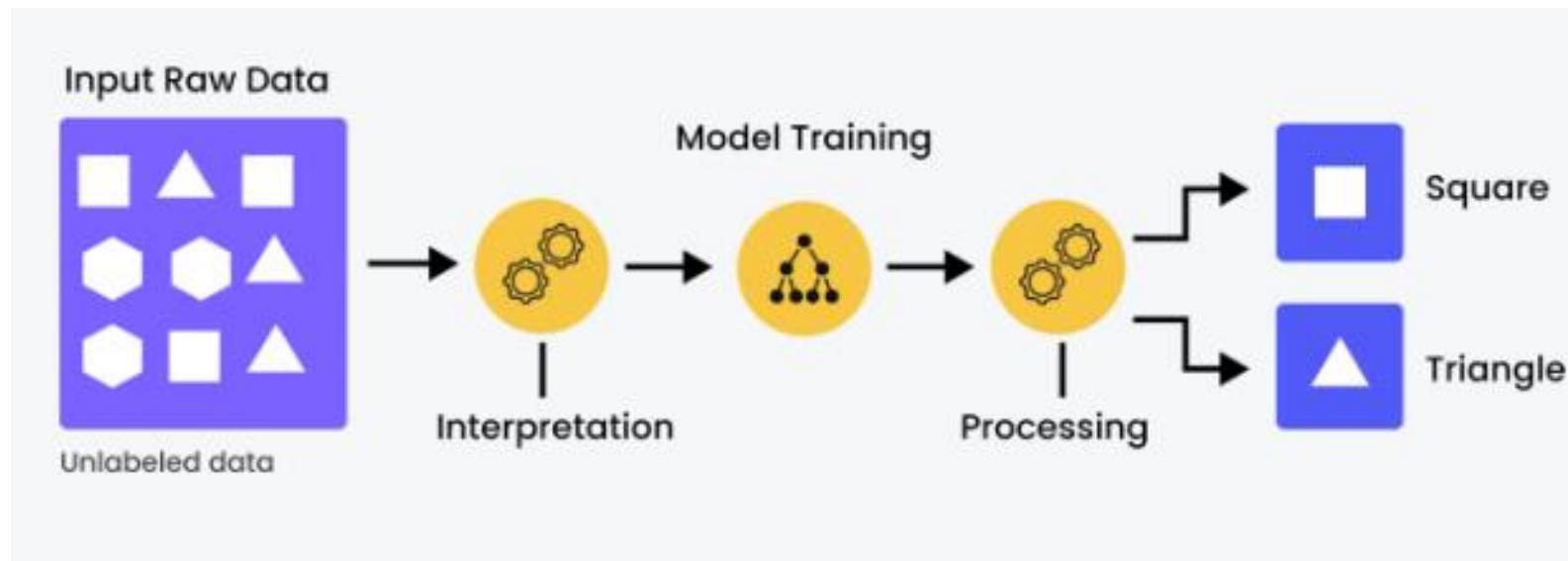
2.2.1 Semi-Supervised Filtering

Example (Bayesian) : HySAD introduces MC-Relief to select effective detection metrics, and uses EM algorithm to precisely separate attackers.



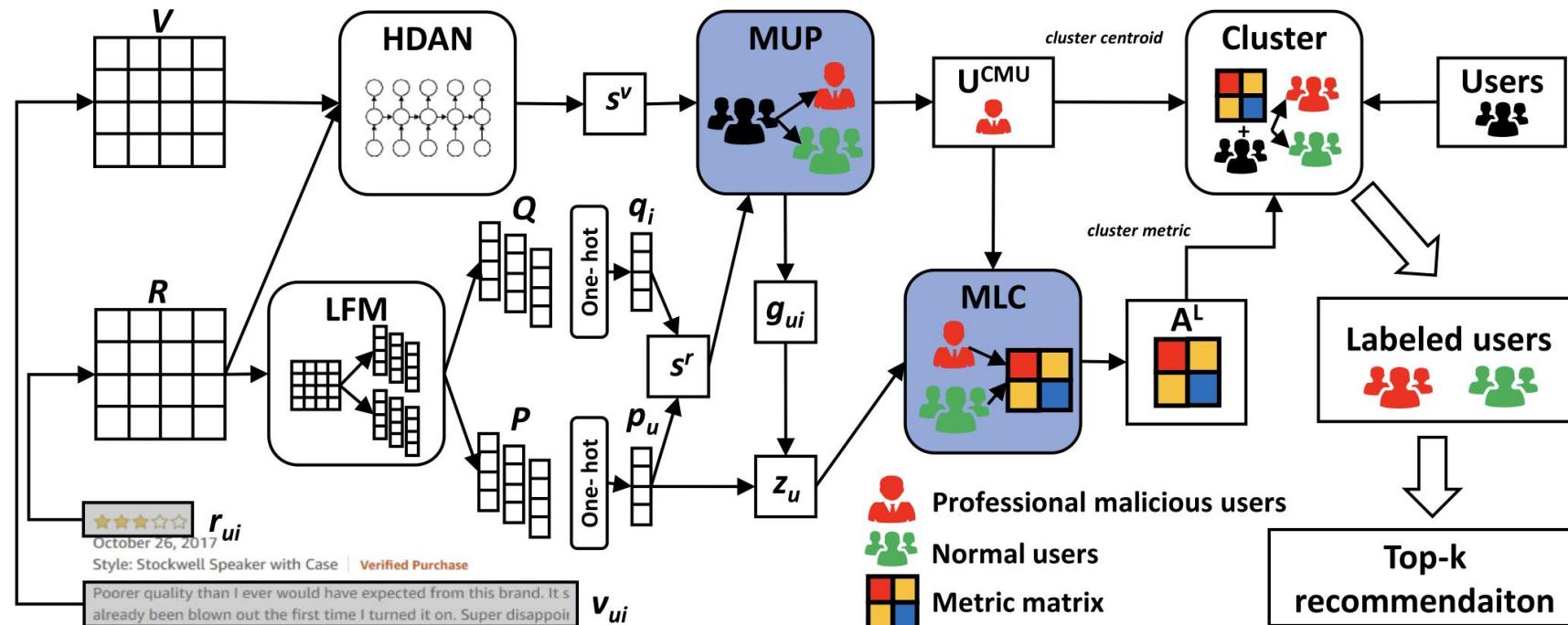
2.2.1 Unsupervised Filtering

Unsupervised Filtering (Label ALL Unknown): this type of methods are applied in scenarios lacking labeled data, which involves segregating data into various clusters without prior knowledge of their properties.

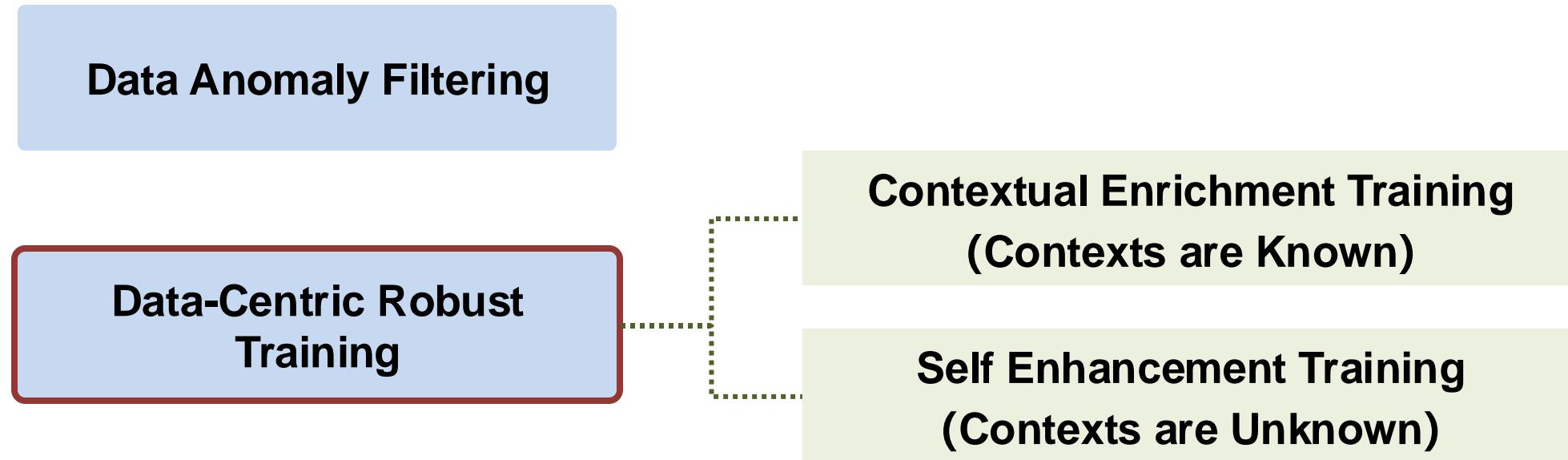


2.2.1 Unsupervised Filtering

Example (Clustering) : MMD utilizes metric learning to capture sentiment gaps of malicious users and true users.

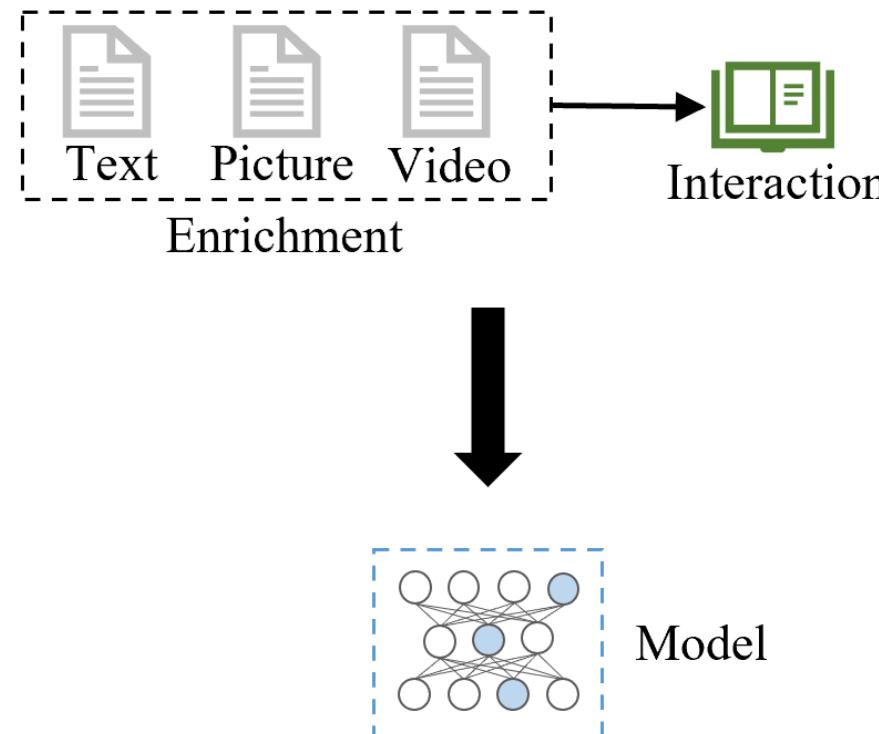


2.2 Protecting Data Integrity



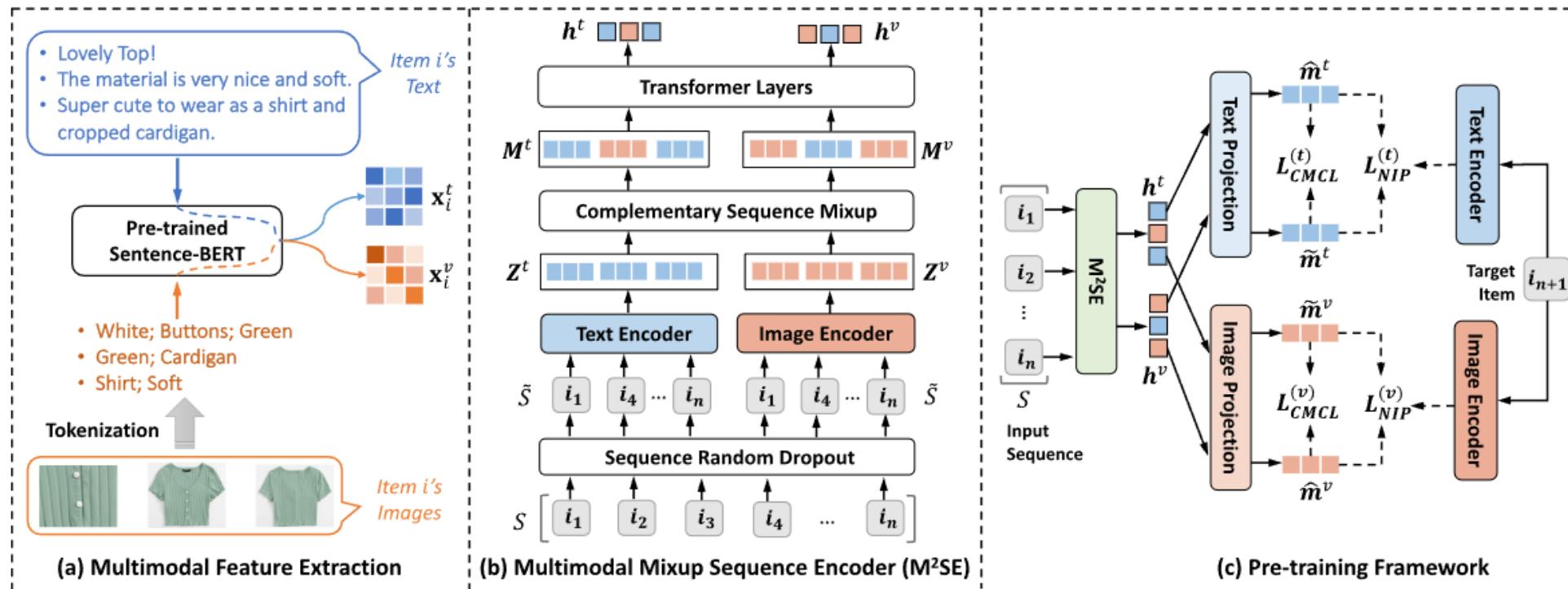
2.2.2 Contextual Enrichment Training

Contextual Enrichment Training (Contexts are Known): defenders expand the training dataset with a wide variety of additional signals such as user attributes, textual content, images, and social relations.



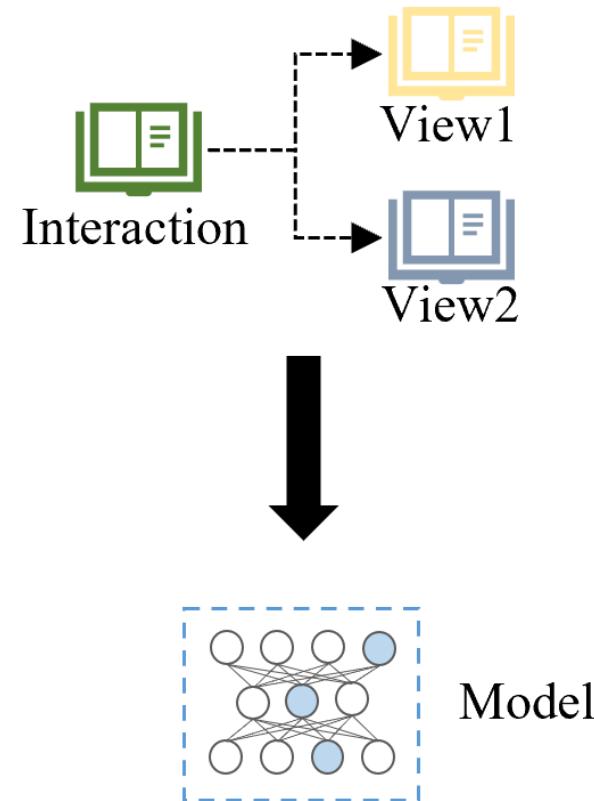
2.2.2 Contextual Enrichment Training

Example (Modality Information): MP4SR improves the modeling of user intentions by incorporating modality information with contrastive learning.



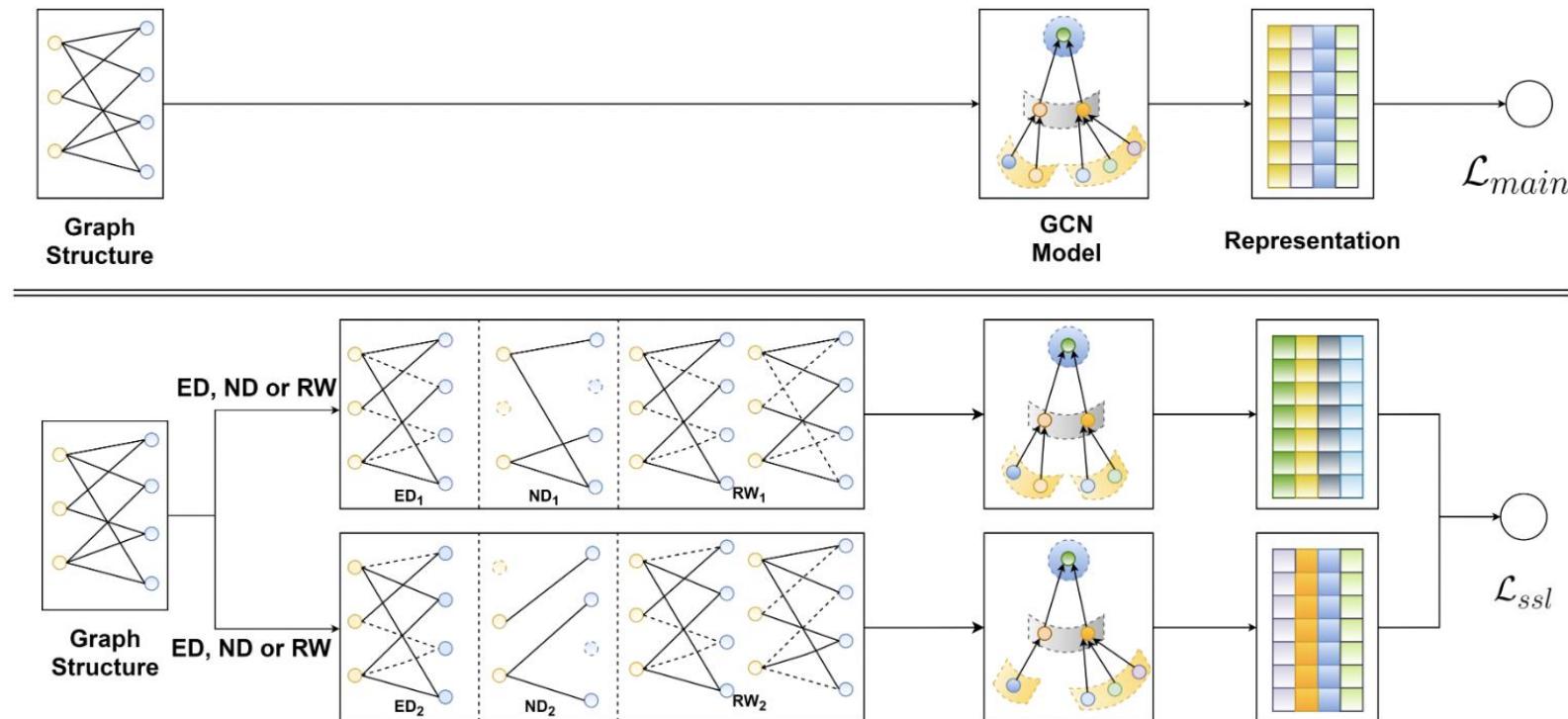
2.2.2 Self Enhancement Training

Self Enhancement Training (Contexts are Unknown): Defenders enhance model training by creating new data signals without additional information.



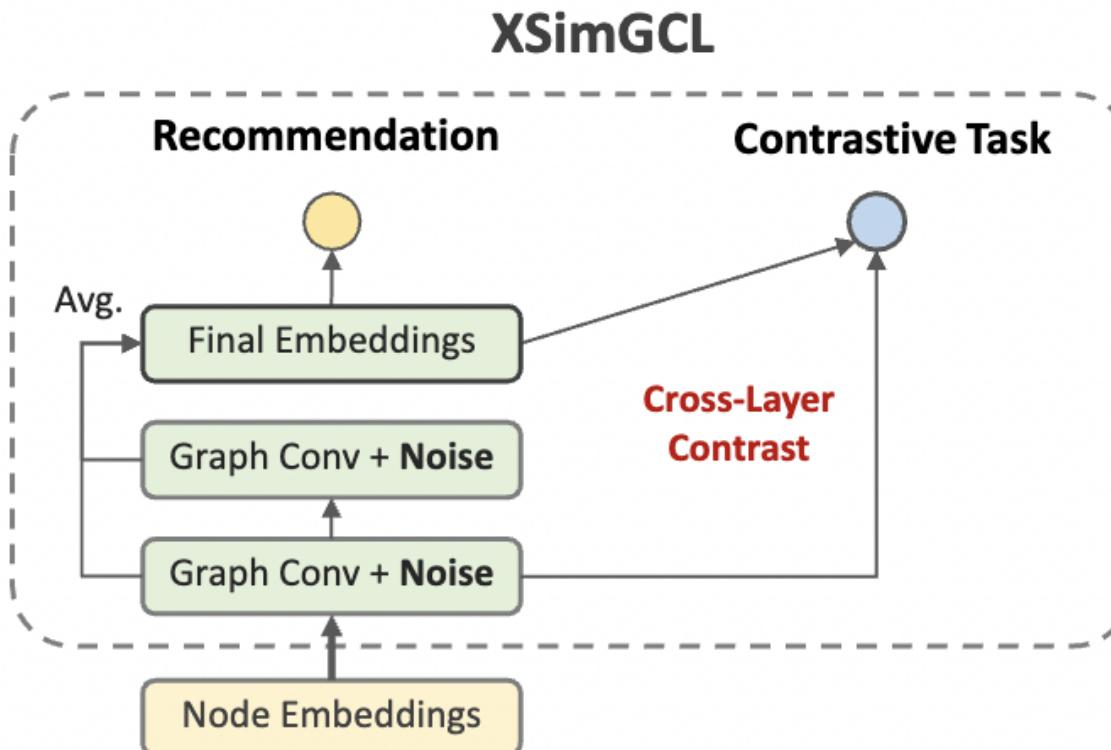
2.2.2 Self Enhancement Training

Example1 (Data-Level Self-Augmentation): SGL generates multiple views of a node, maximizing the agreement between different views of the same node compared to that of other nodes.



2.2.2 Self Enhancement Training

Example2 (Representation-Level Self-Augmentation): XSimGCL employs a simple yet effective noise-based embedding augmentation to generate views for contrastive learning.



1. Introduction

2. Securing Data Integrity

3. Preserving Data Privacy

4. Managing Data Noise

5. Limitations and Opportunities

6. Toolkit

Data Privacy Types

Data Privacy-Preserving Methods

3.1 Data Privacy Types

Main Privacy Types of Interest to Most Attackers

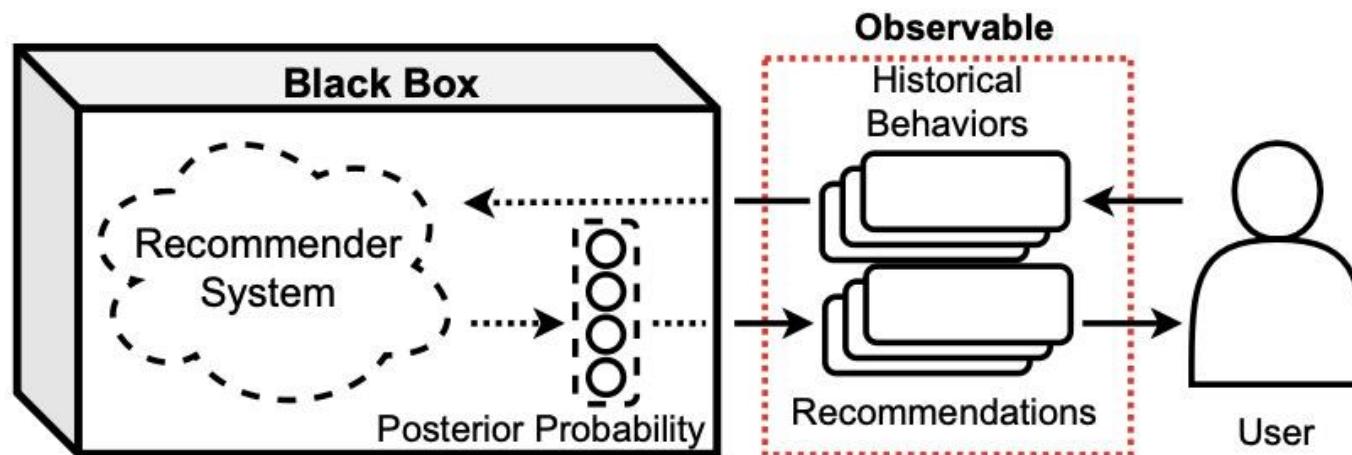
Data Membership Privacy

Data Attribute Privacy

Model Information Privacy

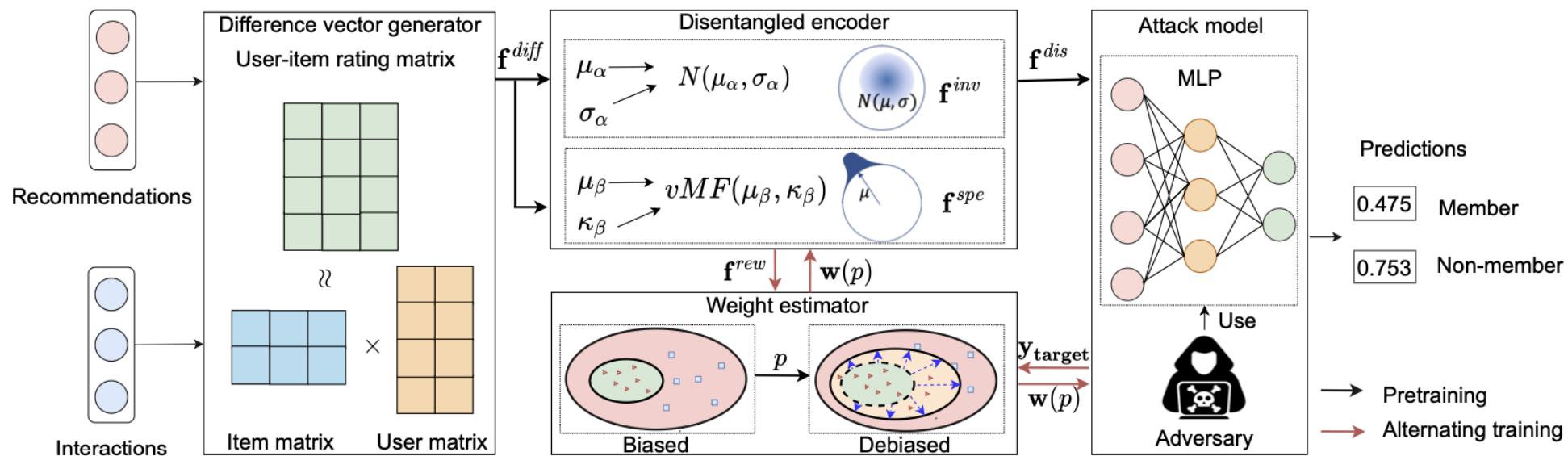
3.1.1 Data Membership Privacy

Data Membership Privacy: An attacker can infer the membership of a user, i.e., distinguish member users whose data was used for training the recommender system from non-member users of the model.



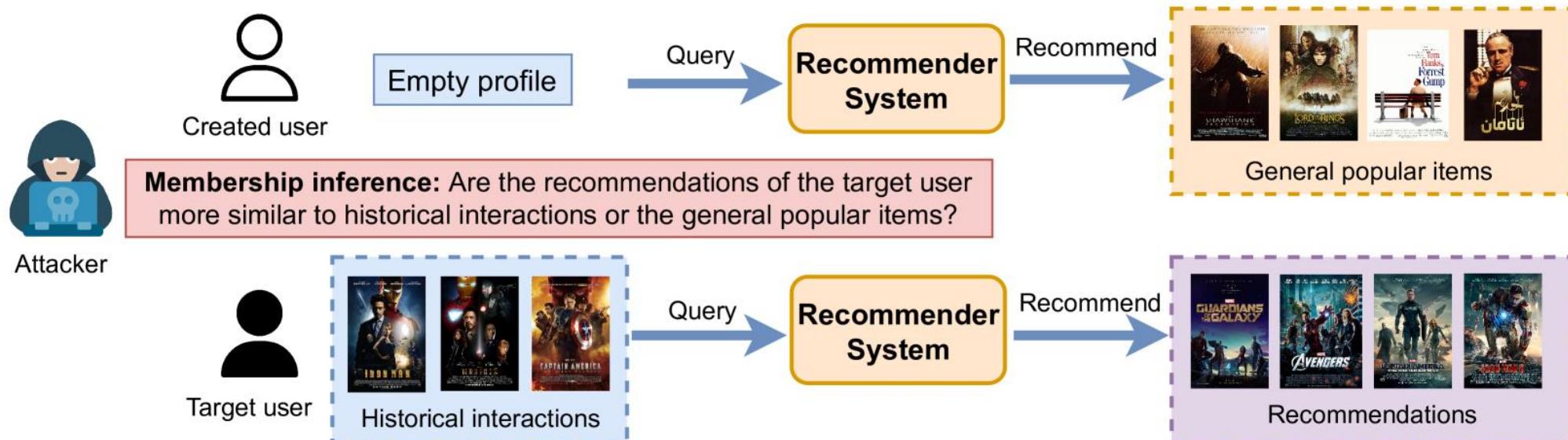
3.1.1 Data Membership Privacy

Example1 (Debias Learning): DL-MIA contains four main components: (i) a difference vector generator, (ii) a disentangled encoder, (iii) a weight estimator, and (iv) an attack model.



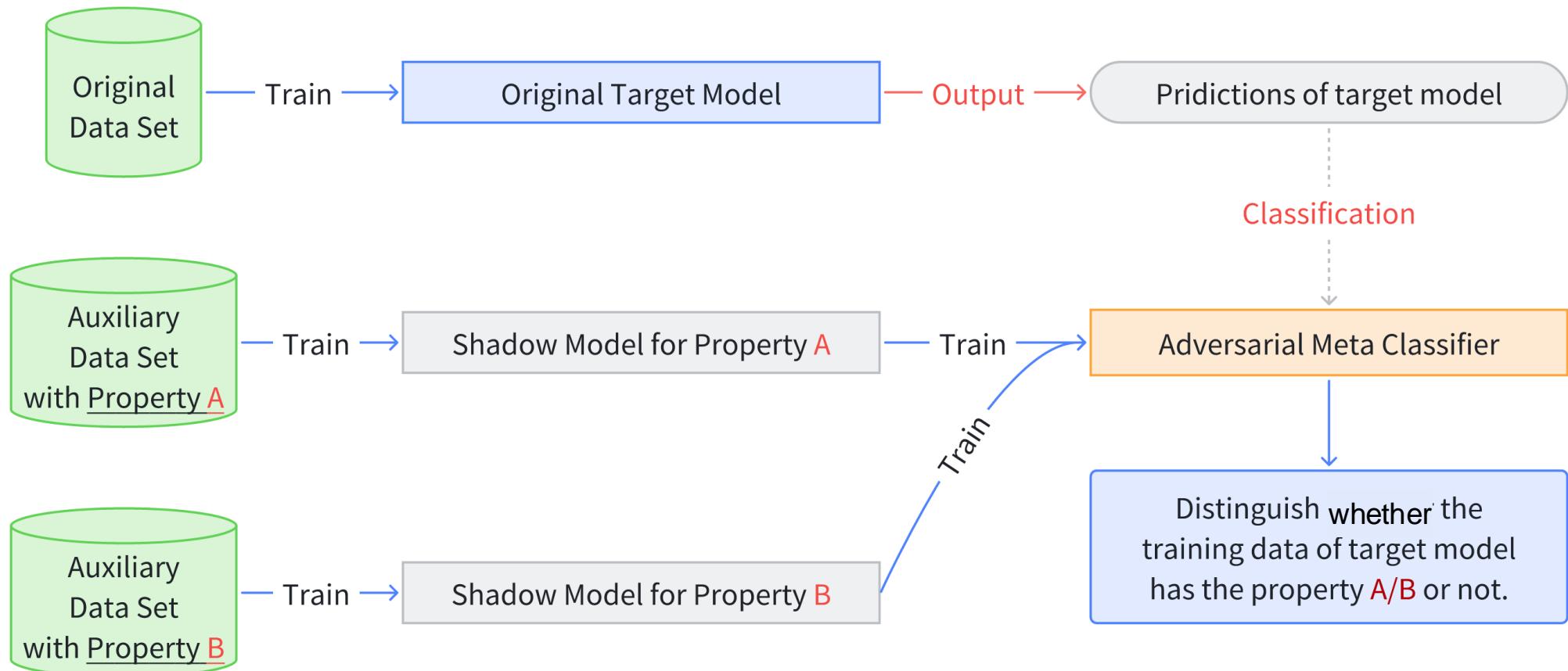
3.1.1 Data Membership Privacy

Example2 (Shadow-free): Shadow-free MIA examines whether the recommendations of the target user are more similar to his historical interactions or the general popular items to determine the membership status of the target user.



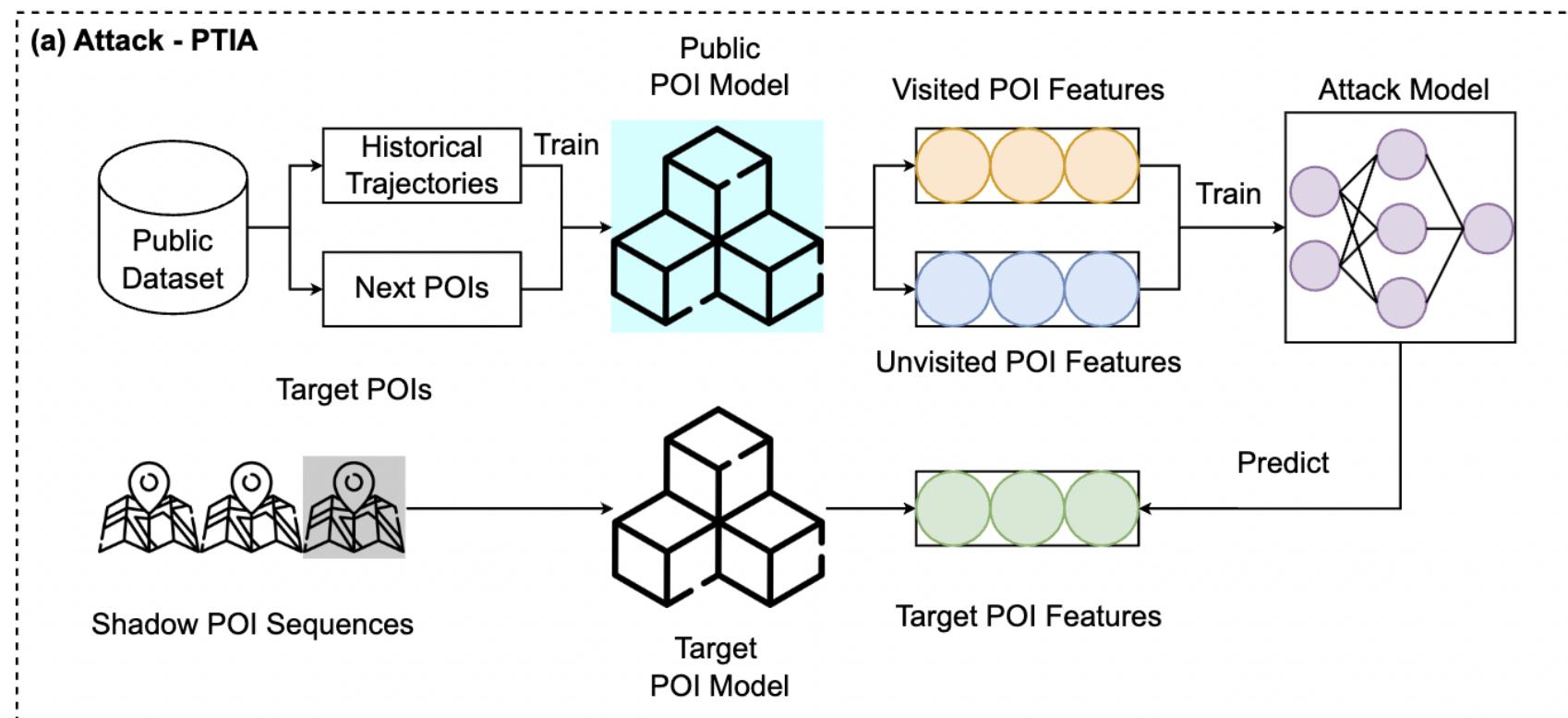
3.1.2 Data Attribute Privacy

Data Attribute Privacy: Attackers attempt to infer a specific attribute's presence or value, which usually contain personal, private, and highly sensitive information.



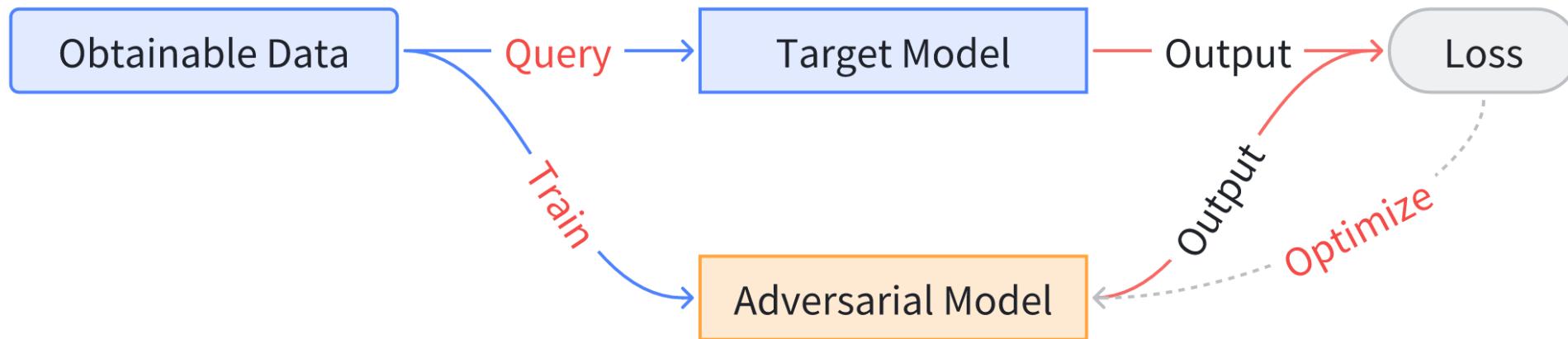
3.1.2 Data Attribute Privacy

Example (POI Information Privacy): PTIA demonstrates the user's POI check-ins in next POI recommendation services can be inferred, even in decentralized models.



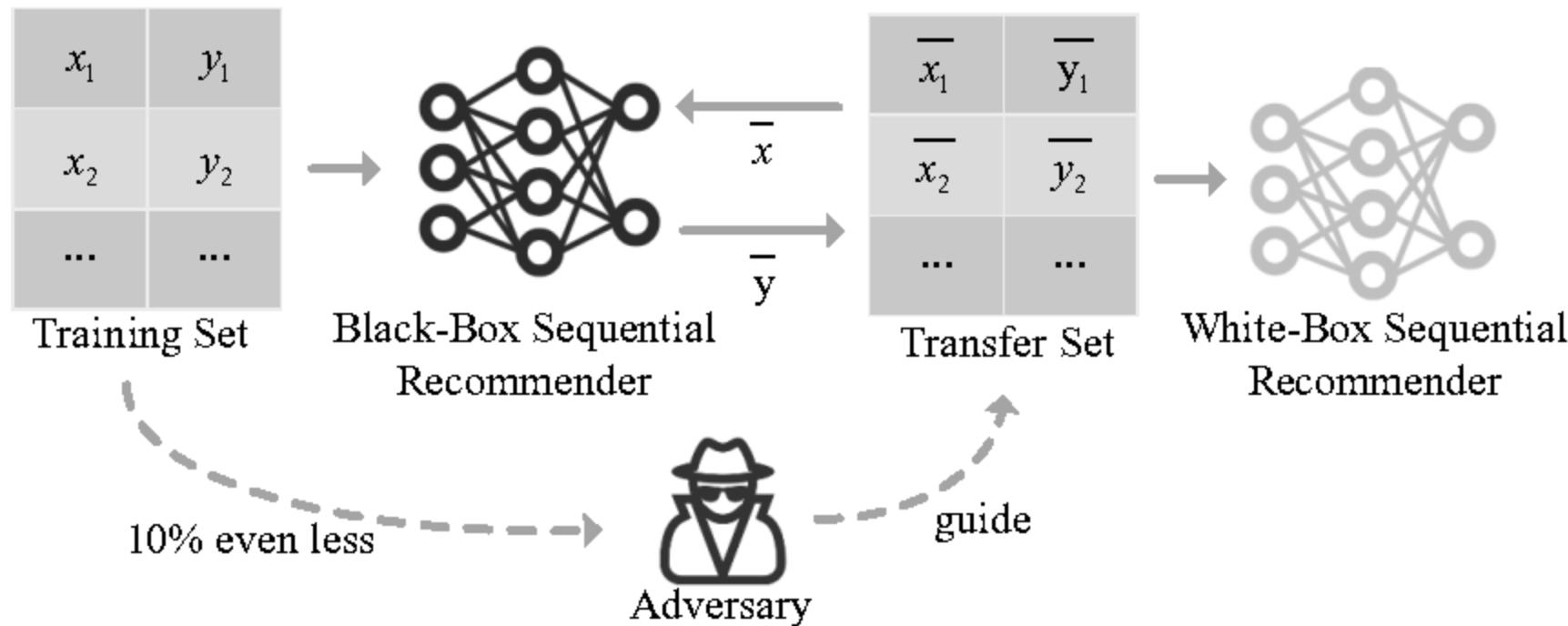
3.1.3 Model Information Privacy

Model Information Privacy: Attacker tries to extract the entire or partial model, its parameters, or the knowledge it has learned in a recommender system.



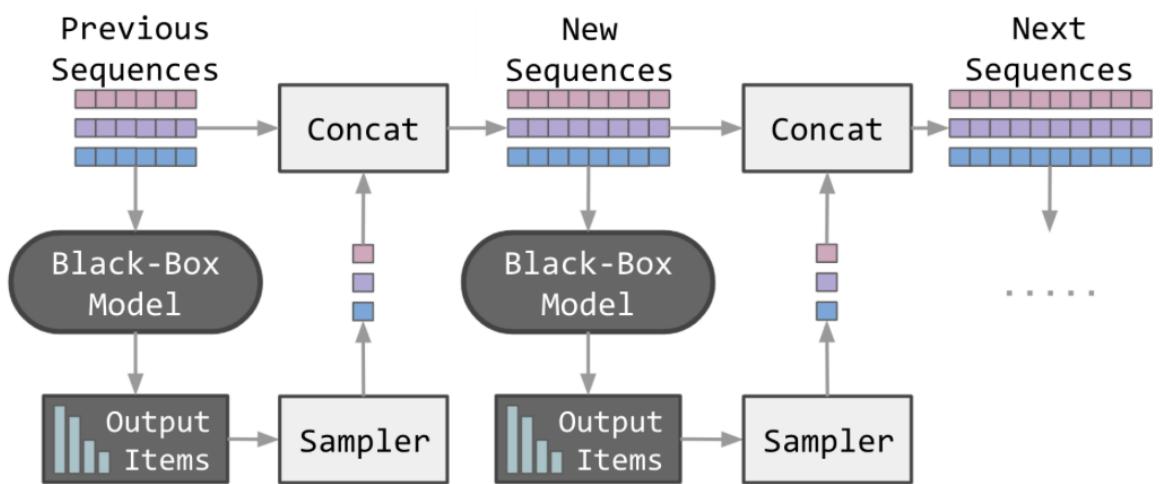
3.1.3 Model Information Privacy

Example1 (Knowledge Distillation): MEA uses few-shot raw data to obtain a white-box model.

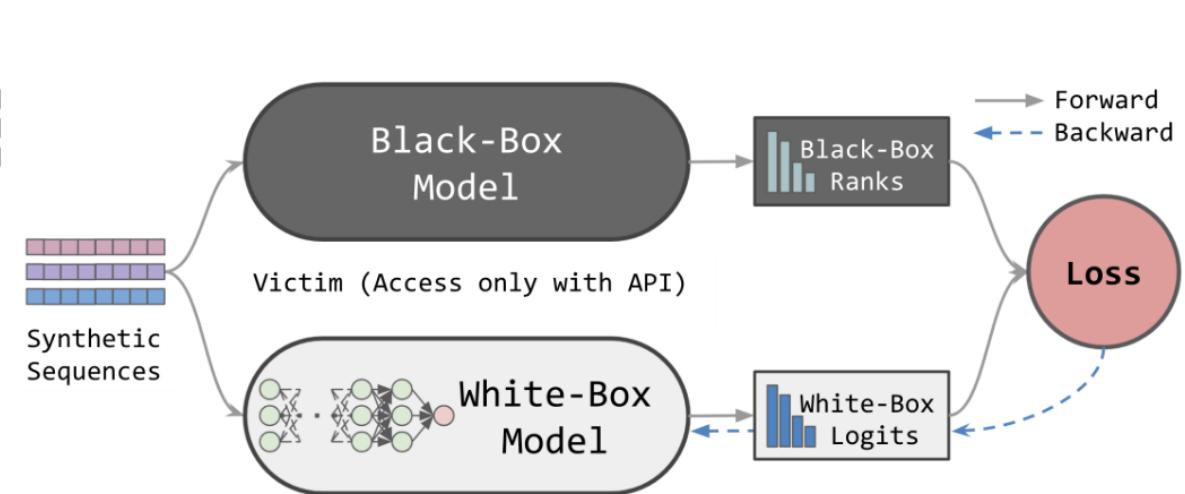


3.1.3 Model Information Privacy

Example2 (Knowledge Distillation): Data-free MEA merely needs limited-budget synthetic data generation and knowledge distillation.



(1) Autoregressive Data Generation



(2) Workflow of Model Extraction Attack on Sequential Recommender

3.2 Privacy-Preserving Methods

Main Defensive Strategies

Differential Privacy

Federated Learning

Adversarial Learning

3.2.1 Differential Privacy

Differential Privacy: A common way to preserve membership inference attacks, which can provide strict statistical guarantees for data privacy.

A randomized algorithm A is (ϵ, δ) -differentially private if for any pair of neighboring datasets D and D' differing in one record, and for all S that is a subset of the range of A , we have

$$\Pr(A(D) \in S) \leq e^\epsilon \Pr(A(D') \in S) + \delta,$$

where the probability is over the randomness of A . Usually, ϵ is assumed to be a small constant, and $\delta \ll 1/|D|$ where $|D|$ is the size of dataset D .

3.2.1 Differential Privacy

Example: EANA modifies DP-SGD. It adds noise only to embedding parameters with non-zero gradients during training.

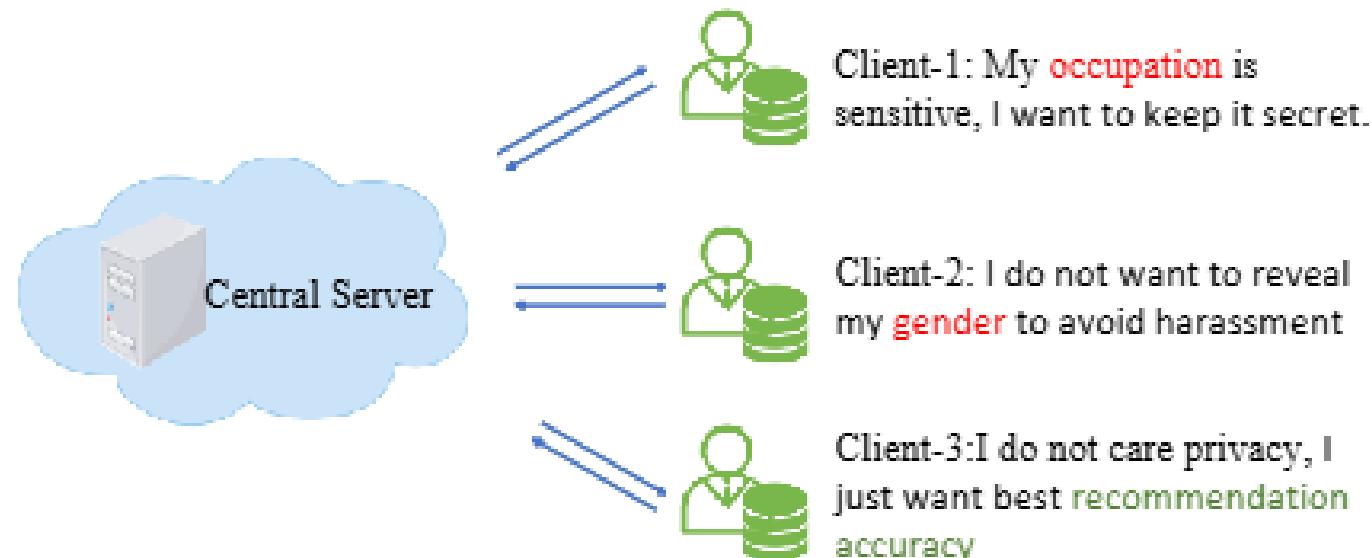
Algorithm 2 EANA

Require: Data set $D = \{d_1, \dots, d_n\}$, model with p non-embedding parameters and e embedding parameters, loss function: $\ell : \mathbb{R}^{p+e} \times D \rightarrow \mathbb{R}$, gradient ℓ_2 -norm bound: L , number of iterations: T , noise variance: σ^2 , learning rate: η .

- 1: Randomly initialize θ_0 .
 - 2: **for** $t = 0, \dots, T - 1$ **do**
 - 3: Randomly sample a batch of examples $B \subseteq D$.
 - 4: $g_t \leftarrow \frac{1}{|B|} \sum_{d \in B} \text{clip}(\nabla \ell(\theta_t; d), L) + \delta_E \odot \mathcal{N}(0, \sigma^2 \mathbb{I})$ where
 $\text{clip}(x, L) = \min\{1, L/\|x\|_2\} \cdot x$, $\delta_E \in \{0, 1\}^{p+e}$ is 1 for embedding parameters and 0 elsewhere, and \odot represents the element-wise product between two vectors.
 - 5: $\theta_{t+1} \leftarrow \theta_t - \eta g_t$.
 - 6: **end for**
 - 7: **return** θ_T .
-

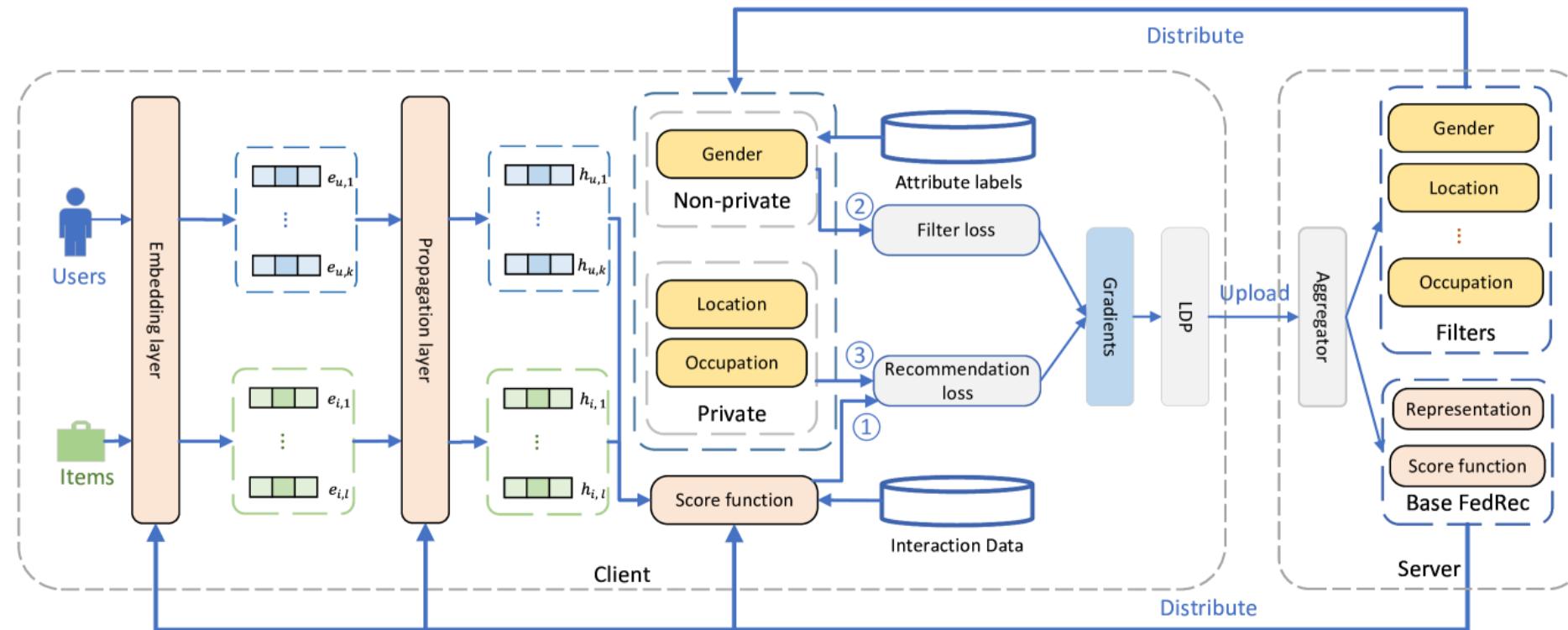
3.2.2 Federated Learning

Federated Learning: It isolates users' data and the cloud server by only transferring the gradients between them.



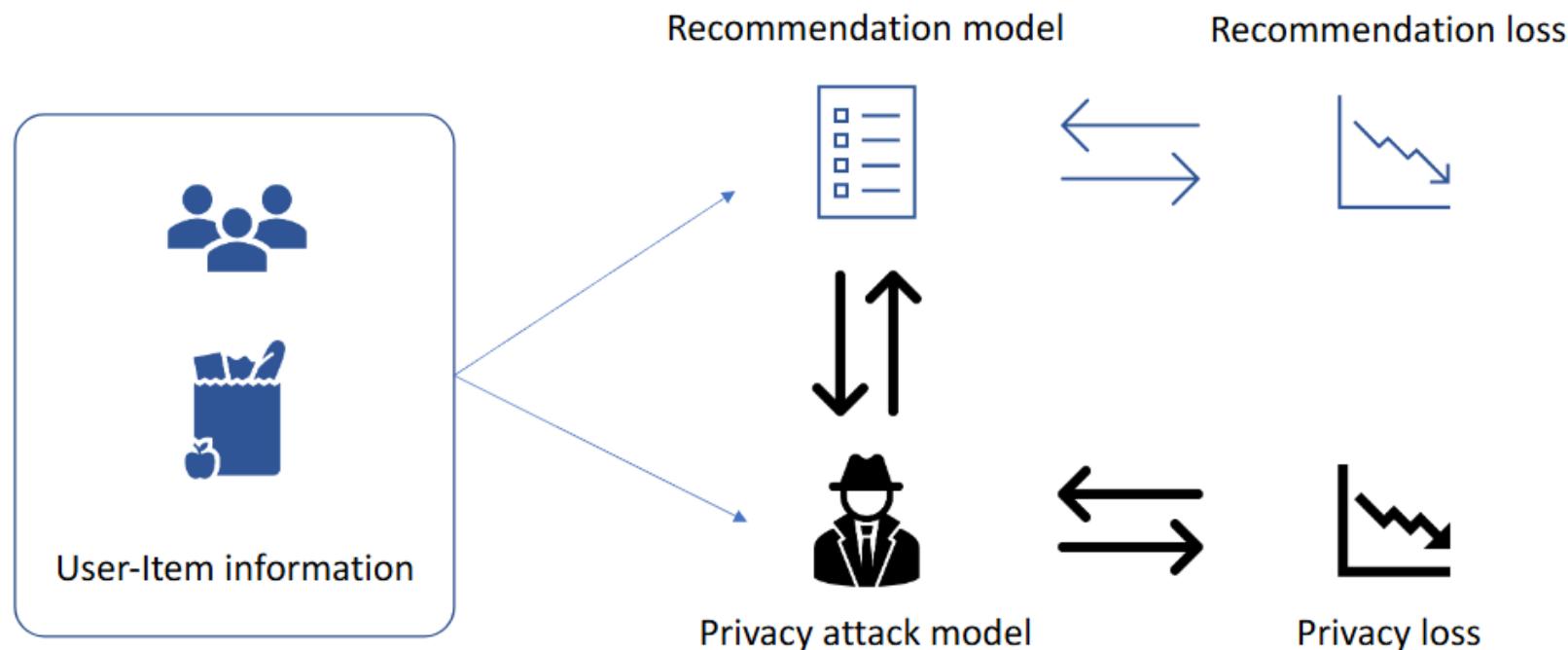
3.2.2 Federated Learning

Example: UC-FedRec ① train the recommendation part as the primary goal; ② train attribute distribution estimator on personal non-private attributes; ③utilize estimators to eliminate private information in the representation model.



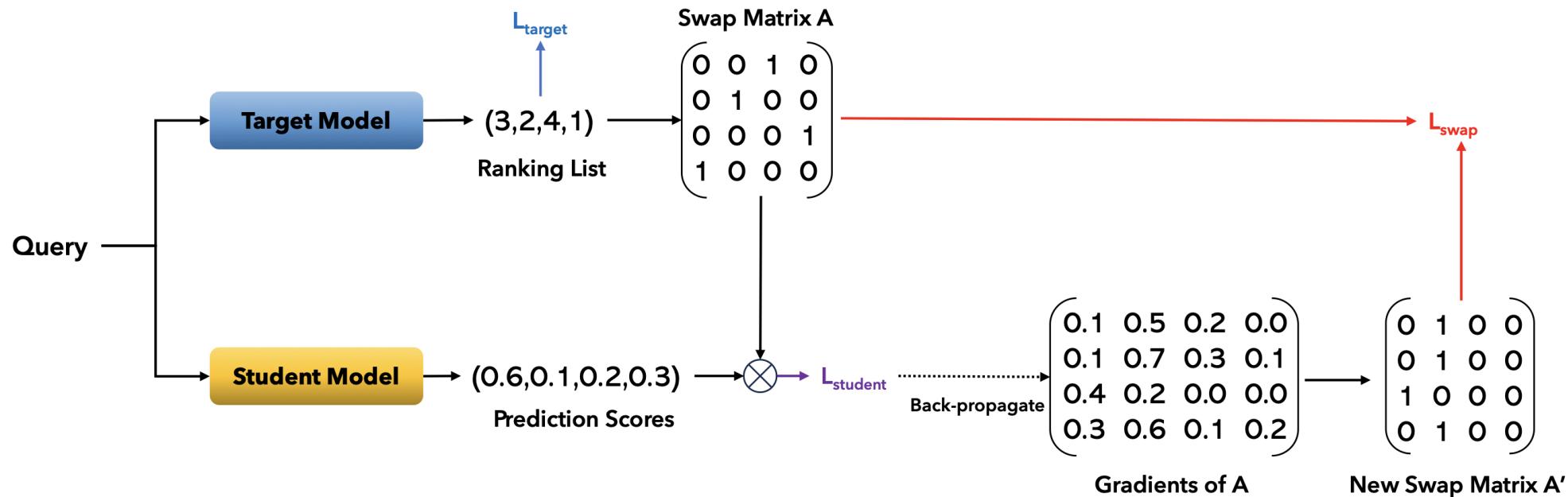
3.2.3 Adversarial Learning

Adversarial Learning: It minimizes the loss of the protected target model while maximizing the cost of the attacker.



3.2.3 Adversarial Learning

Example: GRO minimizes the loss of the protected target model while maximizing the loss of the attacker's surrogate model.



1. Introduction

2. Securing Data Integrity

3. Preserving Data Privacy

4. Managing Data Noise

5. Limitations and Opportunities

6. Toolkit

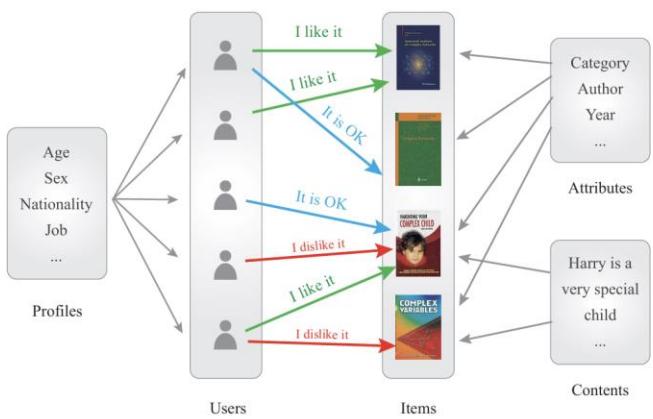
Origins and Types of Data Noises

Data Denoising Methods

4.1 Origins and Types of Data Noises



Noise from User Behaviors



Noise from RS Itself

**Data Noise
is Everywhere**



Noise from Items



External Noise

4.1 Noise in User Behavior Data

Mis-Operation Noise



Unintentional
clicks, or random
operations

Malicious Behavior Noise



Fake orders or
false reviews

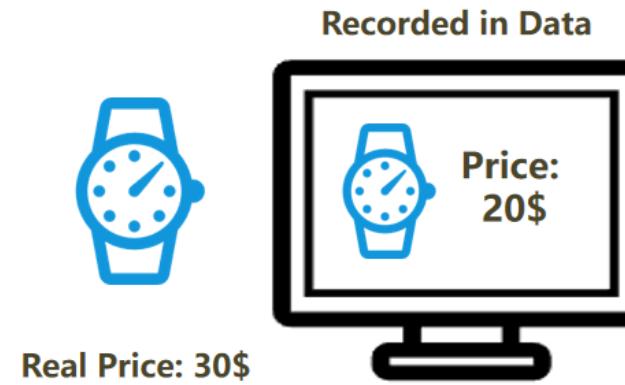
4.1 Noise in Items

Descriptive Noise



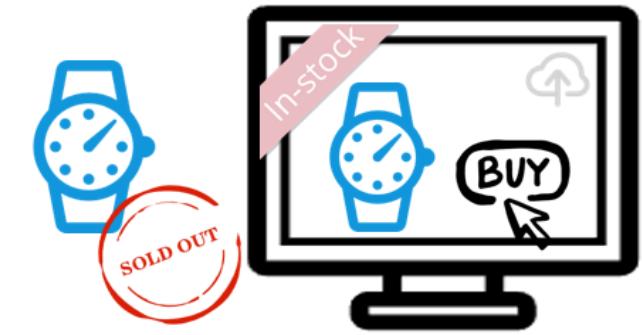
Typos, misspelling, or improper encoding can interfere with text-based item profiling methods

Attribute Noise



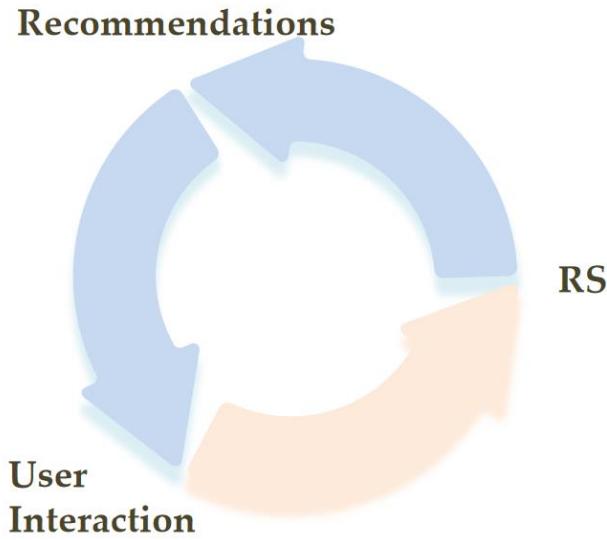
The actual attributes of the item, such as price and rating, are inconsistent with those recorded ones.

Dynamic Content Drift



Item information changes are not updated in a timely manner.

4.1 Noise From System



One characteristic of RS is the feedback loop — the exposure mechanism of the RS determines user behaviors, which are then fed back as training data for the RS. This feedback loop not only creates bias-involved noises but also intensifies them over time.

4.1 Noise From External Noise

Environmental Noise



Unexpected situations, such as the economy and politics, can lead to drastic changes in user behavior.

User Emotional Noise



Temporary emotional changes in users may cause their behavior to deviate from the normal state.

4.2 Data Denoising Methods

Data Cleaning

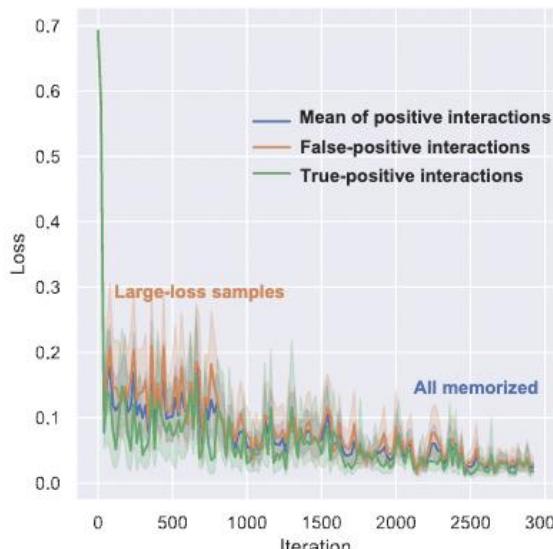
Data Augmentation

**Model / Training
Enhancement**

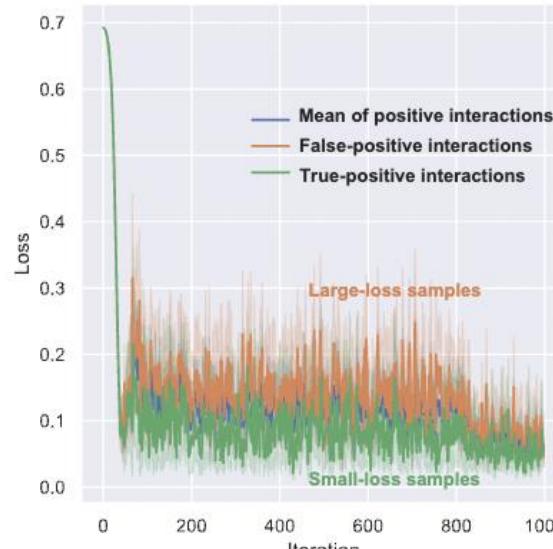
4.2.1 Data Cleaning

Data Cleaning refers to the process of identifying, correcting, or removing errors, inconsistencies, and irrelevant data from the dataset to improve the quality and reliability of the recommendations.

Example of noise identification: T-CE identifies samples with high loss as more likely to be noisy



(a) Whole training process

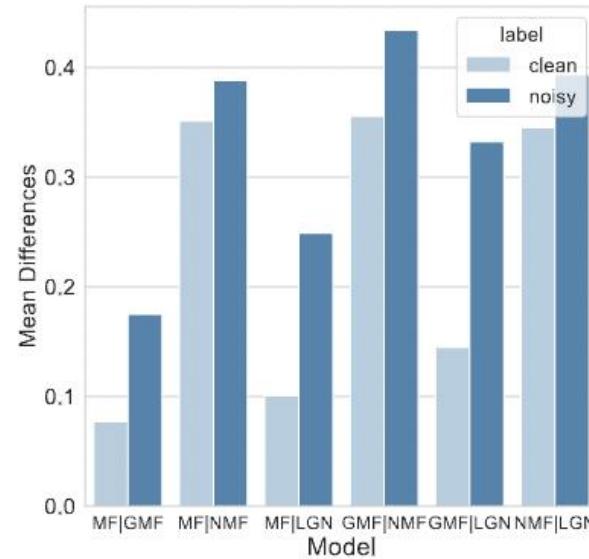


(b) Early training stages

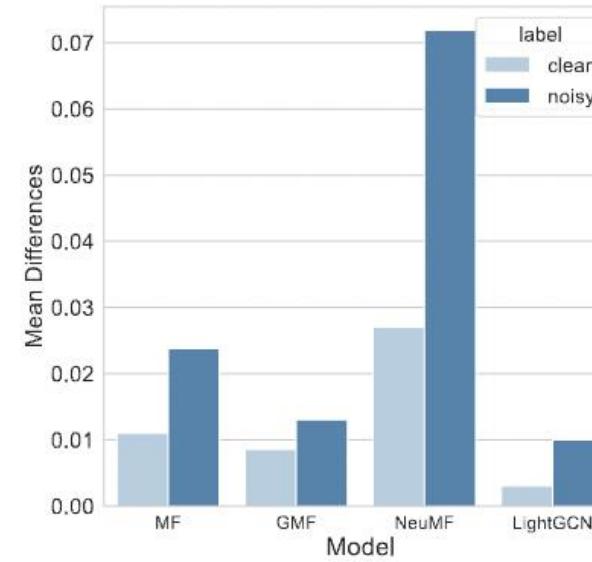
$$\mathcal{L}_{T-CE}(u, i) = \begin{cases} 0, & \mathcal{L}_{CE}(u, i) > \tau \wedge \bar{y}_{ui} = 1 \\ \mathcal{L}_{CE}(u, i), & \text{otherwise,} \end{cases}$$

4.2.1 Data Cleaning

Example2: DeCA finds that clean samples produce similar predictions across different models. By jointly training two recommendation models, it differentiates clean samples from noisy ones based on prediction disagreements.



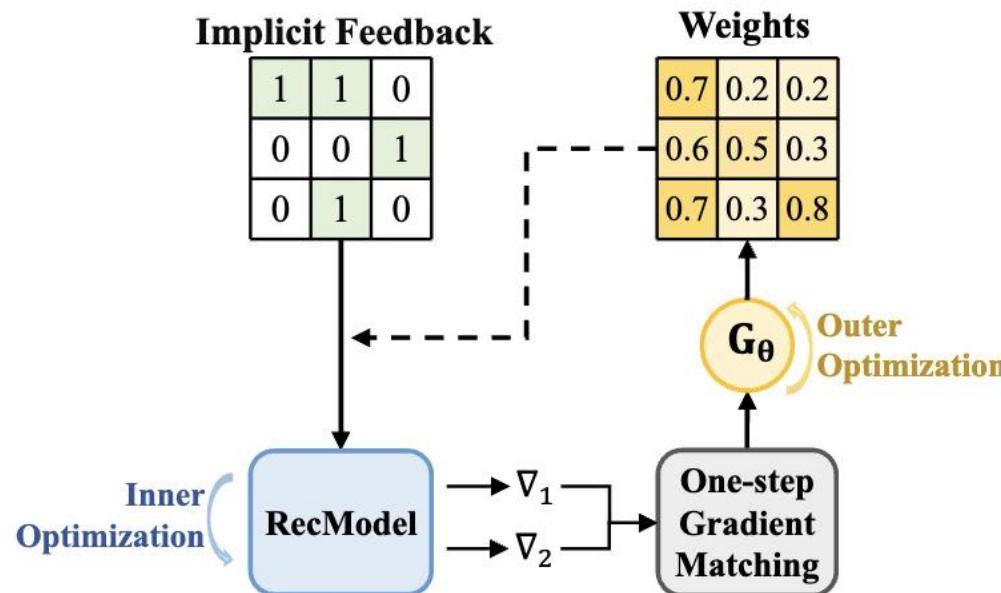
(a) Different models.



(b) Different random seeds.

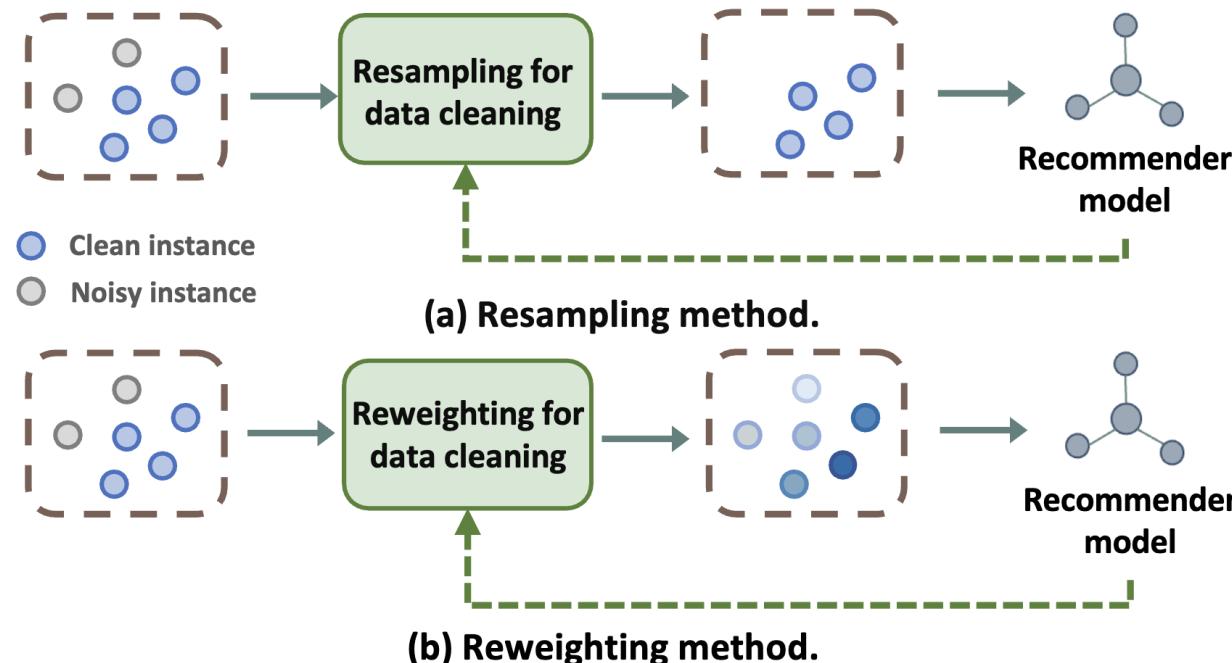
4.2.1 Data Cleaning

Example3: BOD finds that samples yielding consistent losses across different loss functions are likely to be clean.



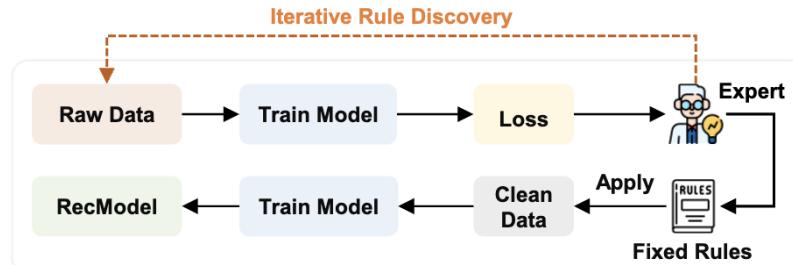
4.2.1 Data Cleaning

Resampling methods aim to identify noisy interactions and sample more clean interactions for training. **Reweighting methods** utilize all training interactions yet assign lower weights to potential noisy ones.

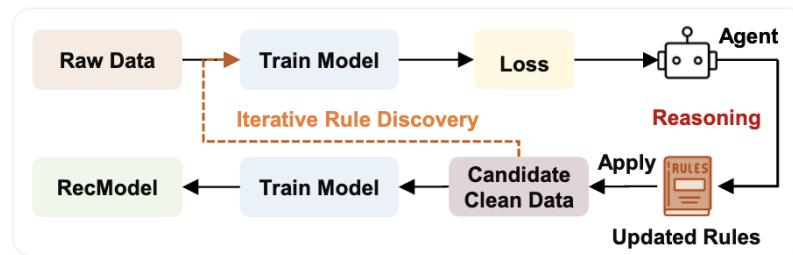


4.2.1 Data Cleaning

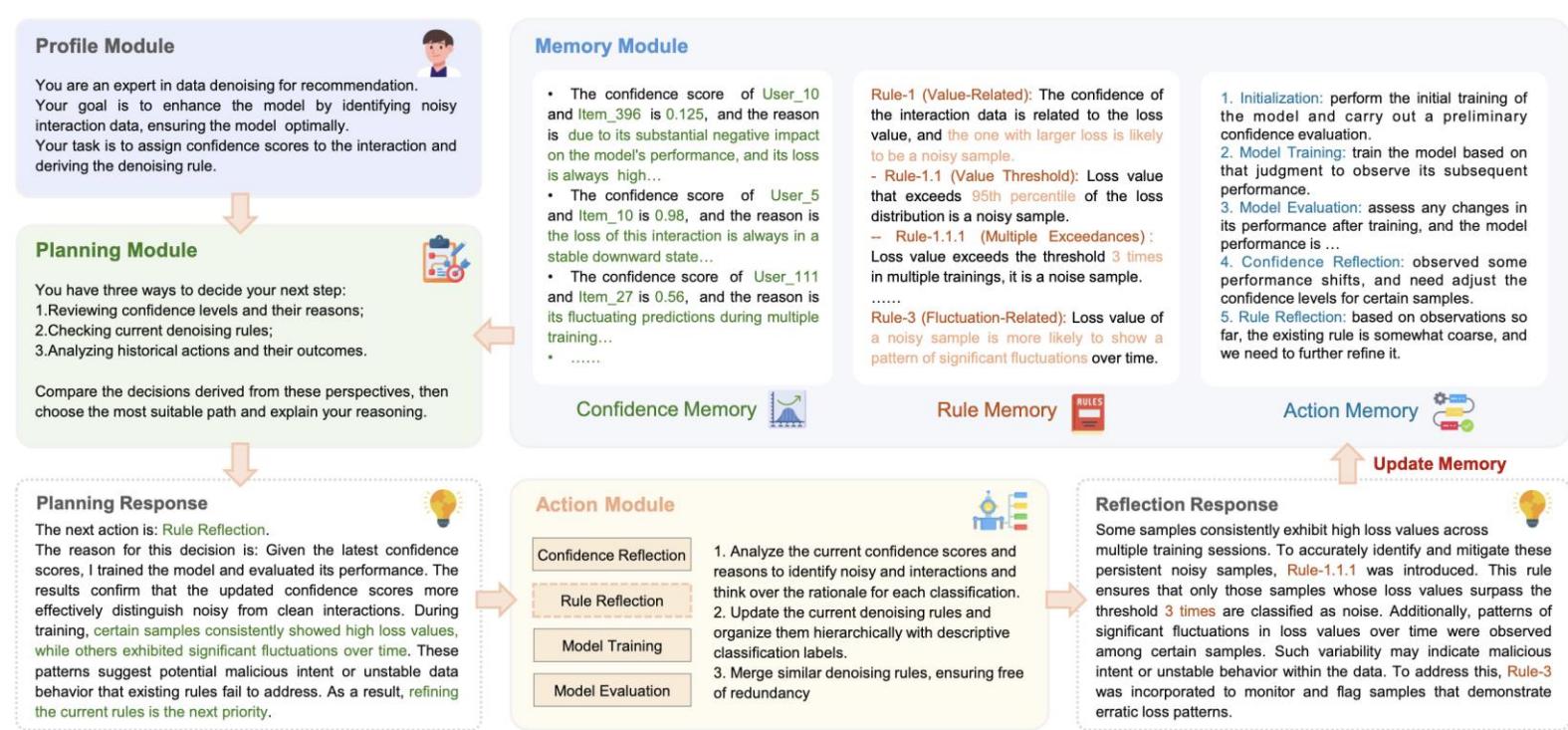
Data cleaning with LLM: RuleAgent leverages the potential of LLM-based agent to offer rapid and dynamic rule discovery for data cleaning.



(a) Expert-Guided Rule Discovery

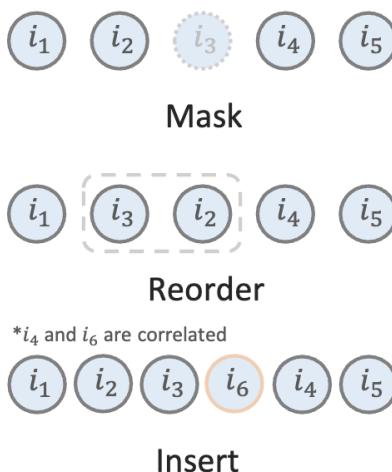
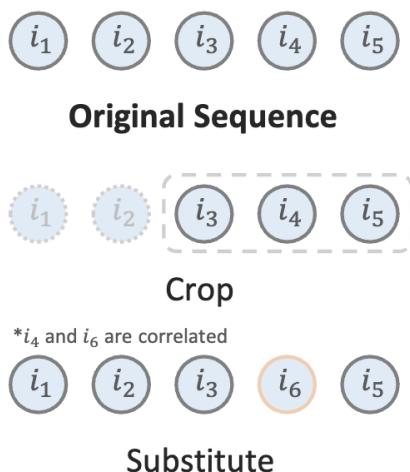


(b) Agent-Driven Rule Discovery

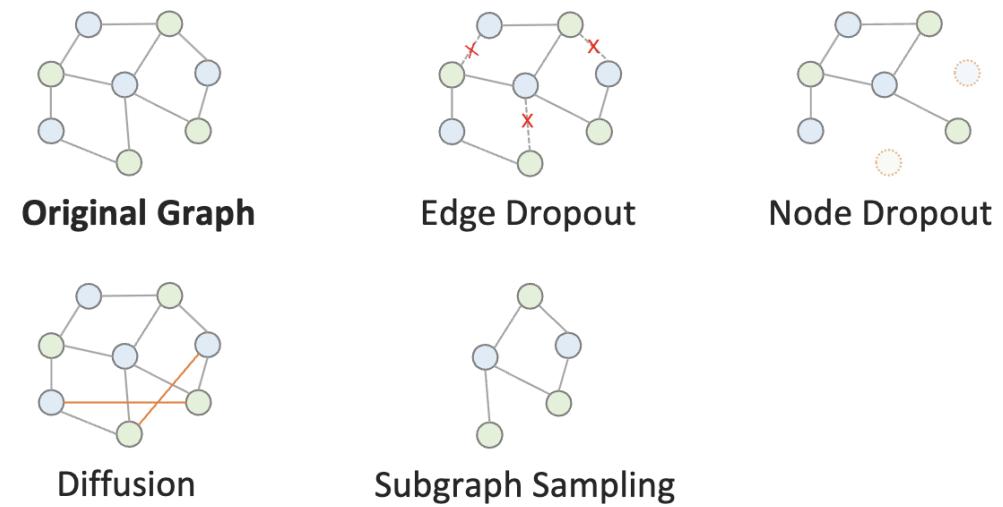


4.2.2 Data Augmentation

Data Augmentation expands the training data by generating new data instances or modifying existing ones to improve the model's performance, generalization, and robustness against noises.



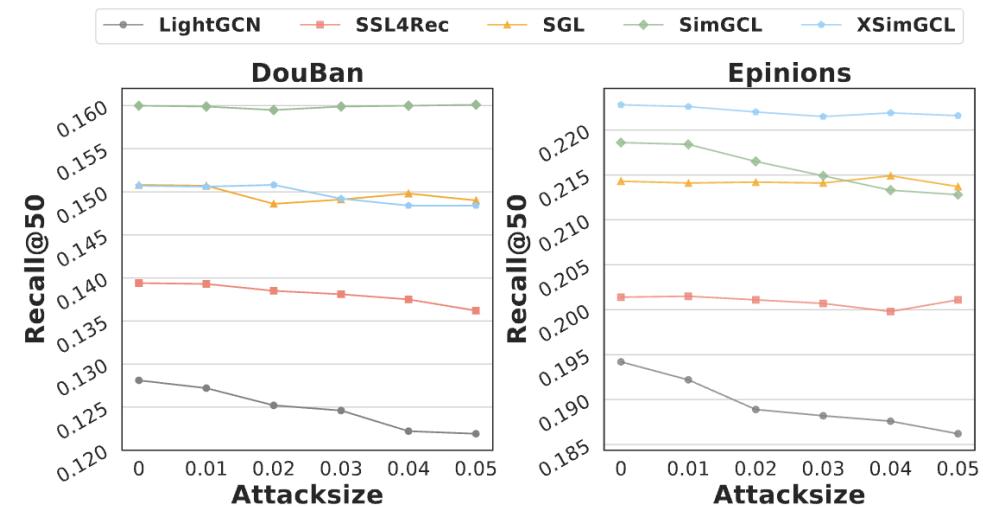
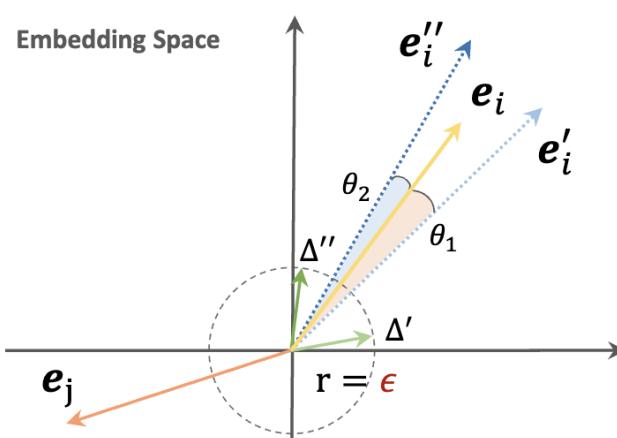
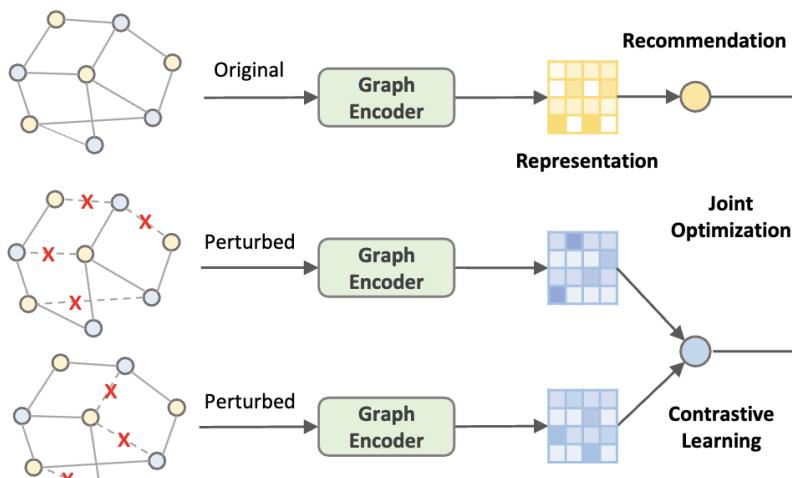
(a) Sequence-based data augmentation



(b) Graph-based data augmentation

4.2.2 Data Augmentation

Example1: SGL and SimGCL/XSimGCL mitigates the negative effects of the noise behaviors by augmenting graph data/feature and leveraging contrastive learning.



Wu, Jiancan, et al. "Self-supervised graph learning for recommendation." *S/G/I/R*. 2021.
 Yu, Junliang, et al. "XSimGCL: Towards extremely simple graph contrastive learning for recommendation." *IEEE Transactions on Knowledge and Data Engineering* 36.2 (2023): 913-926.

4.2.2 Data Augmentation

Example2: CoSeRec introduces two informative augmentation operators leveraging item correlations to create high quality views for contrastive learning.

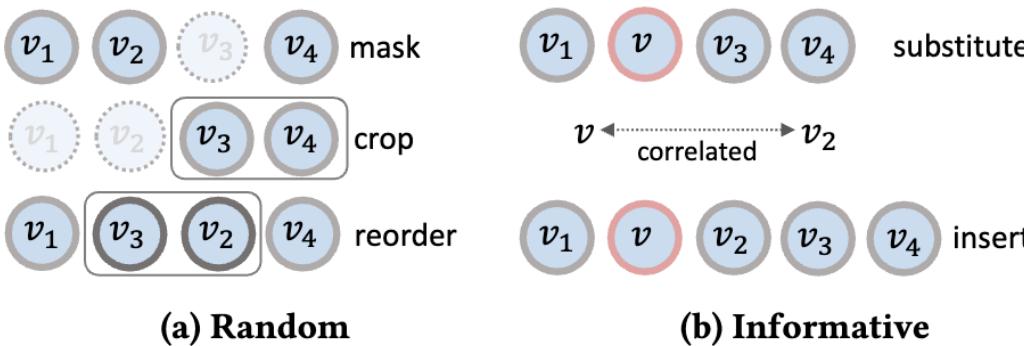
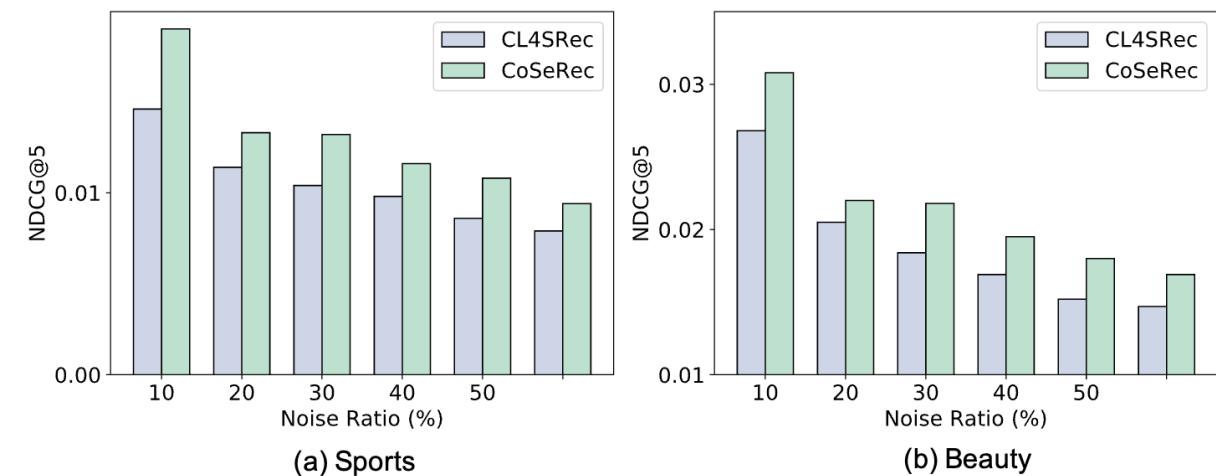
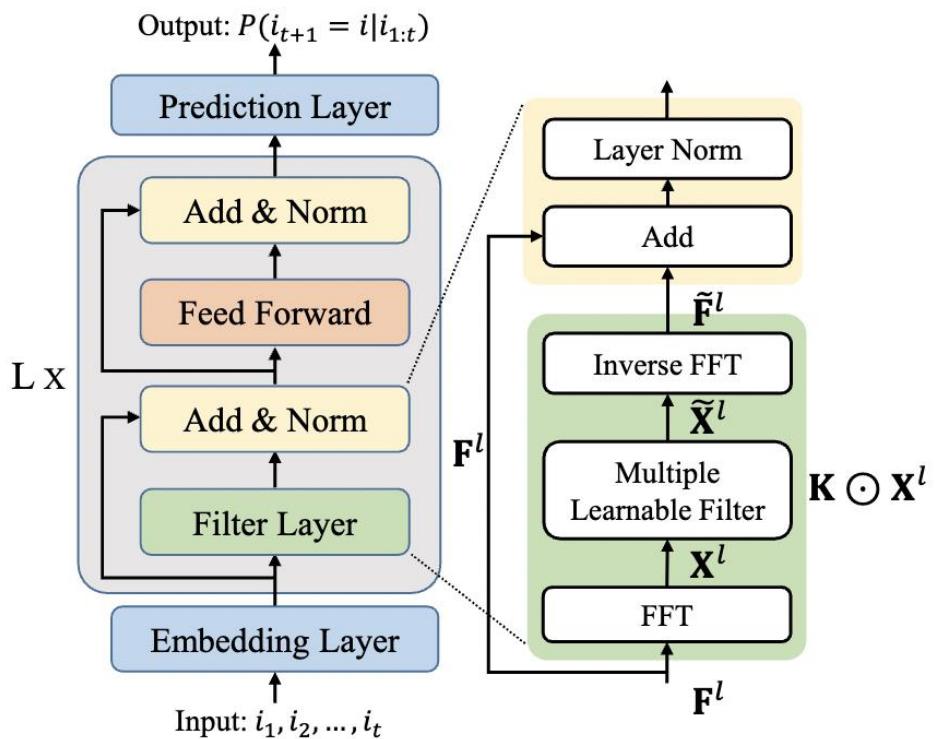


Figure 2: Five augmentations adopted in this paper. The original sequence is $[v_1, v_2, v_3, v_4]$. Informative augmentations incorporate correlations among items.



4.2.3 Model Enhancement

Model / Training Enhancement: Improve the structure or training method of the recommendation model itself to enhance its robustness to noise. This can be achieved by introducing a more complex model architecture or adopting specialized training techniques.



Example1: FMLP-Rec denoises sequential recommendation data by applying Fast Fourier Transform (FFT) to convert item embeddings into the frequency domain. It then uses learnable filters to suppress noise and highlight meaningful signals before transforming the data back via Inverse FFT (IFFT).

4.2.3 Training Enhancement

Example2: Wang's work proposes Adaptive Denoising Training (ADT) strategies for recommenders, which dynamically prunes the large loss interactions along the training process.

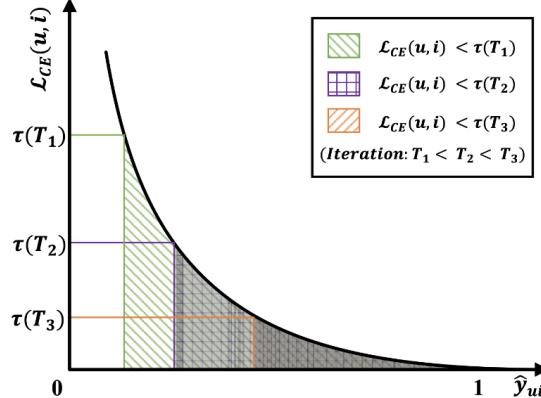
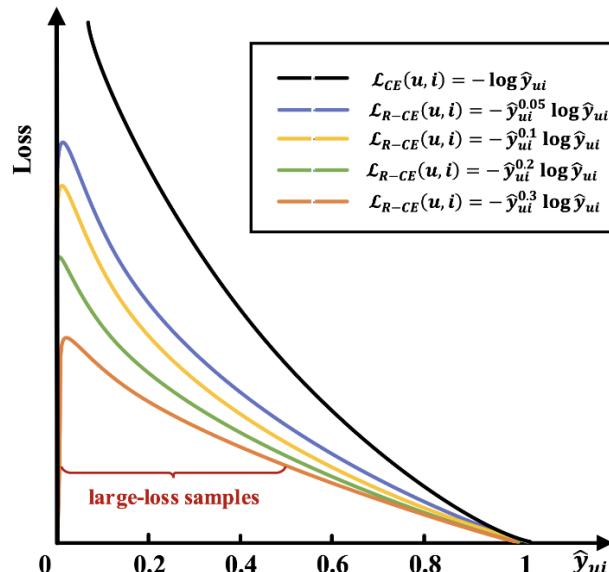


Figure 4: Illustration of T-CE loss for the observed interactions (i.e., $\hat{y}_{ui} = 1$). T_i is the iteration number and $\tau(T_i)$ refers to the threshold. Dash area indicates the effective loss and the loss values larger than $\tau(T_i)$ are truncated.



(a) R-CE loss for the observed positive interactions. The contributions of large-loss interactions are greatly reduced.

Two paradigms for adaptive loss formulation:

- **Truncated Loss (T-CE)** that discards the large-loss samples with a dynamic threshold in each iteration;
- **Reweighted Loss (R-CE)** that adaptively lowers the weights of large-loss samples.

1. Introduction

2. Securing Data Integrity

3. Preserving Data Privacy

4. Managing Data Noise

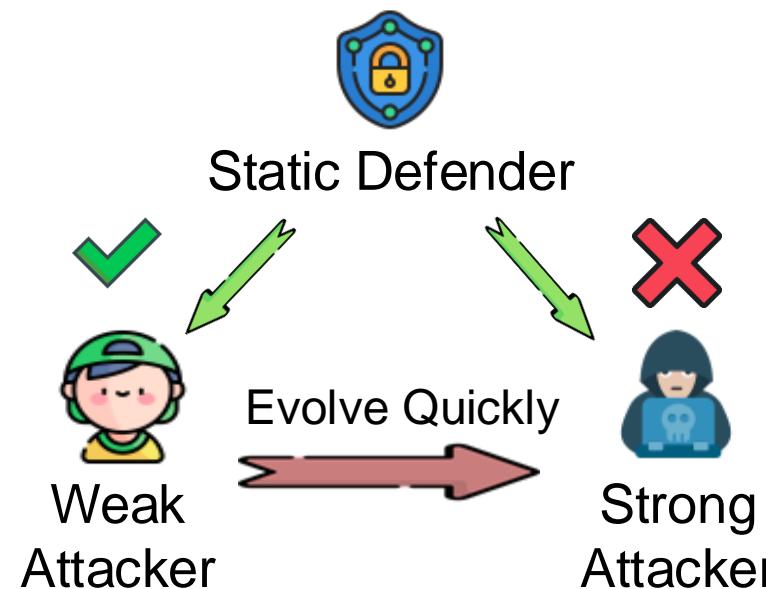
5. Limitations and Opportunities

6. Toolkit

5. Limitations and Opportunity

Lagging of Data Security Defense

Current defense mechanisms (e.g., anomaly detection) exhibit limited adaptability to evolving attack vectors (e.g., dynamic gradient manipulation, cross-platform collusion), as they primarily rely on pre-defined attack signatures rather than real-time data behavior monitoring.



5. Limitations and Opportunity

Data Quality Degradation via Over-Aggressive Sanitization

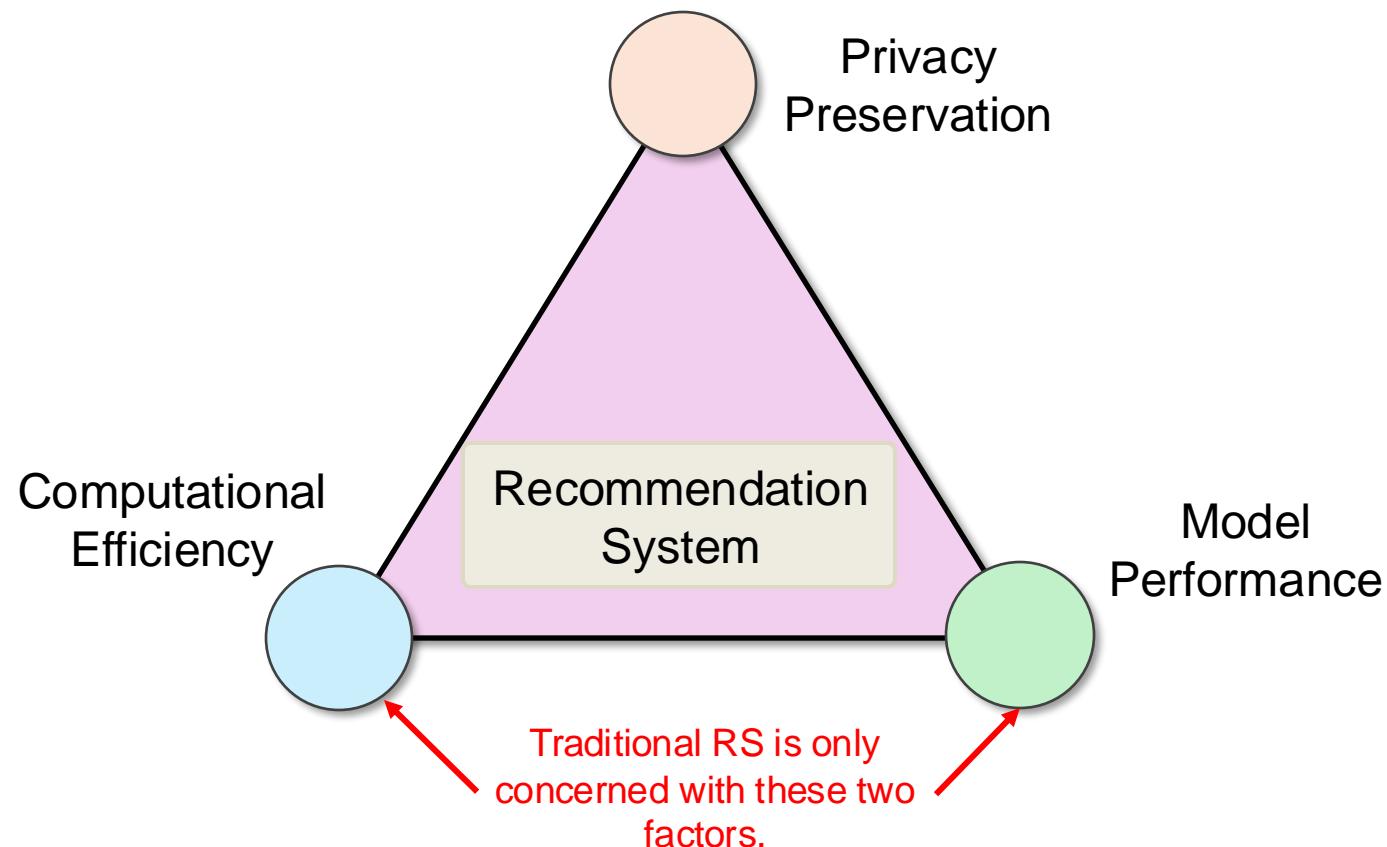
Data cleaning and verification technologies may mistakenly delete high-value data (such as sparse but important user behaviors), resulting in reduced recommendation accuracy.

		Predicted Value	
		Positive	Negative
Actual Value	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

5. Limitations and Opportunity

The Trade-Off between Data Protection and Recommendation Performance

Protection methods degrade the semantic richness of user interaction data, leading to homogenized recommendations and reduced personalization.



5. Limitations and Opportunity

Difficulty in Distinguishing Natural Noise and Malicious Noise

Natural noise (such as user mis-clicks, device failures) and maliciously injected noise (such as adversarial samples, false comments) may be highly similar in behavior patterns, and existing methods are difficult to accurately distinguish.



5. Limitations and Opportunity

Integrity, Privacy, and Noise Issues Are Treated in Isolation

Most studies deal with security attacks (such as poisoning), privacy leaks (such as attribute reasoning), and noise issues in isolation, without considering the correlation between the three.



5. Limitations and Opportunity

Dynamic Data Monitoring and Adaptation



Opportunity 1: Light-Weight and Real-Time Data Trustworthiness Scoring

Design and implement real-time streaming frameworks for evaluating data authenticity by leveraging multimodal indicators, including temporal interaction patterns and cross-platform consistency metrics.



Opportunity 2 : Malicious Data Tracing

Investigate causal inference/watermarking methodologies to differentiate the underlying factors contributing to data anomalies.

5. Limitations and Opportunity

Multiple Data Goal Collaborative Defense



Opportunity: All-in-one Unified Data Representation Learning

Develop all-in-one frameworks to construct feature representations that simultaneously preserve privacy-sensitive attributes and maintain robustness against adversarial attacks through encoded interaction patterns.

5. Limitations and Opportunity

LLM-Driven Data Curation and Protection

Opportunity 1: Semantic-Aware Data Sanitization



Utilize large language models as contextual filtering mechanisms for detecting sophisticated adversarial content through semantic pattern recognition (e.g., identifying semantically inconsistent reviews that conceal promotional intent).

Opportunity 2: Synthetic Data Augmentation



Generating more synthetic data can help alleviate data sparsity while also incorporating additional credible data.

Opportunity 3: Autonomous Data Governance Agents



Develop LLM-powered monitoring agents that dynamically adjust data ingestion policies based on real-time trust metrics.

1. Introduction

2. Securing Data Integrity

3. Preserving Data Privacy

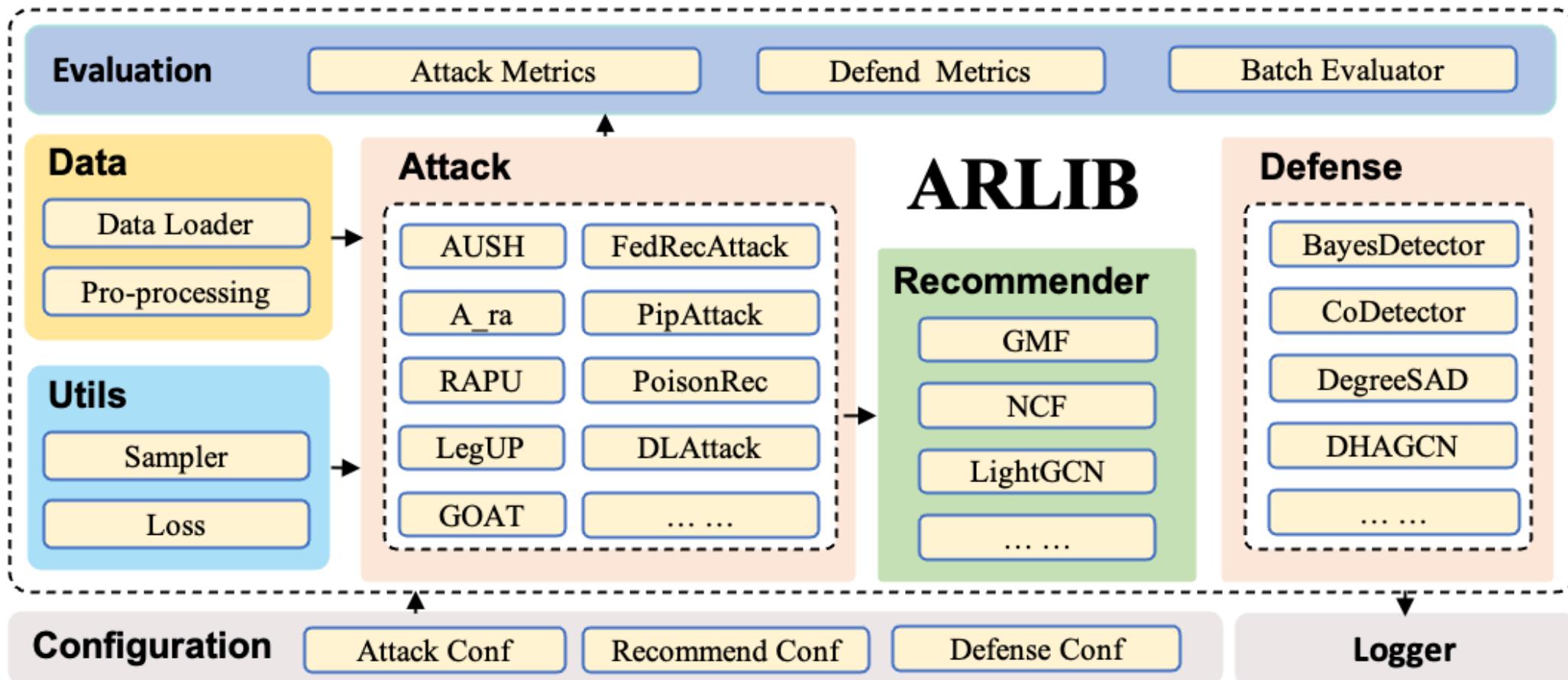
4. Managing Data Noise

5. Limitations and Opportunities

6. Toolkit

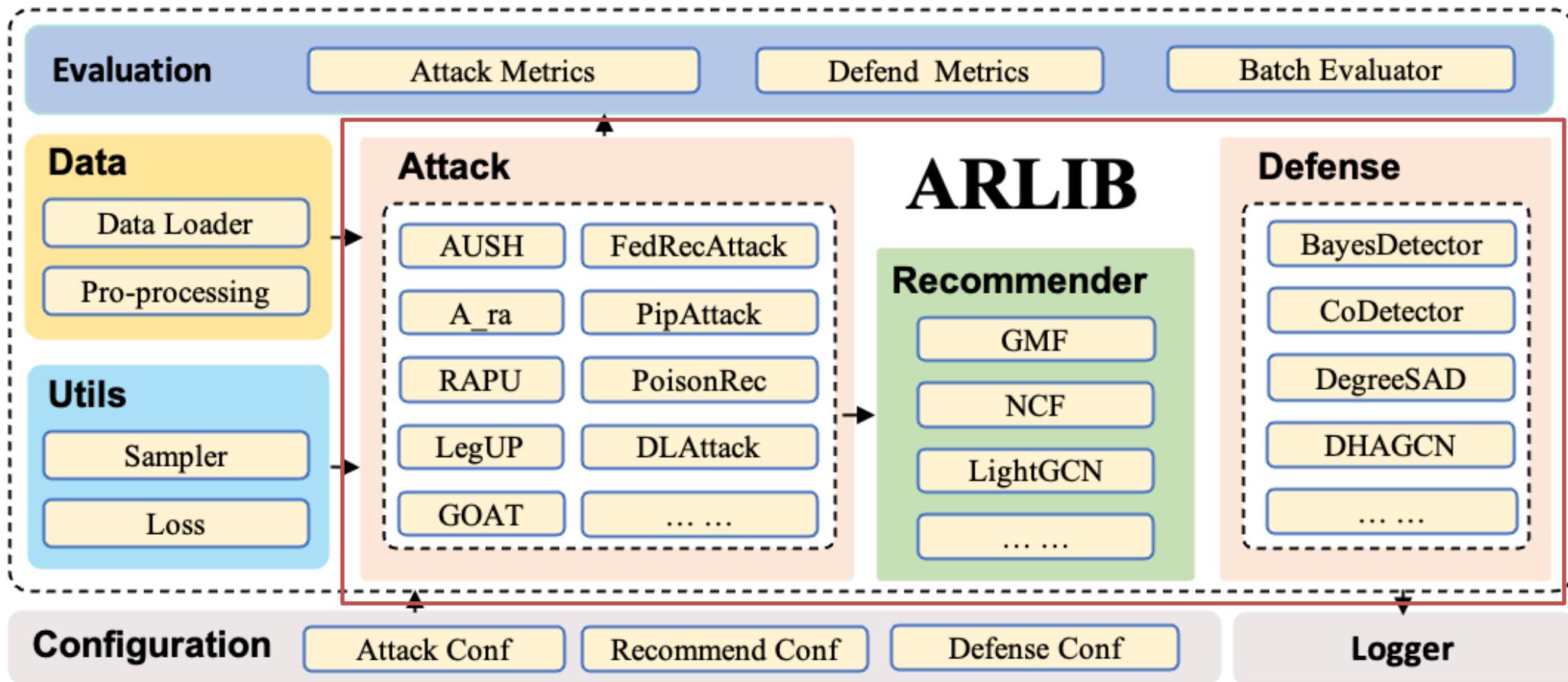
6. Toolkit-ARLib

An open-source framework for conducting data poisoning attacks and corresponding defense on recommendation systems, designed to assist researchers and practitioners.



6. Toolkit-ARLib

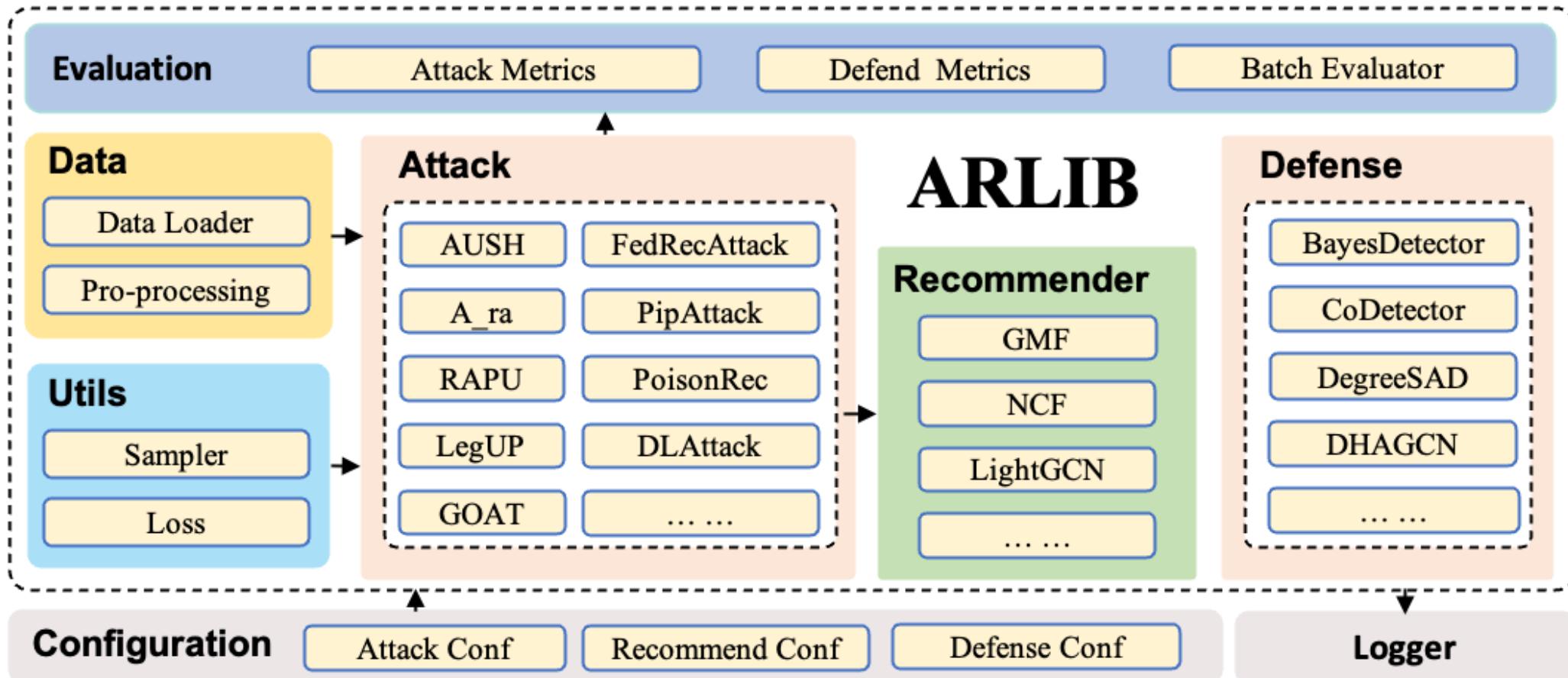
Implement **11** recommendation algorithms, **15** attack methods, and **10** detection methods.



6. Toolkit-ARLib

Features: Fast Implementation / Modularized Architecture

Usages: Comprehensive Assessment / Highly Scalability / Poisonous Data Simulation



6. Dataset

Dataset	#Users	#Itmes	#Interactions	Sparsity	Average action per user	Average action per item	Timestamp	Social data
Last.fm ¹⁰	1,892	17,632	92,834	99.72%	49.07	5.27	✓	✓
Pinterest ¹¹	55,187	9,916	1,500,809	99.73%	27.19	151.35		
Gowalla ¹²	107,092	1,280,969	6,442,892	99.99%	60.16	5.03	✓	✓
Adressa 2M Compact ¹³	15,514	923	2,717,915	81.02%	175.19	2944.65	✓	

Implicit feedback dataset

Dataset	#Users	#Itmes	#Ratings	Rating range	Sparsity	Average action per user	Average action per item	Timestamp	Social data
MovieLens-100K ¹	943	1,682	100,000	1-5	93.70%	106.04	59.45	✓	
MovieLens-1M ¹	6,040	3,706	1,000,209	1-5	95.53%	165.60	269.89	✓	
MovieLens-10M ¹	69,878	10,681	10,000,054	1-5	98.66%	143.11	936.25	✓	
MovieLens-20M ¹	138,493	27,278	20,000,263	0.5-5.0	99.47%	144.41	733.20	✓	
Amazon Review 2014 ²	20,980,000	9,350,000	82,830,000	1-5	99.99%	3.95	8.86	✓	
Yelp ³	1,987,897	150,346	6,990,280	1-5	99.99%	3.52	46.49	✓	✓
Netflix ⁴	480,189	17,770	100,480,507	1-5	98.82%	209.25	5654.50	✓	
Epinions ⁵	22,164	296,277	922,267	1-5	99.99%	41.61	3.11	✓	✓
FilmTrust ⁶	1,508	2,071	35,497	0.5-4.0	98.86%	23.54	17.14		✓
Ciao ⁷	10,877	103,935	269,197	1-5	99.98%	24.75	2.59	✓	✓
CiaoDVD ⁸	17,615	16,121	72,665	1-5	99.97%	4.13	4.51	✓	✓
Book-Crossing ⁹	105,283	340,556	1,149,780	0-10	99.99%	10.92	3.38		

Explicit feedback dataset



重庆大学
CHONGQING UNIVERSITY



THE UNIVERSITY
OF QUEENSLAND
AUSTRALIA



Q&A