Modeling Heterogeneous Statistical Patterns in High-dimensional Data by Adversarial Distributions: An Unsupervised Generative Framework (FIRD)

Han Zhang[1] Wenhao Zheng[3] Charley Chen[1] Kevin Gao[1]

Yao Hu[3] Ling Huang[2] Wei Xu[1]

[1]Tsinghua University [2]AHI Fintech [3]Youku Cognitive and Intelligent Lab, Alibaba Group

Fraud Hurts E-commerce Platform in Many Ways

Fraud Patterns V.S. Normal Patterns [1, 2]

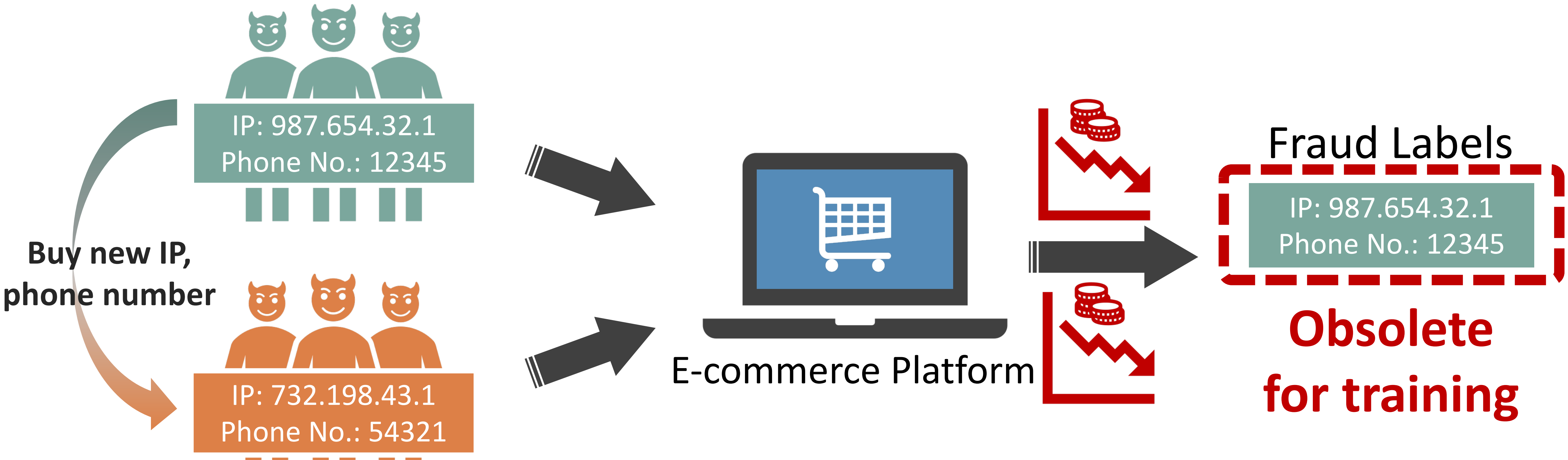
- Fraudsters display **synchronized** behaviors.

- In contrast, normal users are usually **randomly distributed.**

[1] Girish Keshav Palshikar. 2002. The hidden truth-frauds and their control: A critical application for business intelligence. Intelligent Enterprise 5, 9 (2002), 46–51.

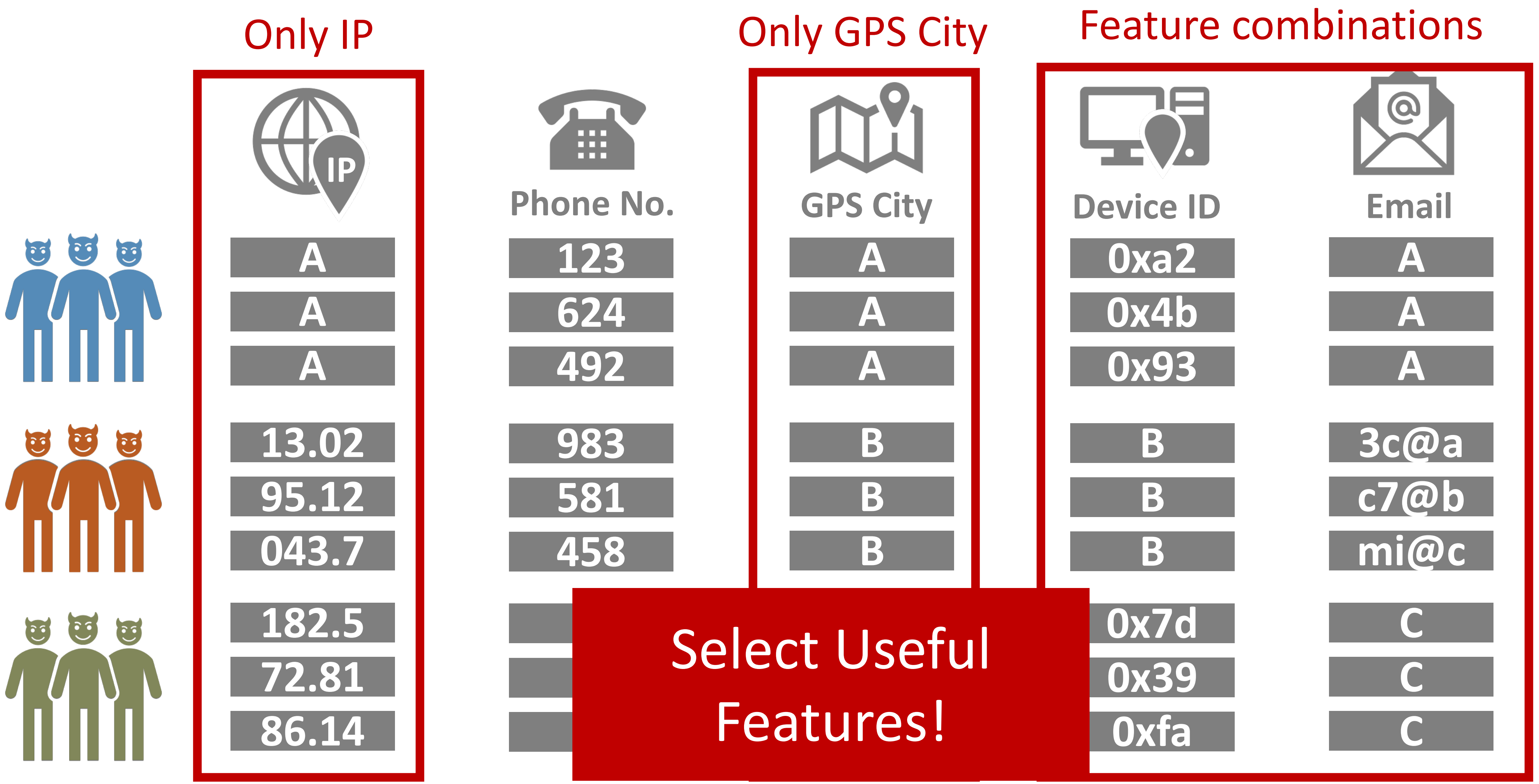
[2] S Benson Edwin Raj and A Annie Portia. 2011. Analysis on credit card fraud detection methods. In 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET). IEEE, 152–156.

Challenge 1: Fraud pattern **changes** after exposure.

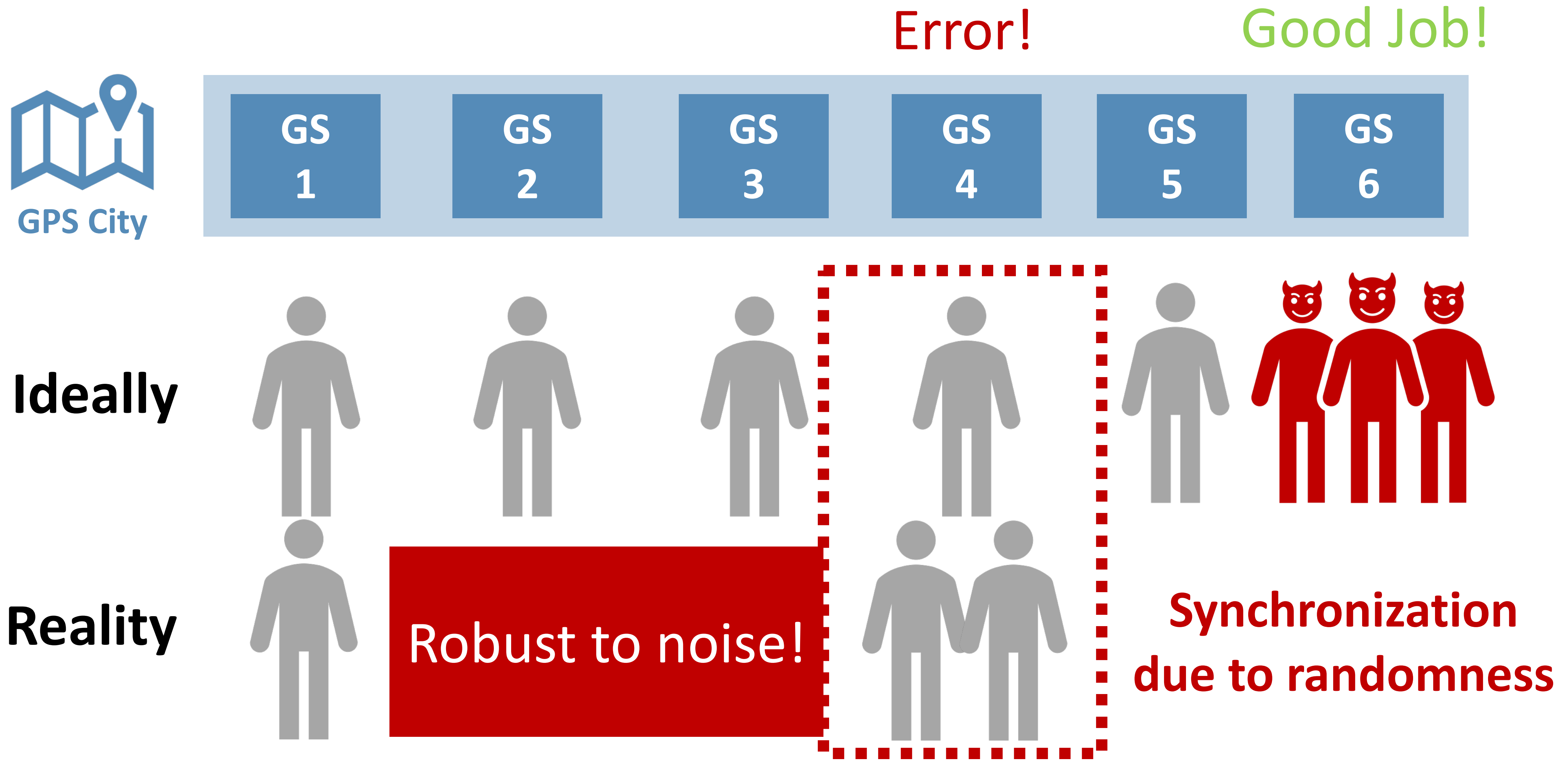


Use **Unsupervised** Methods!

Challenge 2: Different **Local** Clustering Patterns



Challenge 3: Noisy Random Normal Users



Problem Definition – Clustering + Feature Selection

- **Discrete feature space.**
  - Given dataset $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^{N}$, where each feature x_{nm} takes **discrete** values from $\{X_{mi}\}_{i=1}^{D_m}$.
- **Local clustering patterns**.
  - Data points are grouped into **clusters** $\left\{\mathcal{G}_g\right\}_{g=1}^{G}$.
  - Within each cluster $\mathcal{G}_g$, there exists a feature subset $\mathcal{F}_g$, such that $\forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{G}_g, \forall m \in \mathcal{F}_g, x_m = x'_m$ with high probability.
- **Goal**: find all $\mathcal{G}_g$ and $\mathcal{F}_g$, while tolerating the noise.

Key Results

- Applicable to a variety of applications.
  - Fraud detection + anomaly detection.
- Superior fraud detection performance.
  - **18%** AUC improvement.
  - **Interpretable** results.
- Superior anomaly detection performance.
  - Over **5%** AUC improvement in average.
- **Robust** to noise and hyperparameters.

Feature Selection in Clustering

- **Idea**: delete some feature, then cluster the data.
  - No feature should be deleted globally.

Challenge 2: LOCAL clustering patterns!

- 3 types of methods [3]:
  - **Filter model**: filter the low-quality features before clustering.
  - **Wrapper model**: enumerate feature combinations and evaluate clustering performance.
  - **Hybrid model**: select features during clustering.
    - *Suffer from **identifiability issue** in discrete space.

* We provide a proof in our paper.

[3] Salem Alelyani, Jiliang Tang, and Huan Liu. Feature Selection for Clustering: A Review. In Data Clustering: Algorithms and Applications 2013. 29–60.

Dense Block Detection

- **Idea**: high-density blocks in data are potential anomalies [4, 5].
- **Steps**:
  1. Greedy search for the block with highest density.
  2. Delete the block.
  3. Repeat the process on the remaining data.

Challenge 3: Noise!

- Normal users with random synchronization significantly affect the detection performance.

[4] Kijung Shin, Bryan Hooi, and Christos Faloutsos. M-Zoom: Fast Dense-Block Detection in Tensors with Quality Guarantees. ECML PKDD 2016. 264–280.

[5] Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos. D-Cube: Dense-Block Detection in Terabyte-Scale Tensors. WSDM 2017, 681–689.

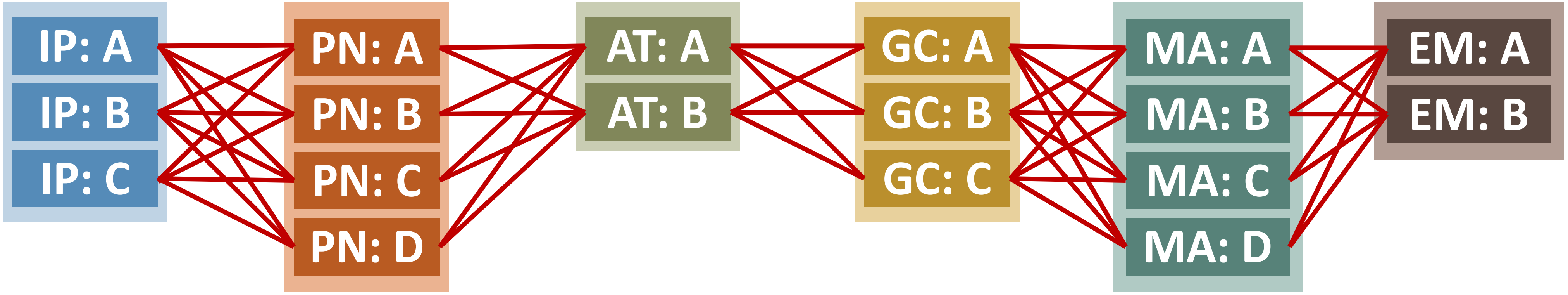
FIRD: A Generative Probabilistic Model

Feature **I**ndependence and adve**R**ersarial **D**istributions.

Enumerating Possible Feature Combinations?

ⓧ **Exponential** feature combinations.

ⓧ **Exponential** feature value combinations.



A Decomposed Way of Feature Selection

✓ Conditional feature independence.

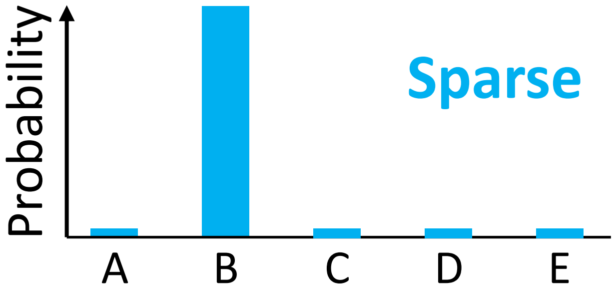
- Features are independent within a cluster.
- Linear complexity.

✓ Recognize clustering pattern on **each** feature, then combine.

- Using the **adversarial distributions** to fit the data.

Fitting Patterns Using Adversarial Distributions in Each Feature

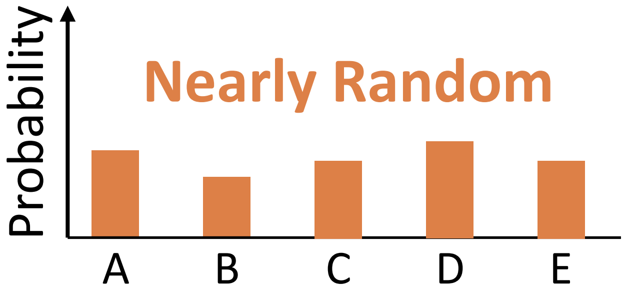
- For **synchronized** features in a cluster



(B, B, B, E

Solved Challenge 2: Detecting Local Clustering Patterns!

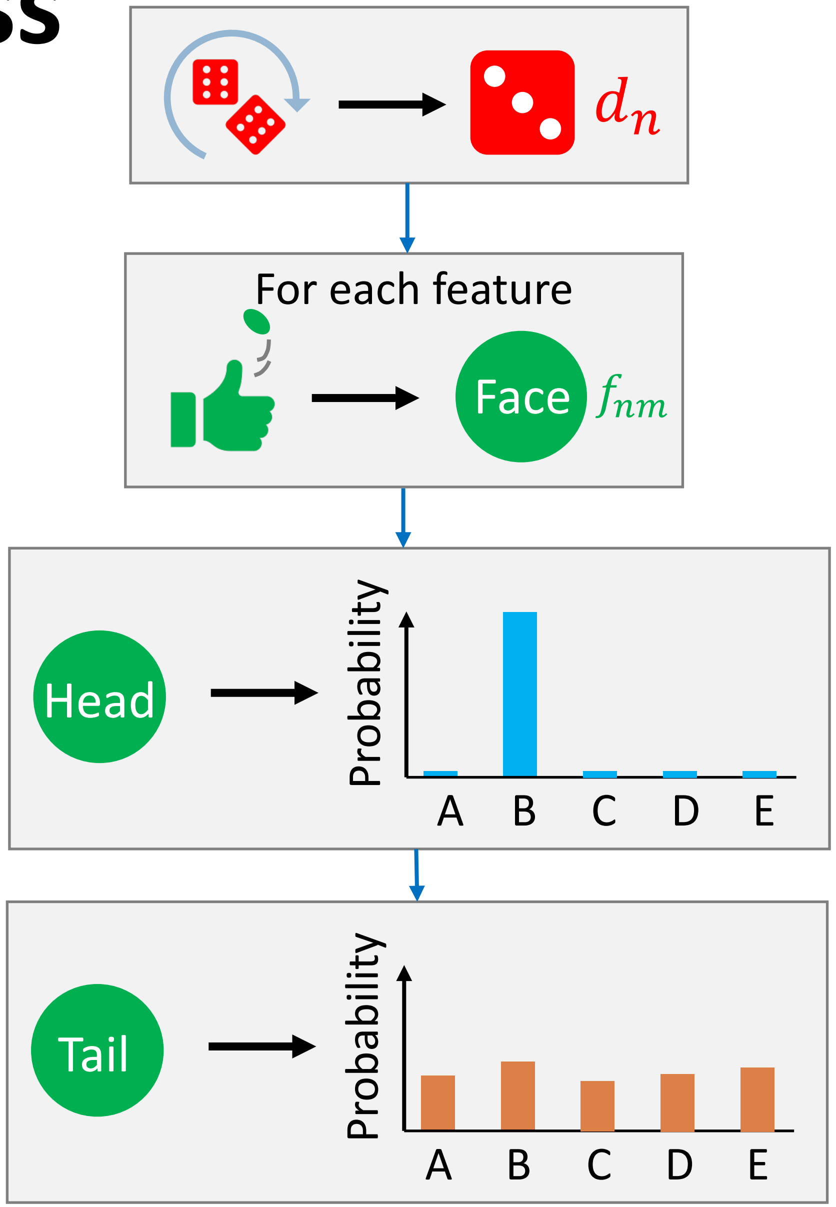
- For **non-synchronized** features in a cluster



(A, D, C, B, E, …)

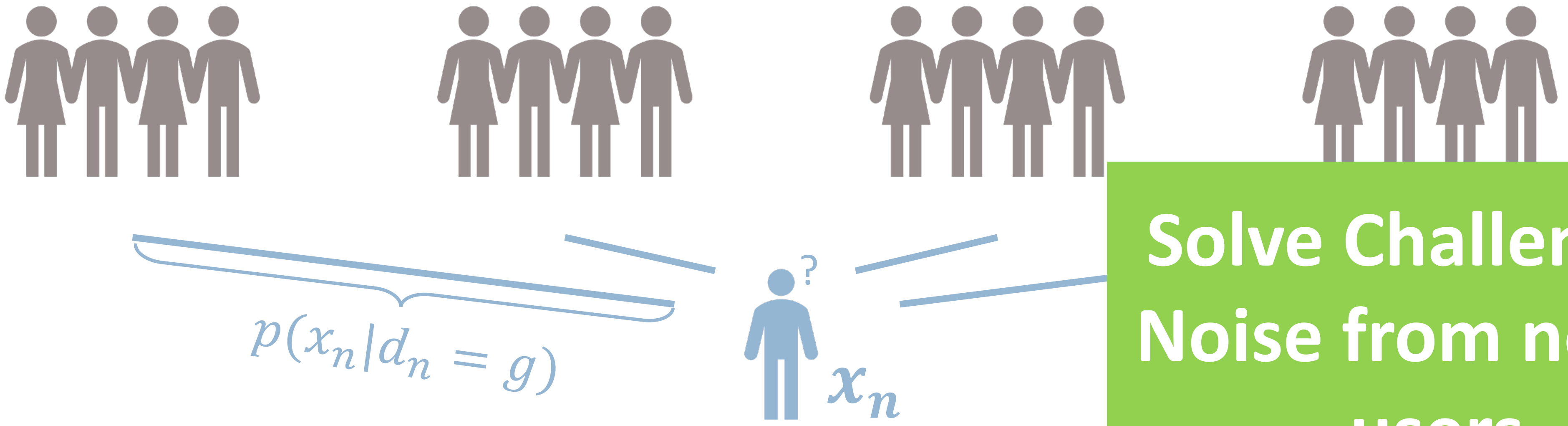
Observation Generation Process

- Choose a cluster $d_n \sim \mathrm{Multinomial}(\boldsymbol{\pi})$
  - For each feature m:
    - Choose indicator variable $f_{nm} \sim Bernoulli(\boldsymbol{\mu}_{d_n})$
    - If $f_{nm} = 1$, generate observation x_{nm} from **sparse** multinomial distribution.
    - If $f_{nm} = 0$, generate observation x_{nm} from **nearly random** multinomial distribution.



Noise Reduction

- **Noise**: outliers that are **unsimilar** to all clusters.



Solve Challenge 3: Noise from normal users.

- An information-theoretic rule to recognize an

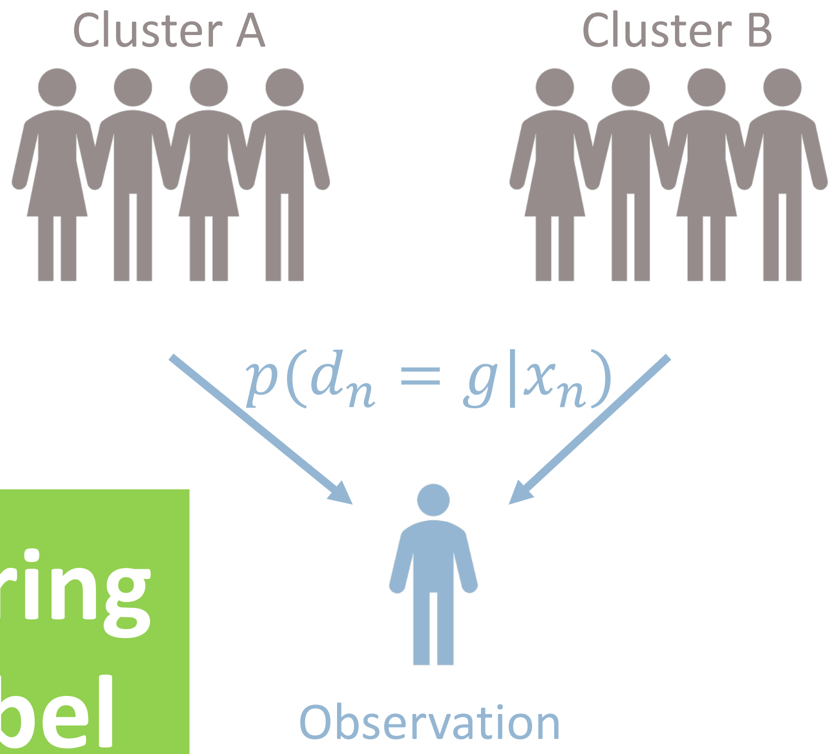
$$I(x_n|d_n = g) = -\log p(x_n|d_n = g) < (1 + \epsilon)H[p(x_n|d_n = g)]$$

Probabilistic Inference Based on FIRD

- Inferring label ℓ for each observation given the label of each cluster.

$$\ell_n \triangleq \mathbb{E}_{d_n}[\ell | x_n] = \sum_{g=1}^{G} p(\ell | d_n = g) p(d_n = g | x_n)$$

- Label of clusters $p(\ell | d_n = g)$ are easier to obtain:
  - #Clusters << #Observations
  - Cluster patterns are easier t



From Clustering to Fraud Label Assignment

Experimental Evaluations

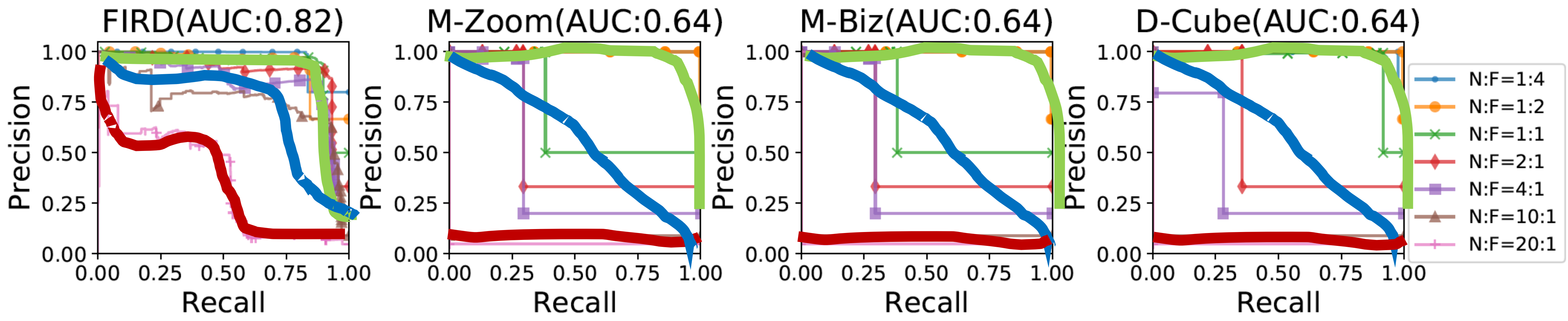
Our Cython code of FIRD is available at https://github.com/fingertap/fird.cython.

Identify Fraud Groups

- Dataset
  - We collect the registration records from an E-commerce platform.
  - An account is labeled as **Fraud** if any malicious behavior is observed.
    - Labels are used only for evaluation.
- Objective
  - Good performance.
  - High interpretability.

Identify Fraud Groups - Performance

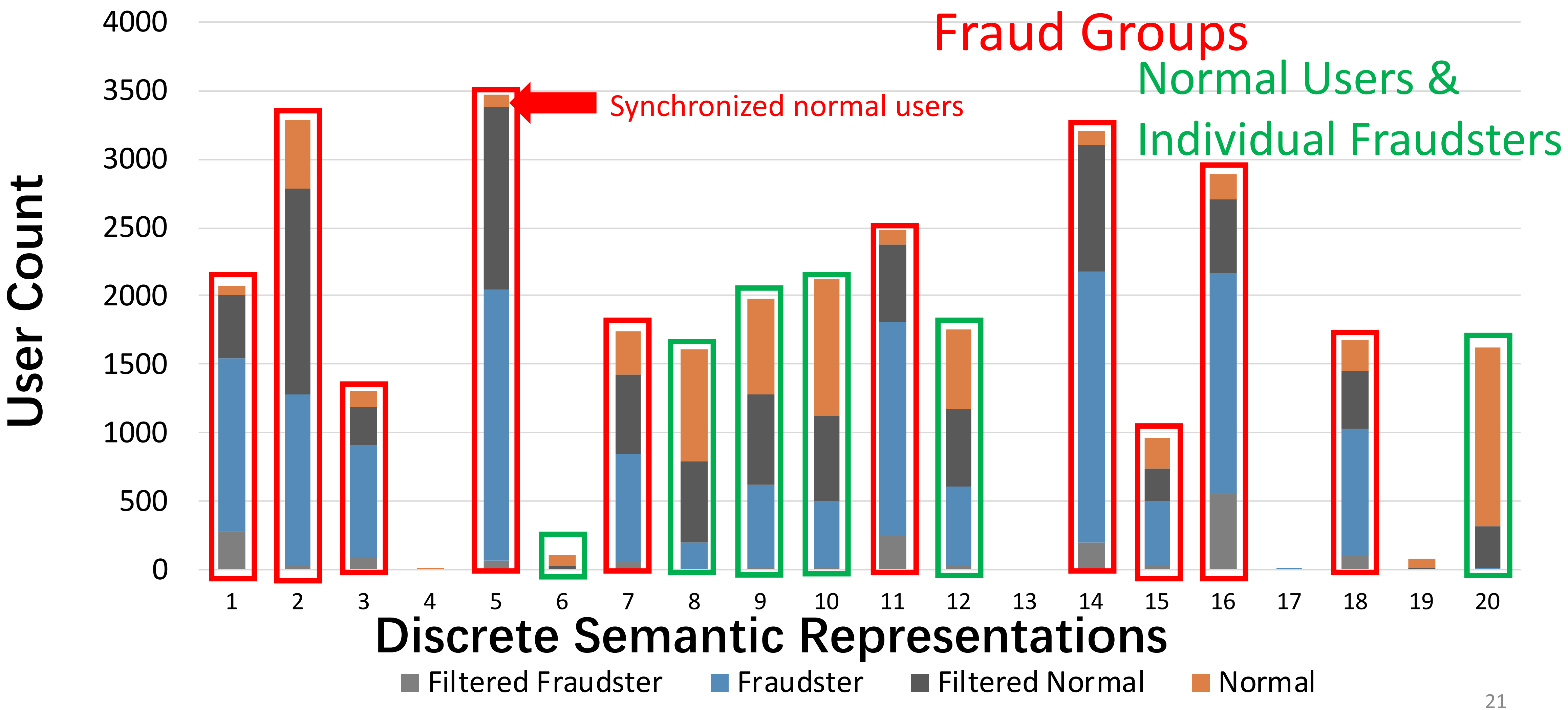
- Compare with dense block detection methods [2, 3]:



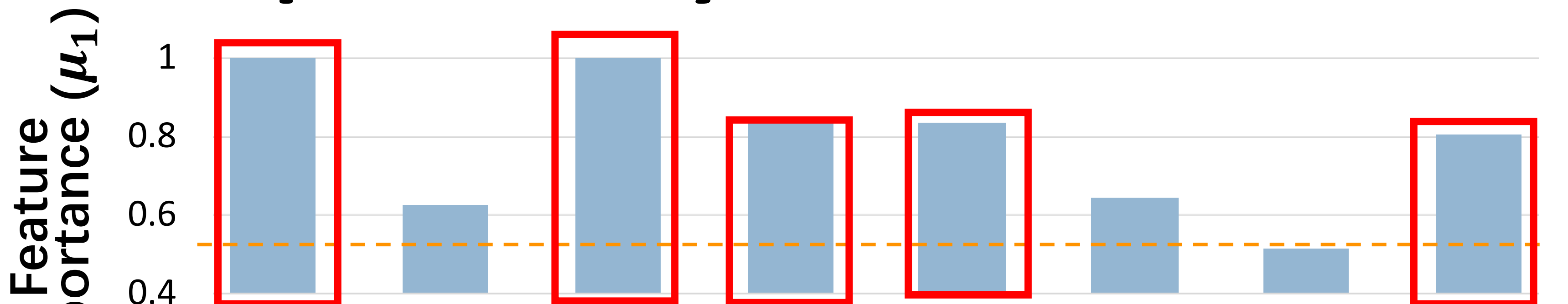
- **N:F** is the fraction between normal user and fraudsters.
- Higher N:F means larger noise.

18% AUC ↑
Robust to noise!

Interpretability: Visualize Detected Clusters



Interpretability: Visualize One Fraud Cluster



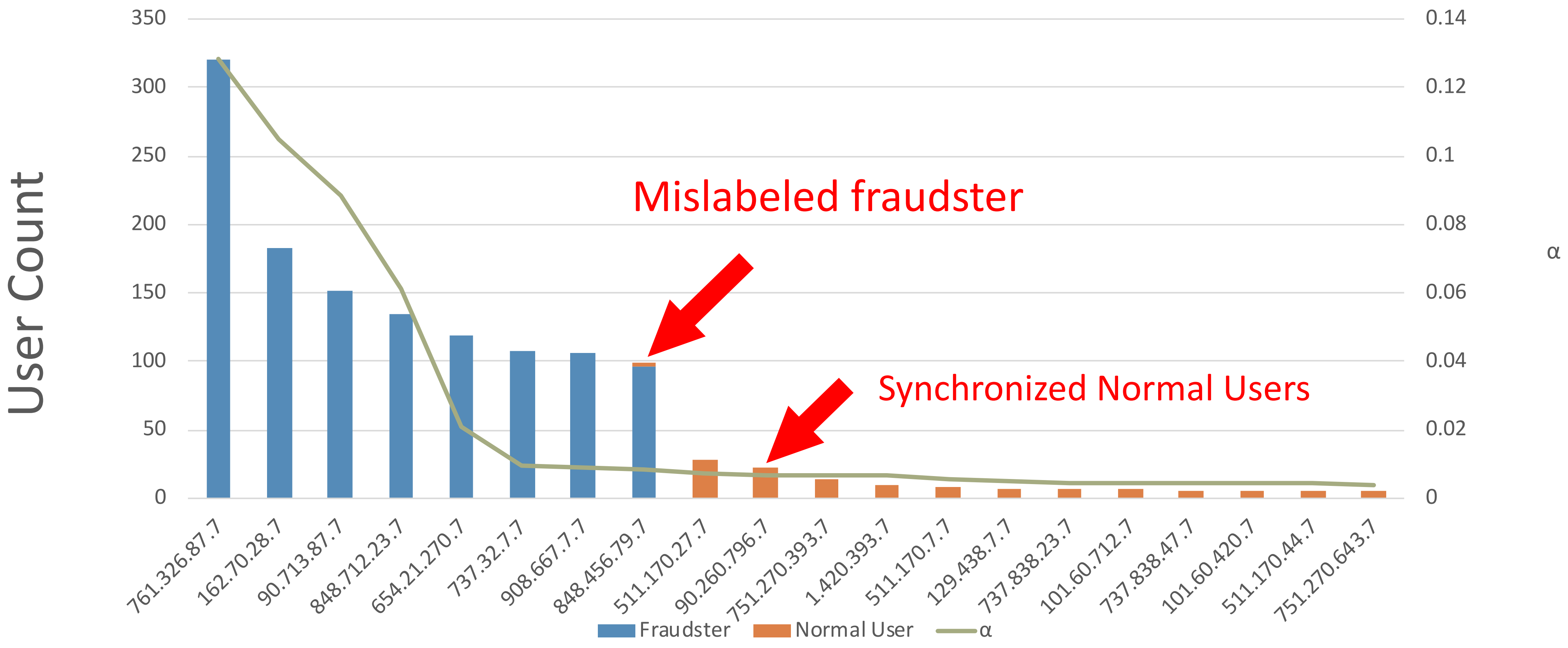
Fraud Signature

10 Random Samples

Channel	Device ID	Time	IP	IP City	Phone	Phone City	OS Type
B	180061	5	737.32.7.7	Coryborough	7037671	West Kristen	sim
B	405376	5	162.70.28.7	Amandaview	[illegible]916214	New Mariafurt	android
B	861328	5	162.70.28.7	Amandaview	[illegible]320211	East Erika	sim
B	201199	5	848.712.23.7	Port Heather	6571178	Valerieside	android
B	162176	15	761.326.87.7	Luisstad	[illegible]064801	Thompsonbury	android
B	498726	5	761.326.87.7	Luisstad	[illegible]932753	Edwardsfurt	android
B	893969	5	654.21.270.7	Luisstad	6699477	New Mariafurt	android
B	195884	5	654.21.270.7	Luisstad		New Robertland	android
B	221445	5	654.21.270.7	Luisstad	2611409	West Kellyport	android
B	148534	5	90.713.87.7	Luisstad	2999196	West Kristen	android

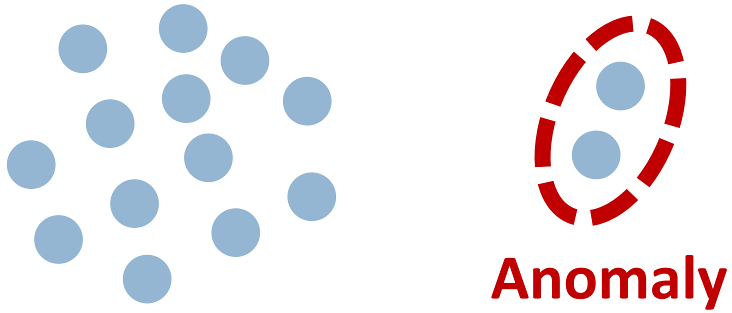
3 Fraud groups

Interpretability: Visualize One Fraud Feature



Anomaly Detection

- **Assumption**: anomalies are **distant** from the data manifolds [9].



- **Feature selection idea**: subsampling and ensemble.
- Still enumerating the exponentially many feature combinations.

[9] Yue Zhao, Zain Nasrullah, Maciej K. Hryniewicki, and Zheng Li. LSCP: Locally Selective Combination in Parallel Outlier Ensembles. SDM 2019. 585–593.

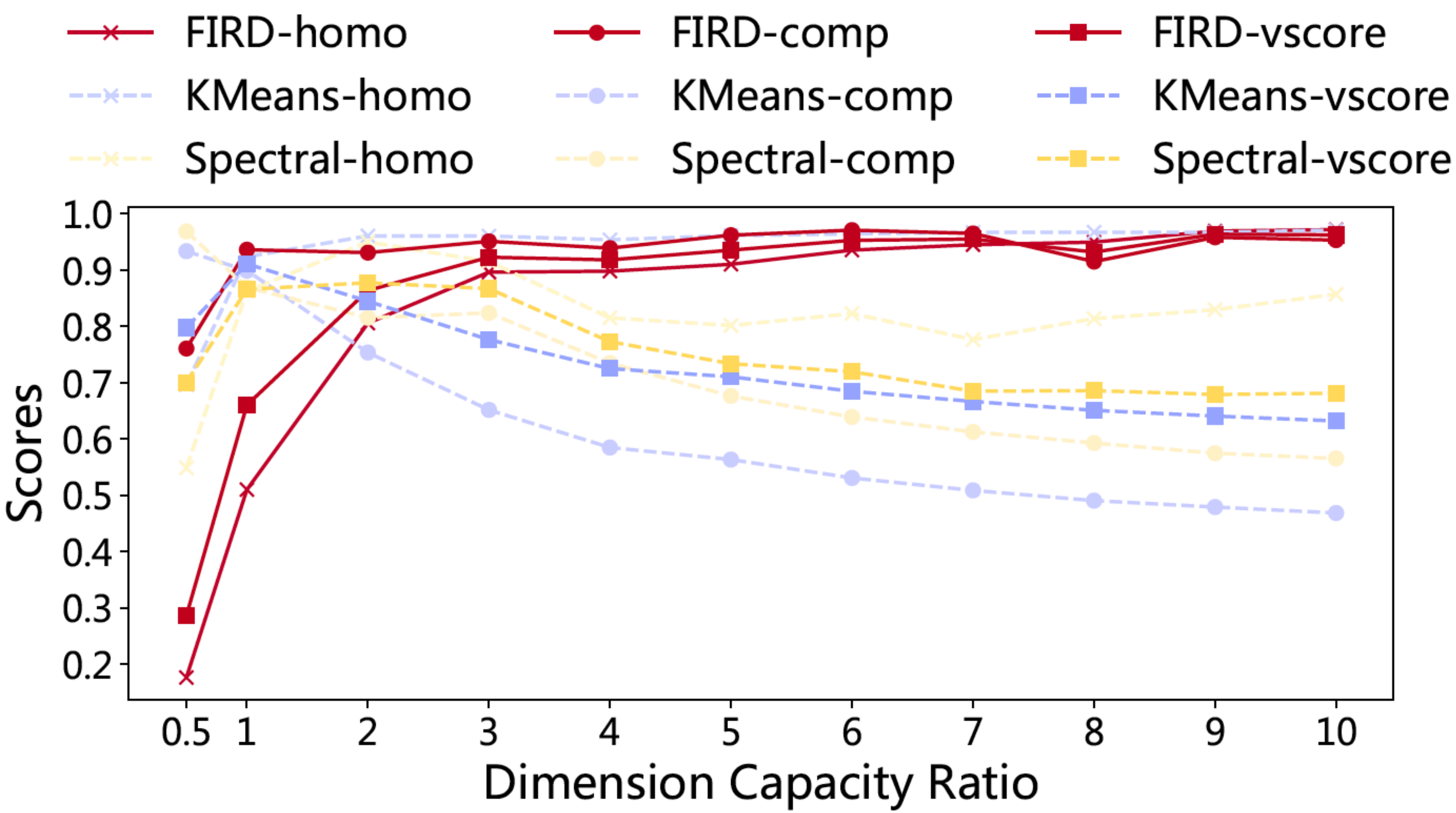
Comparison with SOTA Methods

Dataset	FIRD	HBOS	IForest	OCSVM	LSCP
cardio	**0.949**	0.843.	0.924	0.938	0.901
musk	**1.000**	**1.000**	0.999	**1.000**	0.998
optdigits	**1.000**	0.865	0.714	0.500	-
satimage-2	**0.998**	0.977	0.993	0.997	0.9935
shuttle	0.990	0.986	**0.997**	0.992	0.5514
satellite	**0.900**	0.754	0.701	0.660	0.6015
ionosphere	**0.946**	0.5569	0.8529	0.8597	-
pendigits	**0.972**	0.9247	0.9435	0.931	0.8744
wbc	0.944	**0.954**	0.9325	0.9376	0.945

Local Clustering Pattern **matters** in various cases!

- More benchmark results are available at PyOD benchmark.

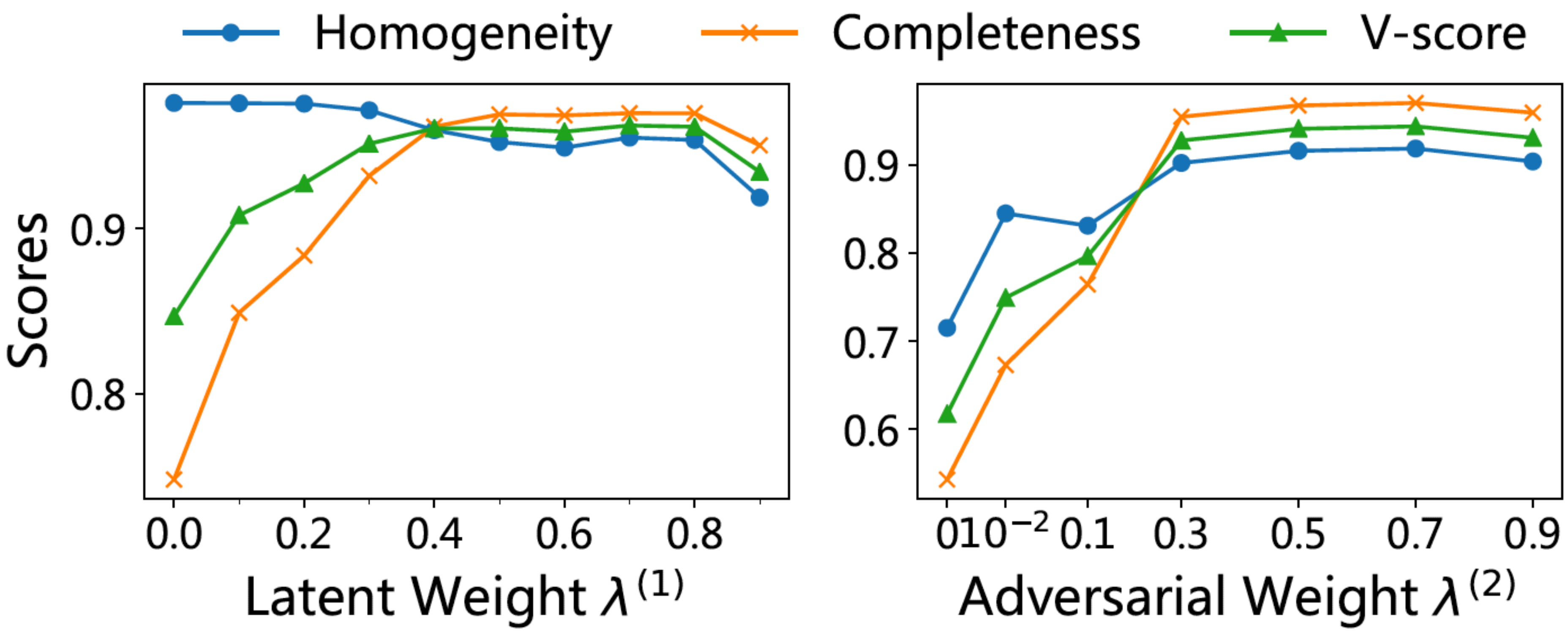
Model Analysis – #Clusters: G



Just choose a larger G

*__Dimension Capacity Ratio__: the ratio of the parameter G to the ground-truth number of clusters.

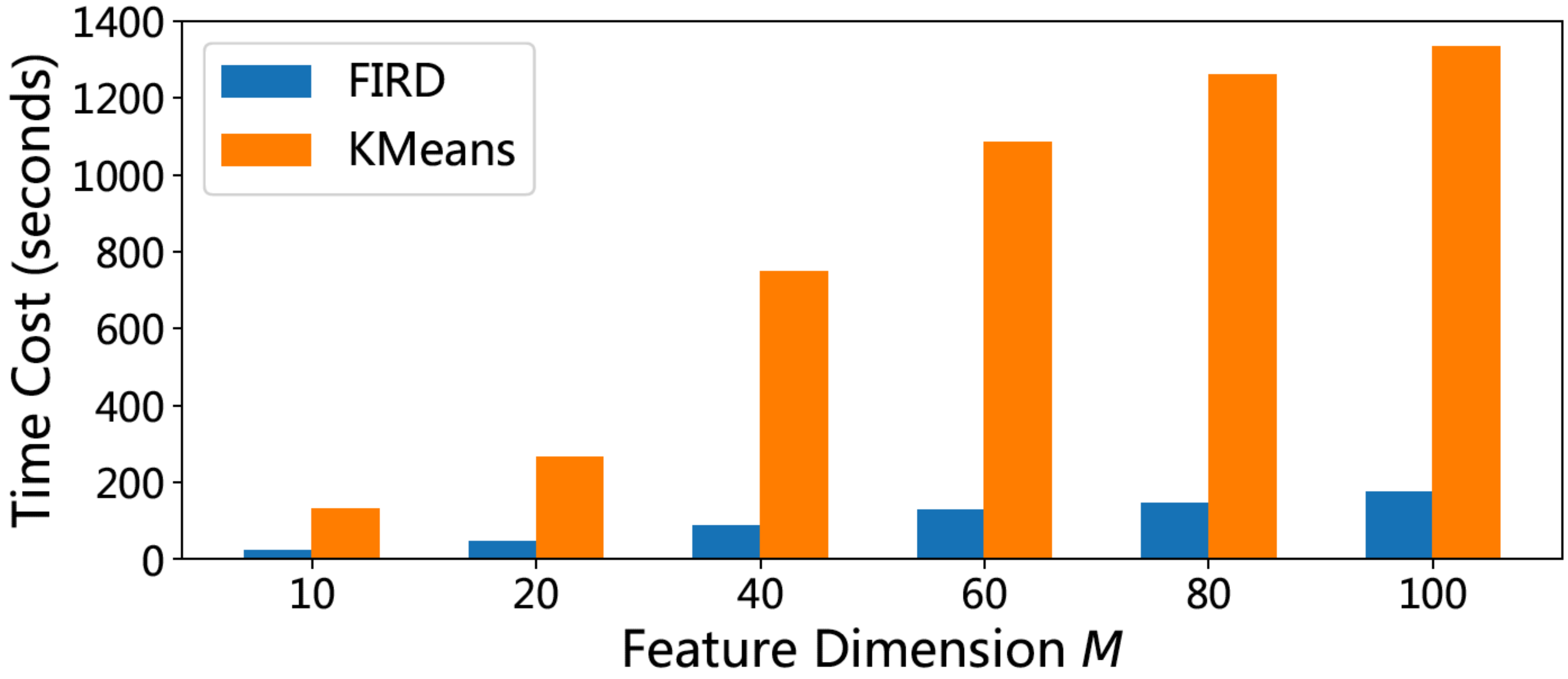
Model Analysis – Regularizer Weight: λ



Just choose a relatively larger λ

*$\lambda^{(1)}$ controls selecting effective clusters. $\lambda^{(2)}$ controls adversarial distributions.
*$0 < \lambda^{(1)}, \lambda^{(2)} < 1$, poorer regularization effect near the border (0 and 1).

Model Analysis – Running Time



Linear running time w.r.t M

*We compare with the K-Means implemented in the Python package Scikit-Learn.
*Fix the #samples and the #values in each feature.

Conclusion

- Fraud groups display synchronized behaviors on a subset of features.
- Use adversarial distributions to select useful features by competing.
- Identifying local clustering patterns benefits various applications.
  - Up to **18%** increase on fraud detection and **5%** on anomaly detection.

Thank you!

Q&A