AN INVERTIBLE GRAPH DIFFUSION NEURAL NETWORK FOR SOURCE LOCALIZATION

Junxiang Wang, Junji Jiang, and Liang Zhao

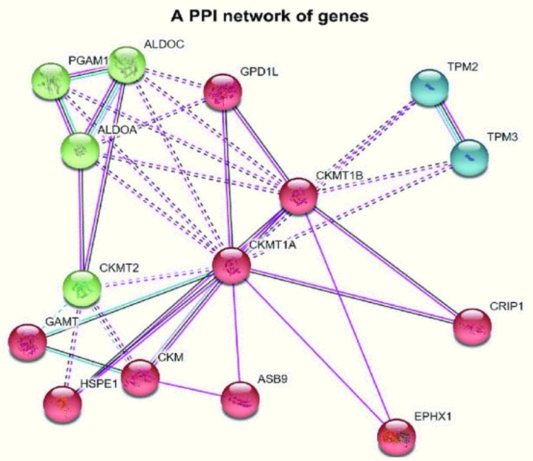
The Web Conference (WebConf) 2022

Content

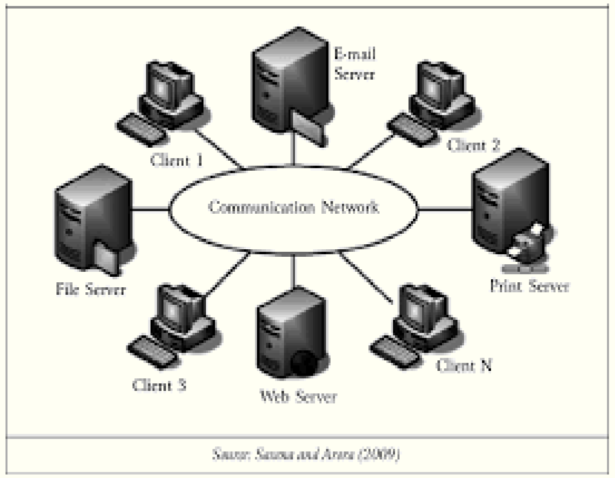
Background: Graph Structures are Prevalent.

- Graphs are so prevalent that they have been applied in various applications.

(a). Social Network.



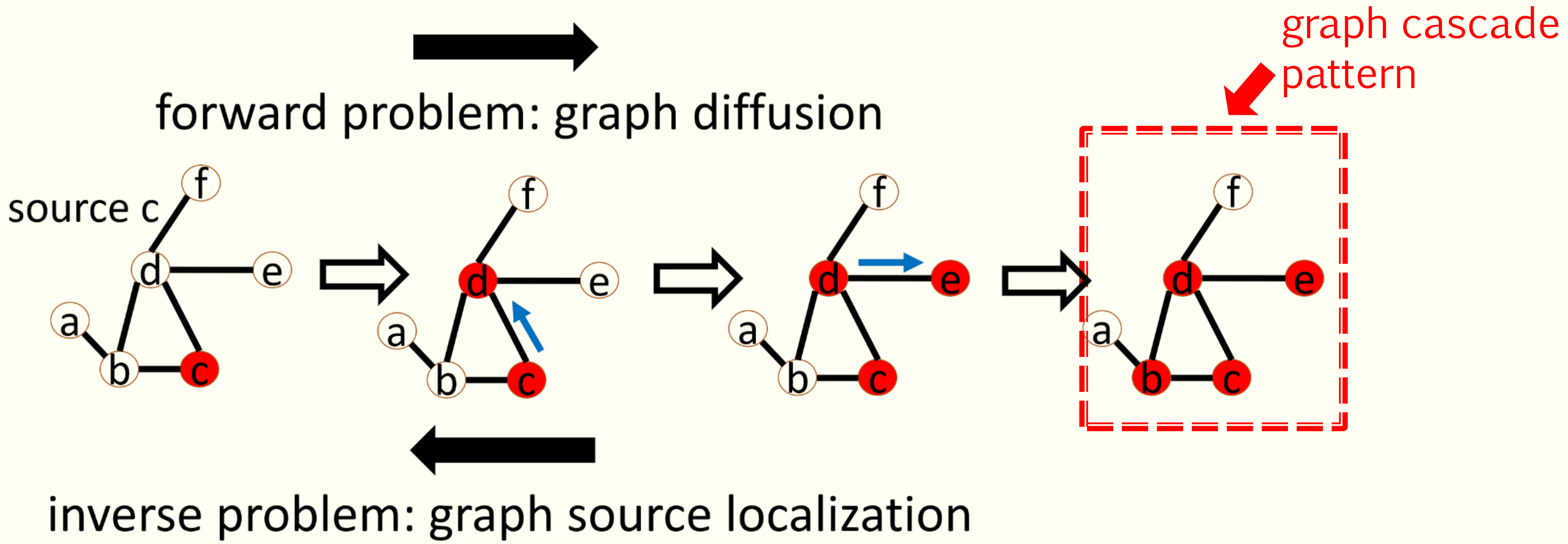
(b). Biological Network.



(c). Computer Network.

Background: Graph Diffusion and Source Localization.

- Graph diffusion aims to predict future graph cascade patterns given source nodes.
- Source localization aims to detect source nodes given their future graph cascade patterns.


graph cascade pattern
forward problem: graph diffusion
source c
f
d
e
a
b
c
inverse problem: graph source localization

Background: Existing Works are Prescribed Methods.

- Source localization models are dominated via traditional prescribed methods, which are based on predefined rules and usually specialized for specific applications.

Problem: What if prior diffusion knowledge is unavailable?

Solution: Graph Neural Networks (GNNs) can be an alternative to prescribed methods because of two advantages:

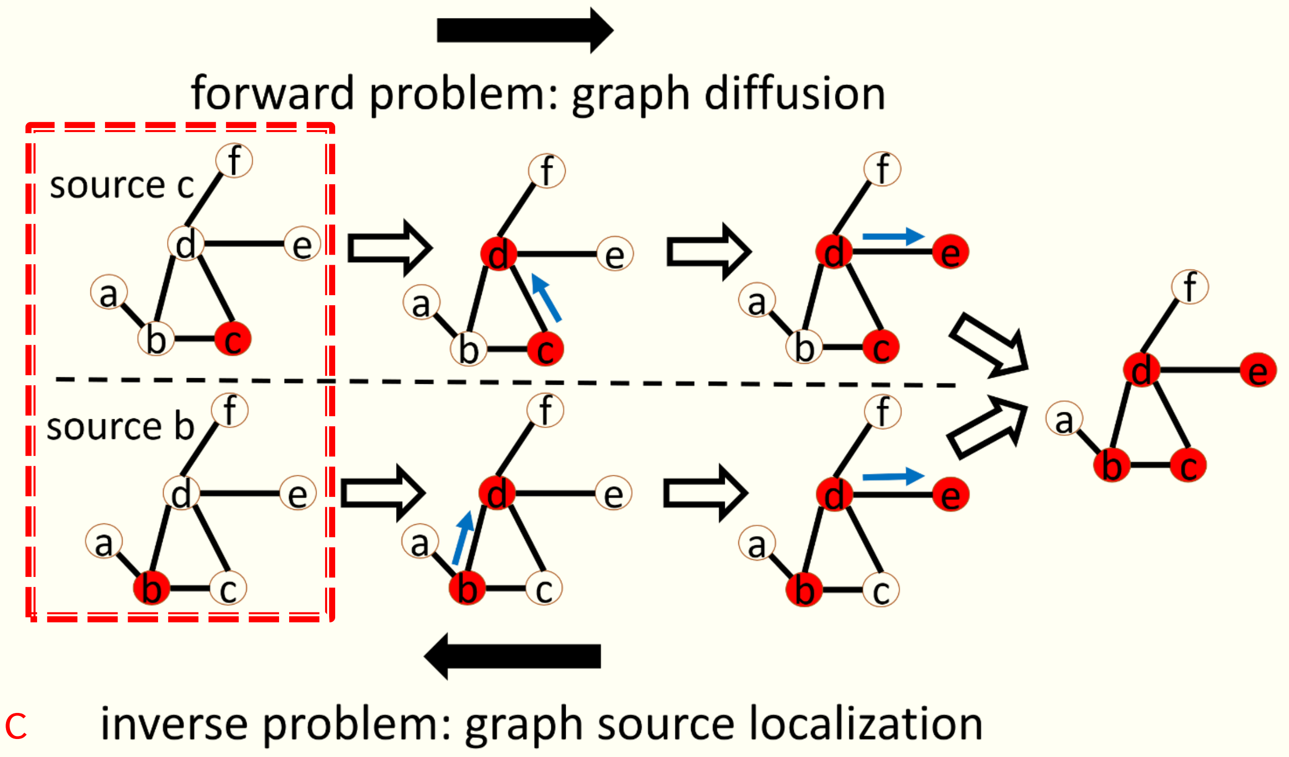
1. GNNs can incorporate node attributes into models and learn node representations effectively by capturing network topology and neighboring information.

2. GNNs can "learn" rules from the data in an end-to-end fashion.

Any ***Challenge*** if GNNs are applied to source localization?

Background: Challenges of Source Localization(1).

1. Difficulty to leverage knowledge in graph diffusion models.


forward problem: graph diffusion
source c
source b
f
d
e
a
b
c
inverse problem: graph source localization

prior knowledge from graph diffusion

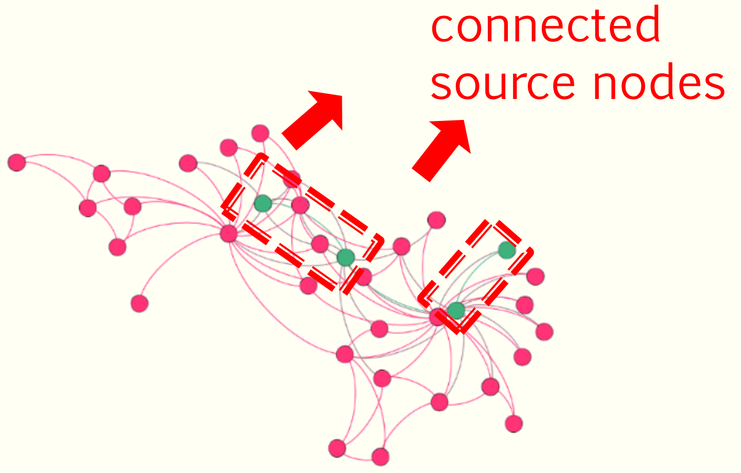
Both nodes b and c generate {b,c,d,e}

It is prohibitively difficult to define hand-crafted rules to inject prior knowledge.

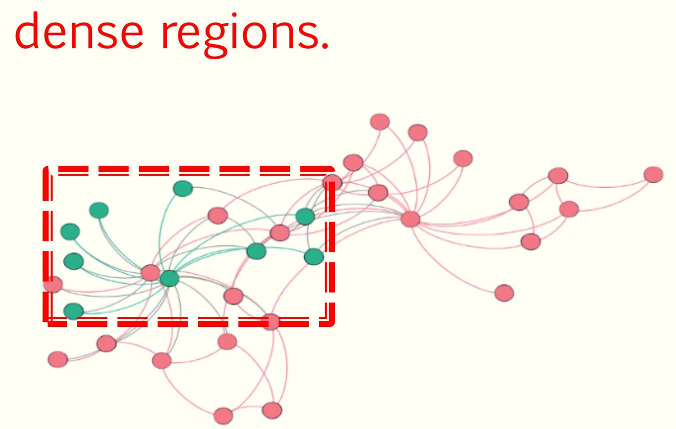
Background: Challenges of Source Localization(2).

2. Difficulty to ensure the validity of the inferred sources.

- Graph sources usually follow validate graph patterns:



(a). Source of misinformation are connected in the social networks.



(b). Sources of malware are dense in some restricted regions of the IoT networks.

Validity constraints require the projection of **MULTIPLE** sources, rather than an individual node.

Problem Formulation: Mathematical Notations.

- Given a graph $G = (V, E)$, assume the graph diffusion model $\theta\colon \mathrm{x} \rightarrow Y_T$ is known.

➢ Then the goal of source localization is to learn the mapping

$$\theta^{-1}\colon \mathrm{Y_T} \rightarrow x \;\; s.t.\, \Phi(x) = 0.$$

➢ Challenges:

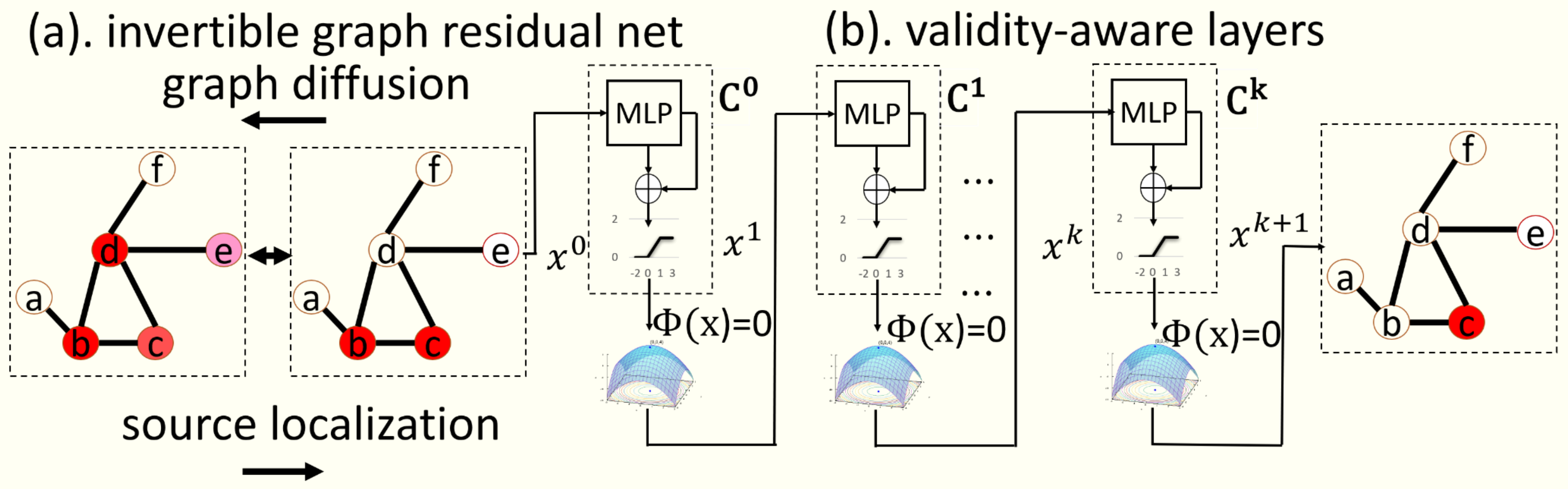
1. The difficulty to integrate information from θ to θ^{-1}.

2. The difficulty to incorporate $\Phi(x) = 0$ into θ^{-1}.

Notation	Definition
V	node set.
E	edge set.
$\Phi(x) = 0$	validity constraint.
S	source set.
T	leap time.
$Y_{t,i}$	whether node i at time t is diffused.
x_i	whether node $i \in S$.

Our method: Invertible Validity-aware Graph Diffusion (IVGD).

- The proposed framework IVGD consists of two modules:



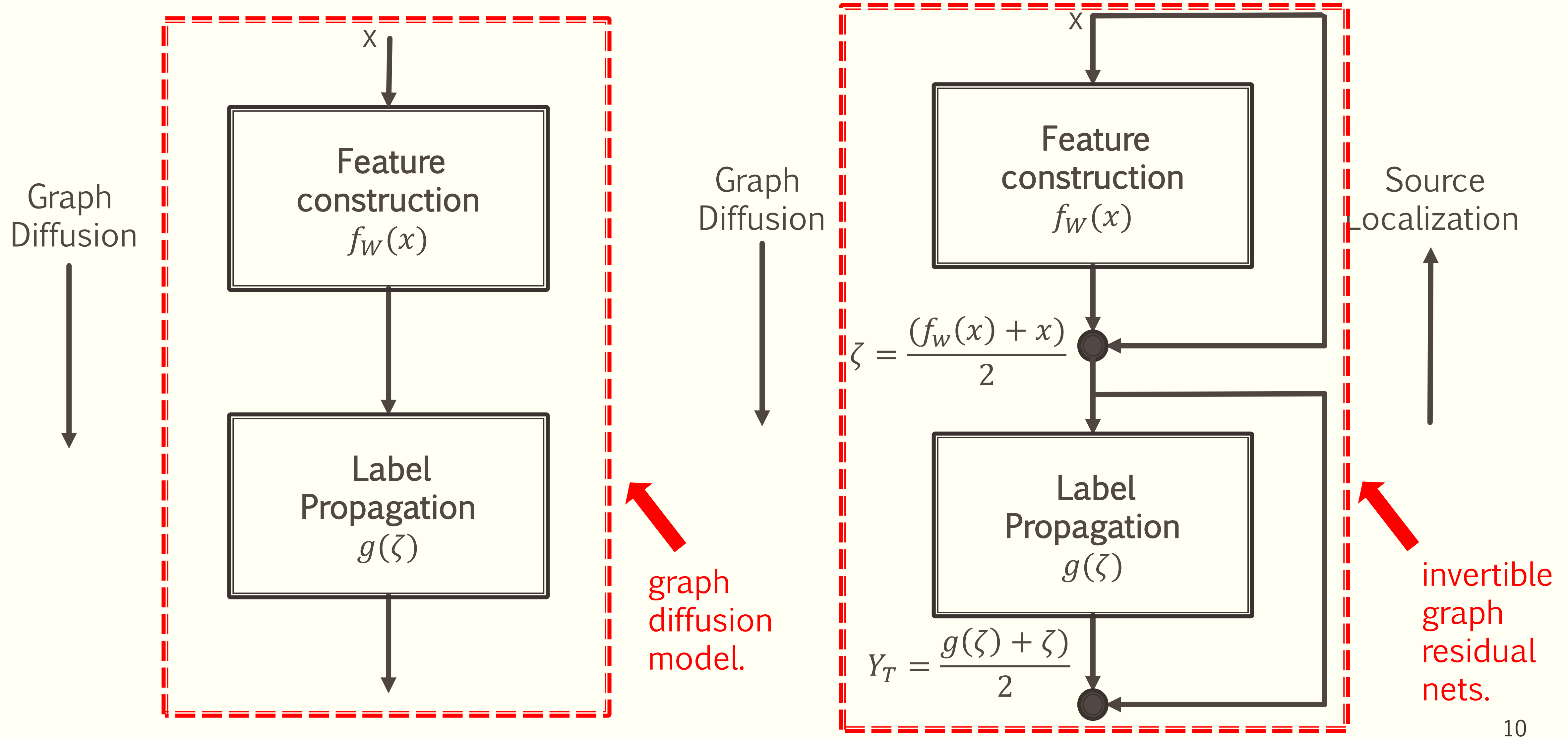
- The invertible graph residual net ➡ Challenge 1

It reverses the graph diffusion models via formulating the graph residual net.

- Validity-aware layers ➡ Challenge 2

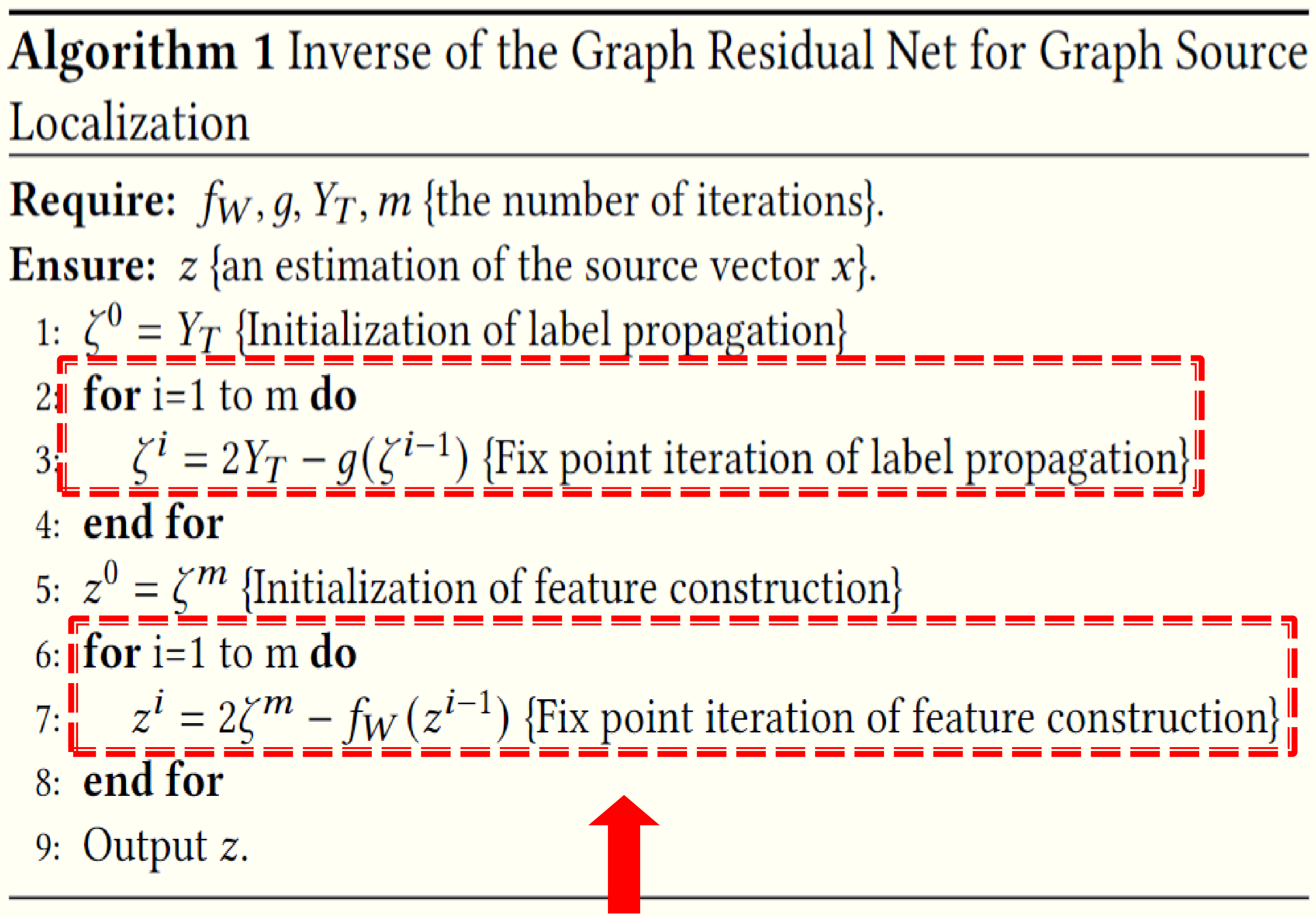
They encode validity constraints into problems with unrolled optimization techniques.

Our method: Invertible Graph Residual Net(IGRN)(1)



Graph Diffusion
x
Feature construction $f_W(x)$
Label Propagation $g(\zeta)$
graph diffusion model.
Graph Diffusion
x
Feature construction $f_W(x)$
$\zeta = \frac{(f_W(x) + x)}{2}$
Label Propagation $g(\zeta)$
$Y_T = \frac{g(\zeta) + \zeta)}{2}$
Source Localization
invertible graph residual nets.

Our method: Invertible Graph Residual Net(IGRN)(2)



Algorithm 1 Inverse of the Graph Residual Net for Graph Source Localization

```
Require: $f_W, g, Y_T, m$ {the number of iterations}.
Ensure: $z$ {an estimation of the source vector $x$}.
$\zeta^0 = Y_T$ {Initialization of label propagation}
for i=1 to m do
    $\zeta^i = 2Y_T - g(\zeta^{i-1})$ {Fix point iteration of label propagation}
end for
$z^0 = \zeta^m$ {Initialization of feature construction}
for i=1 to m do
    $z^i = 2\zeta^m - f_W(z^{i-1})$ {Fix point iteration of feature construction}
end for
Output $z$.
```

The inverse of label propagation

The inverse of feature construction

Our method: The Invertibility of the Graph Residual Net.

- How to ensure the invertibility theoretically?

Assume P is an invertible graph residual net, then we have

Theorem 1: The graph residual net is invertible if $L_f < 1$ and $L_g < 1,$ where L_f and L_g are Lipschitz constants of f_W and $g(\zeta)$, respectively.

Lemma 2: Let L_P and $L_{P^{-1}}$be Lipschitz constants of P and P^{-1}, respectively, then $L_P \leq \frac{(1+L_f)(1+L_g)}{4}$, and $L_{P^{-1}} \leq \frac{4}{(1-L_f)(1-L_g)}$.

- How to ensure the invertibility practically?

➢ For feature construction f_w, the power iteration method can be utilized to satisfy Theorem 1.

➢ For label propagation $g(\zeta)$, common propagation functions such as Independent Cascade (IC) model satisfy Theorem 1.

Our method: Validity-Aware Layers(1).

- Let $z = P^{-1}(Y_T)$, $\mathrm{x} = C(z)$ where P^{-1} is the inverse of graph residual net, and C is an error correction module.

- The validity constraint $\Phi(x) = 0$ is required to satisfy. Assume that $R(x)$ is a loss function, we address the following optimization problem:

$$\min_x R(x) \quad s.t.\ x = C(z),\ \ \Phi(x) = 0. \quad (1)$$

- Due to the potential conflict between $x = C(z)$ and $\Phi(x) = 0$, we relax Equation (1) as follows:

$$\min_x R(x) + \boxed{\frac{\tau}{2}\|x - C(z)\|_2^2}\ s.t.\ \Phi(x) = 0. \quad (2)$$

least square loss

where τ is a tuning parameter. Now the task is to design activation layers to address Equation 2.

Our method: Validity-Aware Layers(2).

- To achieve this, we unroll Equation (2) into a neural net, where each layer is designed as follows:

$$x^{k+1} \leftarrow \min_x J^k(x) \quad s.t.\ \Phi(x) = 0. \quad (3)$$

where $J^k(x) = R(x) + \frac{\tau^k}{2}\left\|x - C^k\left(x^k\right)\right\|_2^2$ and $x^0 = z$.

- To solve Equation (3), the augmented Lagrangian is formulated as follows:

$H^k(x,\lambda) = J^k(x) + \Psi^k(x,\lambda)$ where $\Psi^k(x,\lambda) = \frac{1}{2\rho^k}\left(\lambda + \rho^k\Phi(x)\right)^2 - \lambda^2$, and λ is a dual variable.

Validity-aware layers are designed as follows:

$$x^{k+1} \leftarrow \operatorname*{argmin}_x \mathrm{H}^k(x)$$
$$\lambda^{k+1} \leftarrow \lambda^k + \rho^k\, \Phi(x^{k+1}).$$

- The linearization technique is utilized to transform the nonconvex $\mathrm{H}^k(x,\lambda)$ to the convex $h^k(x)$

$h^k(x) = J^k(x) + \partial_x\left(\Psi^k\right)^T\left(x^k,\lambda^k\right)\left(x - x^k\right) + \frac{\alpha^k}{2}\left\|x - x^k\right\|_2^2$ where $\alpha > 0$ is a tuning parameter.

Finally, validity-aware layers are designed as follows:

$$x^{k+1} \leftarrow \operatorname*{argmin}_x h^k(x)$$
$$\lambda^{k+1} \leftarrow \lambda^k + \rho^k\, \Phi(x^{k+1}).$$

Experiments: Data Setup.

- Nine real-world datasets

Name	#Node	#Edges	Average Degree	Diameter
Karate	34	78	2.294	5
Dolphins	62	159	5.129	8
Jazz	198	2,742	13.848	9
Network Science	1,589	2,742	3.451	17
Cora-ML	2,810	7,981	5.68	17
Power Grid	4,941	6,594	2.669	46
Memetracker	1,653	4,267	5.432	4
Digg	11,240	47,885	8.52	4
Deezer	47,538	222,887	9.377	-

Experiments: Data Setup.

- The strategy of diffusion generation except for Memetracker and Digg:

➢ 10% nodes were chosen as source nodes randomly.

➢ The diffusion was repeated 60 times for each source vector.

➢ The ratio of training and test was set to be 8:2.

➢ Memetracker and Digg provided source vectors and diffusion vectors.

- Comparison methods:

prescribed methods

➢ LPSI. It is inspired by the label propagation algorithm in semi-supervised learning.

➢ NetSleuth. Its goal is to employ the Minimum Description Length principle to identify the best set of source nodes.

GNN method

➢ GCNSI. It used the LPSI to augment input, and then applied the GCN for source identification.

Experiments: Metrics and Parameter Settings

- Metrics:

	Negative	Postive
Negative	TN	FP
Positive	FN	TP

➢ Accuracy(ACC)=(TN+TP)/(TN+FP+FN+TP)

➢ Precision(PR)=TP/(FP+TP)

➢ Recall(RE)=TP/(FN+TP)

➢ F-score(FS)=2RE*PR/(PR+RE)

➢ Area Under ROC curve(AUC): compute the Area under Receiver Operating Characteristic (ROC) curve.

- Parameter settings:

➢ f_w was set to a two-layers MLP model, g was set to the IC model.

➢ The validity constraint was set to $\|x\|_0 = |S|$, which means the number of source nodes was known in advance. But it is non-differentiable and thus was relaxed to $\sum_{i=1}^{n} x_i = |S|$.

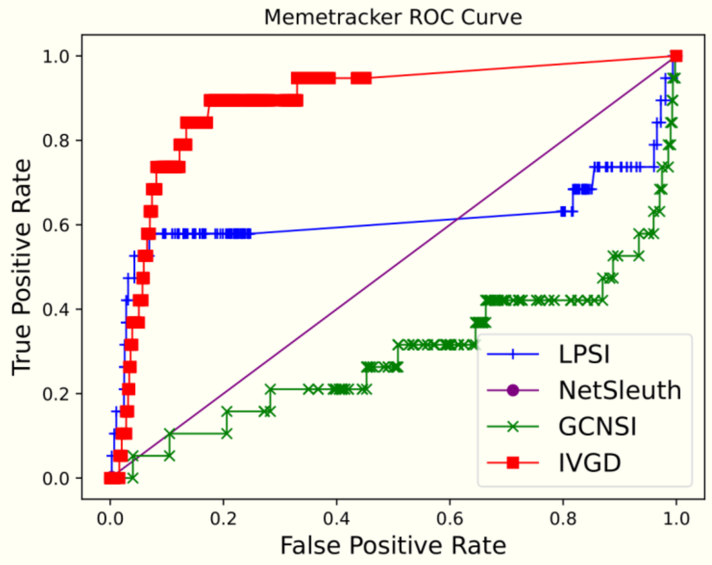
Experiments: Performance on Eight Datasets (1)

	Karate				Dolphins			
Method	ACC	PR	RE	FS	ACC	PR	RE	FS
LPSI	0.9559	0.68	1	0.797	0.9677	0.779	1	0.8717
NetSleuth	0.9147	0.5371	0.6833	0.5965	0.9306	0.6454	0.7425	0.6904
GCNSI	0.7088	0.115	0.2667	0.1581	0.6177	0.1015	0.2548	0.1372
IVGD(ours)	**0.9853**	**0.8717**	**1**	**0.9213**	**0.9935**	**0.9444**	**1**	**0.9701**
	Network Science				Cora-ML			
Method	ACC	PR	RE	FS	ACC	PR	RE	FS
LPSI	0.9831	0.8525	1	0.9202	0.9011	0.5067	0.9993	0.6724
NetSleuth	0.9595	0.7642	0.8429	0.8016	0.8229	0.1627	0.1793	0.1706
GCNSI	0.884	0.0582	0.0135	0.0218	0.858	0.097	0.0478	0.0637
IVGD(ours)	**0.9946**	**0.9476**	**1**	**0.973**	**0.9973**	**0.9744**	**1**	**0.987**

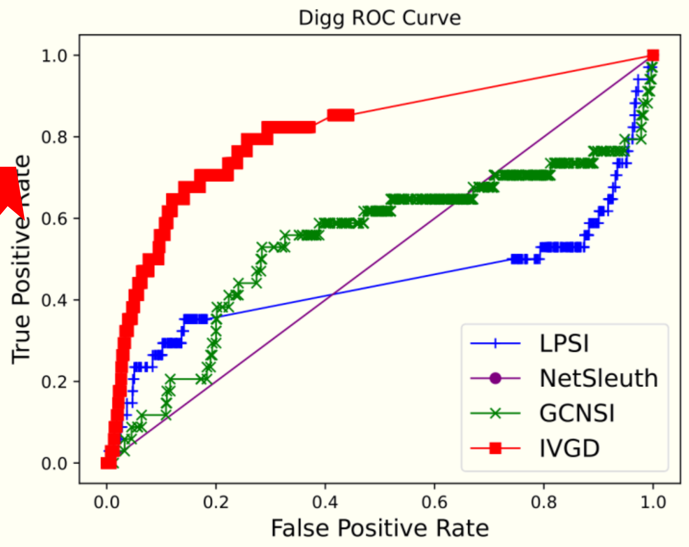
IVGD performs the best

Experiments: Performance on Eight Datasets (2)

	Jazz				Power Grid			
Method	ACC	PR	RE	FS	ACC	PR	RE	FS
LPSI	0.9035	0.6074	0.9944	0.7371	0.9673	0.7584	1	0.8624
NetSleuth	0.9222	0.5904	0.6629	0.6245	0.9276	0.6347	0.6986	0.6651
GCNSI	0.7525	0.0685	0.128	0.0849	0.7125	0.1022	0.2285	0.141
IVGD(ours)	0.998	0.9802	1	0.9899	0.9902	0.9133	1	0.9546



All comparison methods are surrounded by IVGD.



Experiments: Ablation Study

	Karate				Dolphins			
Method	ACC	PR	RE	FS	ACC	PR	RE	FS
IVGD(1)	0.9618	0.7167	1	0.8248	0.9694	0.7873	1	0.8774
IVGD(2)	**0.9882**	**0.8967**	**1**	**0.9356**	0.9726	0.807	1	0.8904
IVGD(3)	0.95	0.6583	1	0.7824	0.9613	0.7519	1	0.8517
IVGD	0.9853	0.8717	1	0.9213	**0.9935**	**0.9444**	**1**	**0.9701**

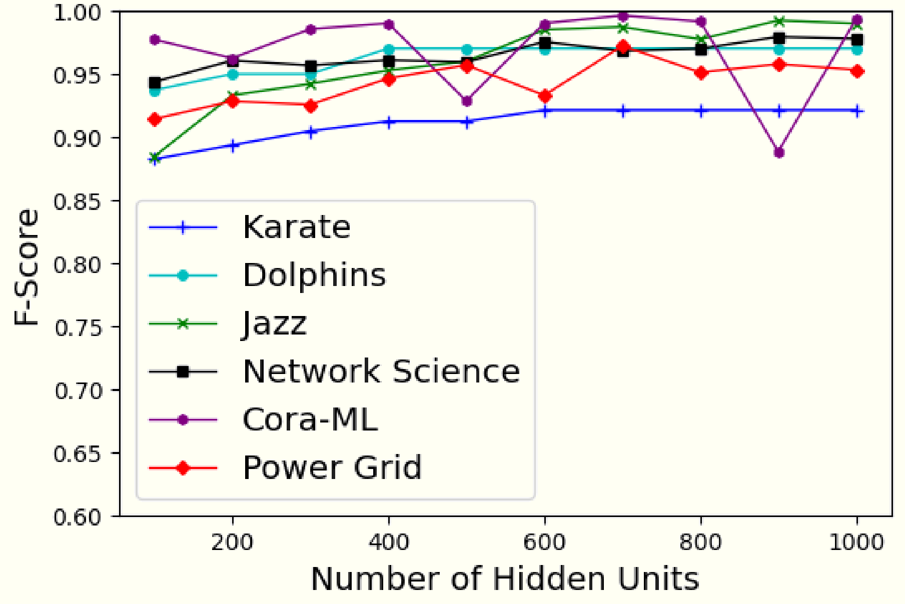
	Network Science				Cora-ML			
Method	ACC	PR	RE	FS	ACC	PR	RE	FS
IVGD(1)	0.9859	0.8736	1	0.9324	0.8712	0.4411	1	0.6121
IVGD(2)	0.9867	0.8799	1	0.936	0.9959	0.9617	1	0.9805
IVGD(3)	0.9812	0.8383	1	0.9119	0.985	0.8717	1	0.9314
IVGD	**0.9946**	**0.9476**	**1**	**0.973**	**0.9973**	**0.9744**	**1**	**0.987**

All modules are necessary to contribute to the outstanding performance of IVGD.

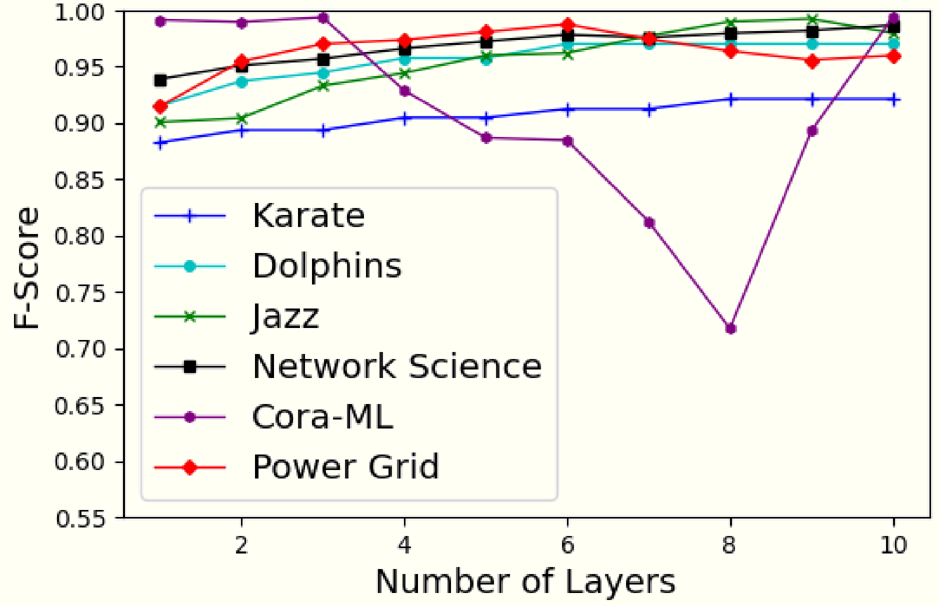
IVGD(1): the graph residual net was removed.
IVGD(2): the error compensation module was removed.
IVGD(3): the validity-aware layers were removed.

Experiments: Sensitivity Analysis

More hidden units and layers lead to better performance.



FS VS Hidden Units



FS VS Layers

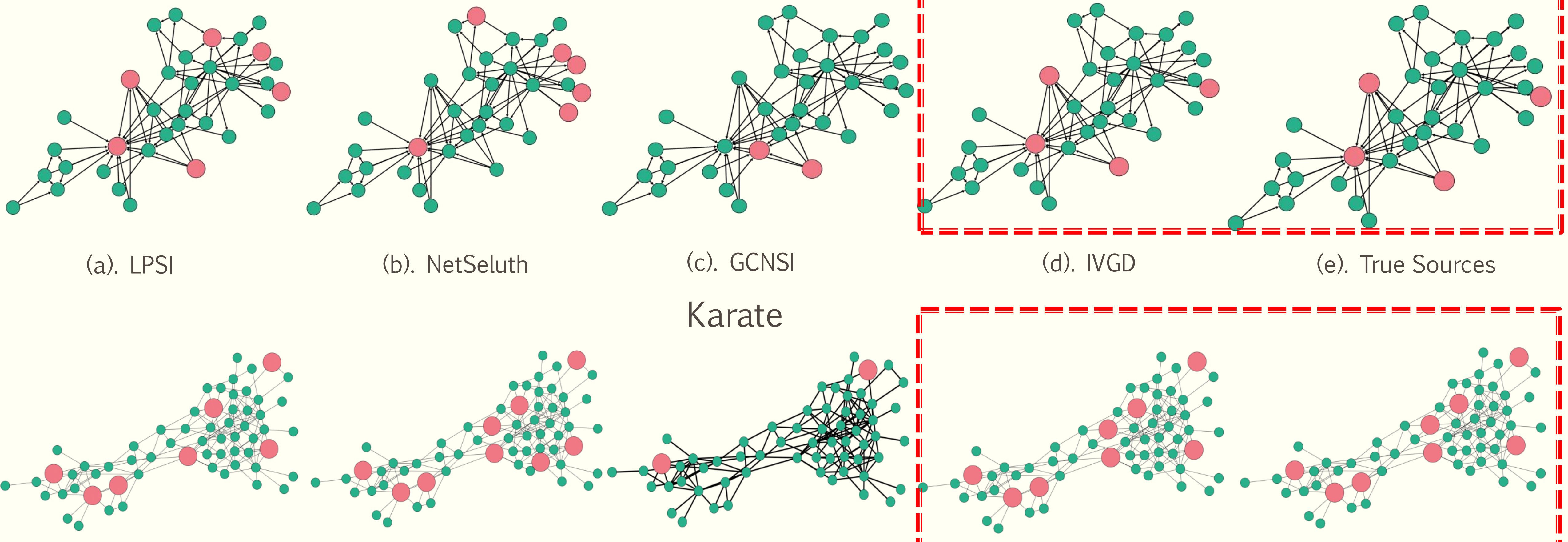
Experiments: Running Time

Method	Karate	Dolphins	Jazz	Network Science	Cora-ML	Power Grid	Deezer
LPSI	**0.26**	**0.27**	**0.76**	52.83	240.88	899.45	94541.13
NetSleuth	0.33	0.48	1.95	32.96	645.04	1260.68	114425.3
GCNSI	1.55	34.62	125.59	283.62	776.7	2324.53	174923.3
IVGD	5.37	6.45	9.78	**23.95**	**92.68**	**177.32**	**46832**

LPSI is the most efficient on small networks.

IVGD runs the most efficiently on large networks.

Experiments: Network Visualization



(a). LPSI (b). NetSeluth (c). GCNSI (d). IVGD (e). True Sources

Karate

(f). LPSI (g). NetSeluth (h). GCNSI (i). IVGD (j). True Sources

Dolphin

IVGD predicts all source nodes accurately.

Summary.

In this work, we

- Design a generic end-to-end framework for source location.
- Develop an invertible graph diffusion model via incorporating prior knowledge on graph mechanism.
- Propose efficient validity-aware layers to maintain the validity of inferred sources.
- Conduct experiments to demonstrate the effectiveness, efficiency and robustness of our proposed model.

Our Dataset

The link of our dataset and code:

https://github.com/xianggebenben/IVGD

dataset and code

Feel free to contact Junxiang Wang (jwan936@emory.edu) if you have any questions.

Thank you. Any questions?