GENERATIVE–DISCRIMINATIVE SOURCE LOCALIZATION FROM NOISY DIFFUSION OBSERVATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph source localization identifies the initial spreaders from node states observed after diffusion. It is an important task in epidemic tracing and information diffusion analysis. One way to assess a source estimate is to compare its predicted diffusion with the observation and refine the estimate by reducing their difference. Under noisy observations, however, we find that a closer fit can leave fewer true sources in the final predicted set. Better diffusion fit therefore does not necessarily improve localization, motivating refinement that directly targets source identification. We propose Generative–Discriminative Source Localization (GDSL). The method first produces an initial source estimate from the observation and network structure. It then predicts the resulting diffusion and computes node-level differences from the observation. A discriminative network combines these differences with the initial source scores and network structure, and learns to adjust node scores using source labels. Diffusion differences thus inform node decisions, while source supervision guides the correction. On SI and SIR simulations over four real-world networks, GDSL achieves the highest Top-K F1 among the evaluated implementations in all eight settings, with mean AUC, AP, and Top-K F1 of 0.963, 0.728, and 0.683. Stage analysis shows that most localization gains arise from discriminative correction, supporting source-label supervision alongside diffusion prediction.

1 INTRODUCTION

Diffusion leaves observations of its outcome, whereas source localization asks where it began. As diffusion spreads over a network, different source sets can produce similar node states, so an observation may not uniquely identify its origins. Missing and false reports further change these records and make source sets harder to distinguish. Source localization must therefore use the diffusion information retained in the observation to assess which source sets can explain it. This problem arises in epidemic tracing, rumor source identification, and network fault analysis (Ling et al., 2022).

Existing work approaches this problem through node identification and diffusion modeling. Some methods directly learn the relationship between observations and source nodes. For example, GCNSI uses graph convolutions to combine node and neighborhood information and learns multiple-source identification from source labels (Dong et al., 2019). Other methods connect a source estimate to the diffusion it could produce. SL-VAE combines a learned source distribution with forward diffusion prediction, while PGSL optimizes source estimates using diffusion observations and a source prior (Ling et al., 2022; Xu et al., 2024). These methods connect candidate sources to observations through a diffusion model, constraining inference with both propagation information and knowledge of source distributions.

Recent studies extend how this relationship is modeled. GDFSL learns forward propagation dynamics and a corresponding reverse process to handle differences between diffusion scenarios. PDSL combines generative source inference with continuous-time propagation modeling (Hou et al., 2025; Wang et al., 2026). SIDSL incorporates structural priors into conditional denoising when diffusion training data are limited (Chen et al., 2025). These advances show how propagation dynamics and network structure can inform source localization. We examine how this information can refine an

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

existing source estimate, and whether improved diffusion prediction translates into a better final source set. Appendix E discusses related work on inverse models, source-set search, and transfer.

This connection is not automatic. Diffusion-fit error measures the difference between predicted diffusion and an observation, while localization evaluates which initial nodes are recovered. Because different source sets may produce similar diffusion outcomes, closeness in the diffusion space alone cannot establish which set is more accurate. In our experiments with noisy observations, some refinements improve the fit but leave fewer true sources in the final output. A lower fitting error is therefore insufficient on its own to identify a useful correction. Diffusion information must also inform node decisions: how should the differences between prediction and observation raise true-source scores and lower false-source scores?

We propose GDSL to connect diffusion prediction with these node decisions. The method first generates an initial source estimate from the observation and network structure. A forward model predicts the resulting diffusion and retains each node's prediction and its difference from the observation. A discriminative network then combines this information with the initial source scores and network structure. It learns score corrections under direct source-label supervision. This design uses diffusion differences as inputs to source identification and lets localization supervision guide the correction. At inference, the model performs a finite number of score updates and retains the candidate with the lowest refinement objective. The evaluated setting outputs the K highest-scoring nodes, as specified in the experimental setup.

We evaluate GDSL on SI and SIR simulations over four real-world networks. The analysis combines overall localization results, changes before and after refinement, and outputs from successive inference stages. In addition to evaluating the complete method, we distinguish the roles of initial prediction, discriminative correction, and subsequent refinement. These experiments connect the method design to the observed mismatch between diffusion fit and localization.

Our main contributions are as follows:

1. We identify a mismatch between improvements in diffusion fit and source-set localization. By comparing fitting error and source-set predictions before and after the same refinement, we show that a lower error can coincide with fewer recovered true sources. We examine this behavior across networks and diffusion settings.
2. We propose a generative–discriminative framework for source localization. It uses the diffusion predictions and node-level differences associated with an initial estimate to guide discriminative correction. Direct source-label supervision teaches the network how to adjust node scores toward a more accurate source set.
3. We evaluate the framework through overall comparisons and stage analysis. GDSL obtains the highest Top-K F1 among the evaluated implementations in all eight SI and SIR settings over four real-world networks. Stage analysis shows that discriminative correction provides most of the localization gain, supporting the combined design.

2 METHOD

2.1 TASK AND OVERVIEW

Let $G = (V, E)$ be a graph with N nodes, adjacency matrix A, and edge transmission probability matrix P, where P_{ij} is the probability of transmission from node i to node j. The true source set is S, with binary labels s. We write y for the cumulative infection state before reporting noise and $\widetilde{y}$ for the observation with missed and false reports. For SIR diffusion, cumulative infection includes both infectious and recovered nodes; the recovery probability is γ. Training samples provide s, y, and $\widetilde{y}$. At inference, the model uses the graph, specified diffusion parameters, and noisy observations. The evaluated setting also supplies the source count K, but not the source locations.

GDSL consists of initial source estimation, propagation prediction, and discriminative correction (Figure 1). An observation encoder and generator first produce source scores x_p. A forward model predicts the corresponding spread. A discriminator then uses node-level differences between the prediction and observation to correct the scores. A limited number of subsequent updates gives the final scores x^*. The evaluated output contains the K highest-scoring nodes.

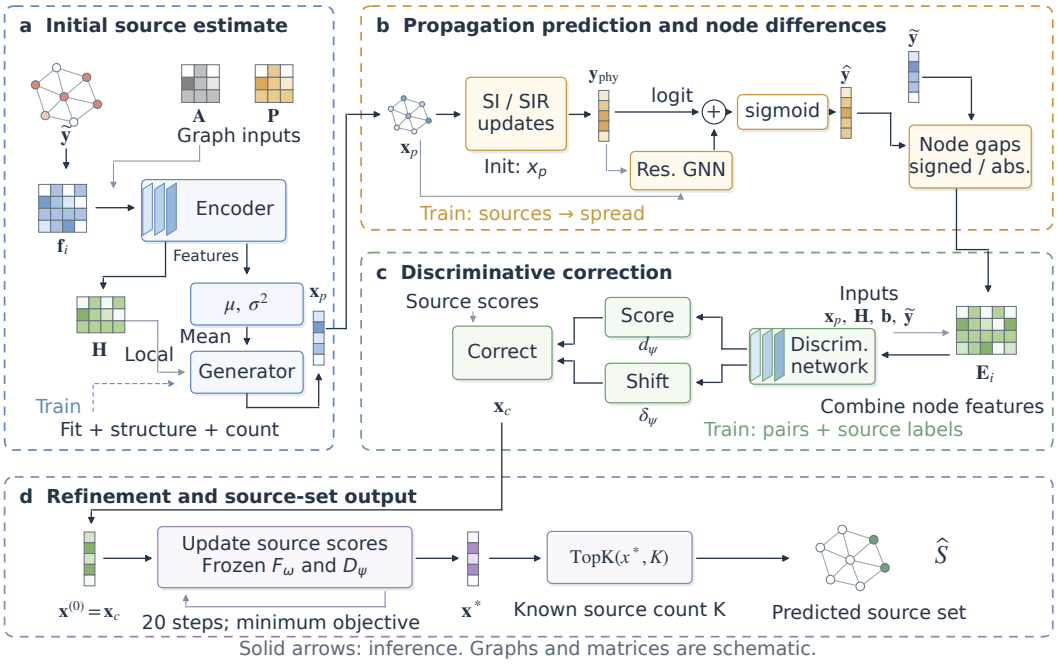

Figure 1: Overview of GDSL. The encoder and generator produce initial source scores. The forward model combines SI/SIR probability updates with a learned residual graph network. The discriminator uses differences between predicted and observed spread to correct source scores. Refinement updates these scores with network parameters held fixed. The four panels show observation encoding, propagation prediction, discriminative correction, and source-set output. Graphs and matrices are schematic.

2.2 Initial Source Estimation

Identifying a source depends on both its own observation and the states of nearby nodes. We symmetrically normalize $A+I$ and $P+I$ to obtain $\bar{A}$ and $\bar{P}$. Self-loops preserve a node's own information, while symmetric normalization scales contributions by the degrees of connected nodes. These matrices are used only for feature aggregation; propagation updates use the original probability matrix P. The input features of node i are

$$f_i = [\widetilde{y}_i, (\bar{A}\widetilde{y})_i, (\bar{P}\widetilde{y})_i, (\bar{P}^2\widetilde{y})_i, d_i, d_i^P] \tag{1}$$

Here, $\bar{A}\widetilde{y}$ aggregates a node's own observation and those of its immediate neighbors. The terms $\bar{P}\widetilde{y}$ and $\bar{P}^2\widetilde{y}$ apply one and two rounds of aggregation weighted by transmission probabilities. The degree d_i and weighted degree d_i^P are each divided by their respective maximum over the graph.

The observation encoder aggregates these features with $\bar{A}$ and $\bar{P}$. Graph convolutional layers with residual connections produce node representations H (Kipf & Welling, 2017). The encoder flattens H, or applies learned compression along the node dimension, and uses fully connected layers to obtain the mean μ and variance σ^2 of a global latent representation:

$$q_\phi(z \mid \widetilde{y}, A, P) = \mathcal{N}(\mu, \mathrm{diag}(\sigma^2)) \tag{2}$$

The generator combines this global representation with node context C. The context contains H, each node's own observation, the three aggregated observations, and the absolute difference between its own observation and the propagation-weighted neighborhood observation. Decoding the latent mean μ gives one vector of initial source scores:

$$x_p = G_\theta(\mu, C) \tag{3}$$

To incorporate local propagation structure into the initial estimate, we construct a structural cue b from the observation:

$$b = \mathrm{clip}_{[\varepsilon,1-\varepsilon]}\left\{0.78\widetilde{y} \odot \mathrm{sigmoid}[4(\widetilde{y} - 0.65\bar{P}\widetilde{y})] + 0.22\bar{P}^2\widetilde{y}\right\}. \tag{4}$$

The operator $\odot$ denotes elementwise multiplication, sigmoid maps values to $(0, 1)$, and clip restricts them to $[\varepsilon, 1-\varepsilon]$, with $\varepsilon = 10^{-5}$. This cue combines a node's own record, its difference from neighboring records, and two rounds of neighborhood aggregation. It does not use source labels.

The initial source generator is trained with

$$\begin{aligned}\mathcal{L}_p = \lambda_b \,\mathrm{BCE}(x_p, b) + \lambda_f \,\mathrm{BCE}(F_\omega(x_p), y) \\ + \lambda_c \left(\frac{\sum_i x_{p,i} - \kappa}{\kappa} \right)^2 + \lambda_{\mathrm{KL}} \ell_{\mathrm{KL}}.\end{aligned} \tag{5}$$

The first two terms encourage agreement with the structural cue and with the clean training propagation state y, respectively. The count term constrains the sum of source scores towards $\kappa = \max(1, r_s N)$, where r_s is the source ratio specified for the data setting. The term ℓ_{KL} regularizes the latent representation towards a standard Gaussian (Kingma & Welling, 2014). The coefficients control the four terms; detailed definitions are given in Appendix B. During generator training, the forward model is frozen, but gradients from the propagation loss pass through it to the generator.

2.3 Predicting propagation from source estimates

The forward model treats source scores as approximate initial infection probabilities. Let $a^{(t)}$ denote the probability of being infectious at step t and $e^{(t)}$ the probability of having been infected by that step, with $a^{(0)} = e^{(0)} = x$. Treating transmission events from neighbors as independent gives

$$\begin{aligned} h_j^{(t)} &= \prod_i (1 - a_i^{(t)} P_{ij}), \\ q_j^{(t)} &= (1 - e_j^{(t)})(1 - h_j^{(t)}), \qquad e_j^{(t+1)} = e_j^{(t)} + q_j^{(t)}. \end{aligned} \tag{6}$$

Here, $h_j^{(t)}$ is the probability of receiving no infection from neighbors, and $q_j^{(t)}$ is the probability of a new infection. In SI, infected nodes remain infectious. In SIR, existing infections recover with probability γ (Pastor-Satorras et al., 2015), giving

$$a^{(t+1)} = \begin{cases} e^{(t+1)}, & \text{SI}, \\ (1-\gamma) a^{(t)} + q^{(t)}, & \text{SIR}. \end{cases} \tag{7}$$

Updates stop when cumulative infection reaches the target scale ρN or the step limit T. The resulting prediction is y_{phy}. A residual graph network learns a bounded correction to this approximation:

$$\begin{aligned} r_\omega &= 3 \tanh\big(\mathrm{GNN}_\omega[x, y_{\mathrm{phy}}, y_{\mathrm{phy}} - x, \bar{P}x, d, d^P]\big), \\ F_\omega(x) &= \mathrm{sigmoid}(\mathrm{logit}(y_{\mathrm{phy}}) + r_\omega). \end{aligned} \tag{8}$$

This retains propagation computation based on P while allowing a trainable graph network to correct the prediction in logit space. Probabilities are clipped before the logit operation for numerical stability. The model learns from true source–propagation pairs using binary cross-entropy, mean squared error, and soft Dice loss (Appendix A). For an estimate x_p, the discriminator receives $\widehat{y} = F_\omega(x_p)$, $\widetilde{y} - \widehat{y}$, and $|\widetilde{y} - \widehat{y}|$, retaining the prediction, discrepancy direction, and discrepancy magnitude.

2.4 Correcting source scores with propagation discrepancies

Discriminative correction learns how to adjust source scores using propagation discrepancies. For candidate scores x, we recompute $\widehat{y} = F_\omega(x)$ and construct the input for node i:

$$\begin{aligned} E_i = [x_i, b_i, \widetilde{y}_i, (\bar{P}x)_i, (\bar{P}\widetilde{y})_i, \widehat{y}_i, \\ |\widetilde{y}_i - \widehat{y}_i|, \widetilde{y}_i - \widehat{y}_i, |x_i - b_i|, d_i, d_i^P]. \end{aligned} \tag{9}$$

These features describe the candidate sources, the node and neighborhood observations, and differences between predicted and observed spread. For the initial correction, a linear projection of the encoder representations H also enters the discriminator to retain observation context.

The discriminator has two outputs. The node output δ_ψ specifies the direction and size of each score adjustment. The graph output d_ψ, obtained through mean and max pooling of node representations,

scores the compatibility of the candidate with the observation. This score also controls the overall adjustment strength. The corrected source scores are

$$x_c = \mathrm{sigmoid}[\mathrm{logit}(\bar{x}_p) + \alpha\{0.5 + \mathrm{sigmoid}(-d_\psi)\}\delta_\psi] \tag{10}$$

Here, $\bar{x}_p$ denotes numerically clipped initial scores and α controls correction strength. The node correction is bounded by $4\tanh$. A lower graph compatibility score increases the gate $0.5 + \mathrm{sigmoid}(-d_\psi)$, allowing a larger adjustment. The two outputs jointly determine x_c.

During training, true source–observation pairs receive higher graph scores than five types of negative pairs, including perturbed source sets and mismatched observations. The graph-level hinge ranking loss $\mathcal{L}_{\mathrm{pair}}$ is defined in Appendix B. Source labels also supervise corrected node scores:

$$\mathcal{L}_D = \lambda_r \mathcal{L}_{\mathrm{pair}} + \lambda_s[\ell_{\mathrm{src}}(x_c, s) + 0.35\ell_{\mathrm{src}}(x_{\mathrm{hard},c}, s)] \tag{11}$$

The vector $x_{\mathrm{hard},c}$ is the result of correcting a node-replacement candidate with the same discriminator. The loss ℓ_{src} combines node classification, source-set overlap, ranking of true sources against high-scoring non-sources, and a constraint on total score mass (Appendix B). Graph-level ranking learns source–observation compatibility, while node supervision learns which nodes should be identified as sources after correction.

2.5 Staged training and inference

Training proceeds through the forward model, initial source generator, and discriminator. The forward model first learns from source–propagation pairs. It is then frozen while the observation encoder and generator are trained. Finally, both upstream components are frozen and the discriminator is trained. All stages use AdamW (Loshchilov & Hutter, 2019), and model selection uses the validation set.

At inference, the generator and discriminator first produce x_c. We then refine the source scores for the current sample over a limited number of steps. We convert x_c to a logit variable v and compute candidate scores as $x = \mathrm{sigmoid}(v)$. Only v is optimized; all network parameters remain fixed. The objective is

$$\begin{aligned}\mathcal{L}_{\mathrm{ref}}(x) = {} & \mathrm{BCE}(F_\omega(x), \widetilde{y}) + \eta_b \,\mathrm{BCE}(x, b) \\ & + \eta_c \left(\frac{\sum_i x_i - R}{N}\right)^2 + \eta_d \,\mathrm{softplus}[-d_\psi(x)].\end{aligned} \tag{12}$$

The count reference $R = \max(1, \mathrm{round}(r_s N))$ is specified by the experimental setting and remains fixed during refinement. The four terms measure propagation fit to the noisy observation, agreement with the structural cue, deviation from the reference count, and source–observation compatibility. Each step recomputes the forward prediction and graph score before updating v with Adam (Kingma & Ba, 2015).

Correction strength is selected on validation data. The evaluated implementation performs 20 updates and returns, for each sample, the candidate with the lowest refinement objective among steps 0–19, including the unrefined discriminator output at step 0. Candidate selection does not use test source locations or test localization scores. Implementation details are given in Appendix C. The final source set is

$$\widehat{S} = \mathrm{TopK}(x^*, K). \tag{13}$$

3 Experiments

We organize the experiments around the relationship between diffusion prediction and source identification. We first test whether a reduction in diffusion-fit error also improves localization. We then compare the complete GDSL framework with baseline implementations and use ablations and stage analysis to examine structural guidance and discriminative correction. Experiments cover SI and SIR diffusion over four real-world networks. Per-sample changes assess whether a refinement is useful, while aggregate localization metrics evaluate the complete method.

3.1 DATA AND EVALUATION

We use Jazz, Network Science, Cora-ML, and Power Grid, which represent musician collaborations, scientific collaborations, paper citations, and power-grid connections, respectively (Gleiser & Danon, 2003; Newman, 2006; McCallum et al., 2000; Watts & Strogatz, 1998). SI and SIR simulations on these networks give eight experimental settings. Table 1 summarizes the network sizes.

Table 1: Structural statistics of the experimental networks. Edges are counted as undirected.

Network	Nodes	Edges	Mean degree
Jazz	198	2742	27.697
Network Science	1589	2742	3.451
Cora-ML	2810	7981	5.680
Power Grid	4941	6594	2.669

Each setting contains 3,000 samples. Sources make up approximately 10% of the nodes and form locally clustered sets. Diffusion targets a cumulative infected fraction of approximately 32%. For each source set, we simulate two independent realizations, add reporting noise to each, and average their node records to form the input observation. The base visibility of infected nodes is 0.88, and the mixed reporting-noise parameter is 0.06. Samples are grouped by source set and split into training, validation, and test sets in proportions of 70%, 15%, and 15%, so the same source set does not appear in different splits. Appendix C describes propagation parameters, noise generation, and training settings.

We report AUC, AP, and Top-K F1. AUC measures how well sources rank above non-sources (Fawcett, 2006). AP summarizes precision across recall levels. Because sources are a minority of the nodes, precision–recall evaluation complements ROC analysis (Davis & Goadrich, 2006; Saito & Rehmsmeier, 2015). Top-K F1 measures overlap between the selected nodes and the true source set. For Top-K F1, every method selects the K highest-scoring nodes, where K is the true source count of the sample. We first average each metric over a test set and then weight the eight settings equally when reporting an overall mean.

3.2 PROPAGATION FIT AND SOURCE LOCALIZATION

We first test whether adjusting source scores improves both diffusion prediction and localization. Using the trained models for Cora-ML/SI and Power Grid/SI, we start from the same post-correction scores for each test sample and continue updating them solely to reduce the binary cross-entropy between predicted diffusion and the observation. Network parameters remain fixed, and the source-set output rule is unchanged. Each setting includes all 450 test samples. Figure 2 compares the changes in fitting error and Top-K F1.

Fitting error decreases for every sample in both settings, but localization does not improve consistently. F1 decreases for 178 samples in Cora-ML/SI (39.56%) and 204 samples in Power Grid/SI (45.33%). The figure also shows samples whose localization improves or remains unchanged. A reduction in fitting error alone therefore does not determine whether a score update recovers more true sources.

We observe the same behavior in a separate experiment with a single diffusion observation, different reporting noise, and a common propagation horizon. Appendix D reports per-sample statistics for all eight network–diffusion settings. The fraction of samples with better fit but worse localization ranges from 29.11% to 58.44%. Mean F1 still increases in Jazz/SI and Network Science/SIR, showing that an average gain can coexist with losses for individual samples. These results motivate evaluating the final source set separately from diffusion fit and examining how diffusion differences can inform source-node decisions.

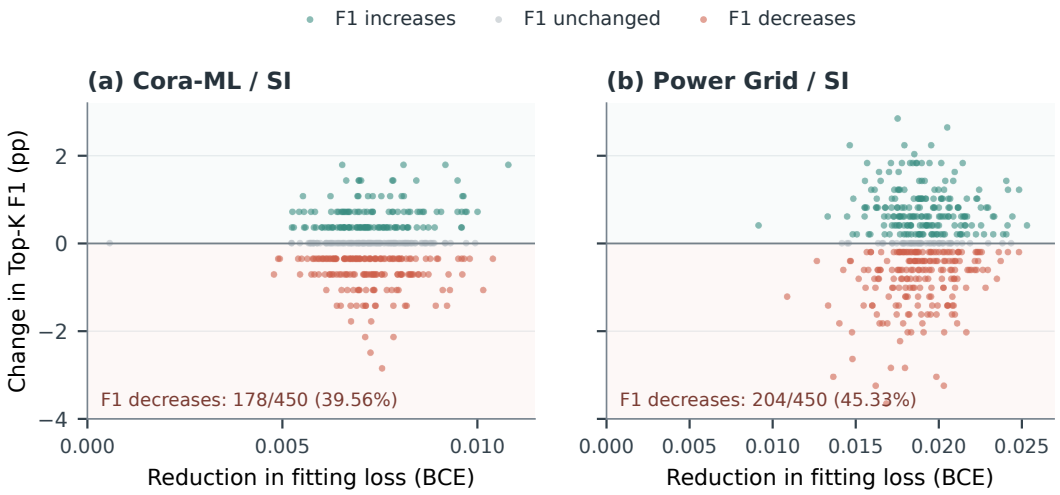

Figure 2: Per-sample localization changes when diffusion-fit error decreases. The horizontal axis is the fitting error before refinement minus the error after refinement; positive values indicate a closer fit. The vertical axis is the change in Top-K F1; negative values indicate fewer recovered true sources. Each panel contains 450 test samples.

3.3 Comparison methods

We compare the following source-localization approaches. SL-VAE learns a source distribution with a variational autoencoder and uses diffusion prediction to recover source states (Ling et al., 2022). GCNSI combines node and neighborhood observations through graph convolutions to predict source labels (Dong et al., 2019). NetSleuth identifies sources from the spectral structure of the infected subgraph (Prakash et al., 2012). We also include the inverse prediction branch of our local PGSL implementation (Xu et al., 2024), denoted PGSL†; Appendix C defines its implementation scope. All evaluated implementations use the same data splits and metrics.

3.4 Overall localization results

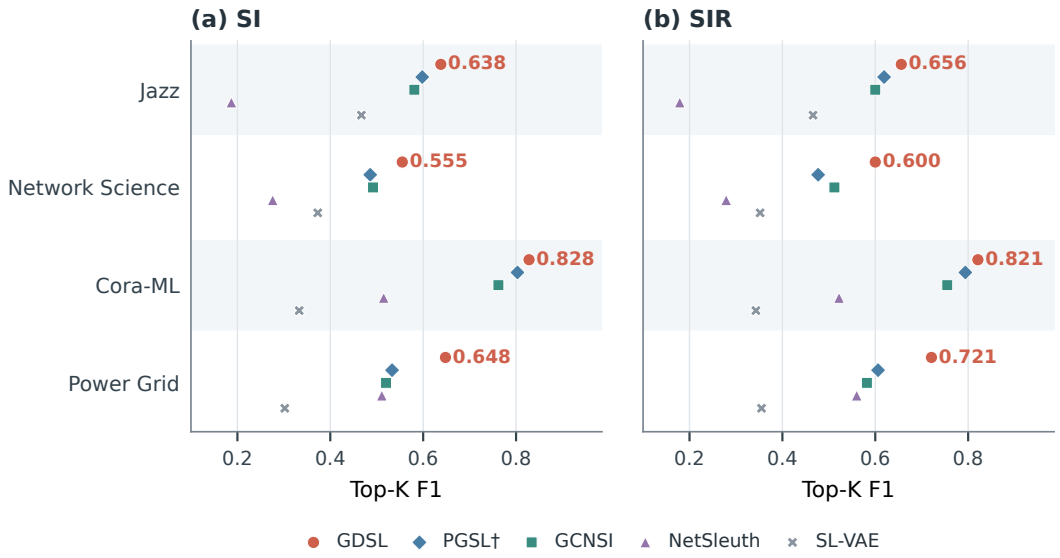

Figure 3: Top-K F1 across eight settings. Points show test-set means, with GDSL scores labeled. PGSL† is the inverse prediction branch defined in Appendix C.

Figure 3 compares Top-K F1 across the eight settings. GDSL obtains the highest score in every setting, with a mean F1 of 0.683. This exceeds PGSL† by 6.89 percentage points. On Cora-ML, GDSL scores 0.828 under SI and 0.821 under SIR, improving on PGSL† by 2.5 and 2.7 points. The gains are 11.5 points in both Power Grid settings. These results show that the complete combination of initial source estimation, diffusion prediction, and discriminative correction improves source selection under the evaluated conditions.

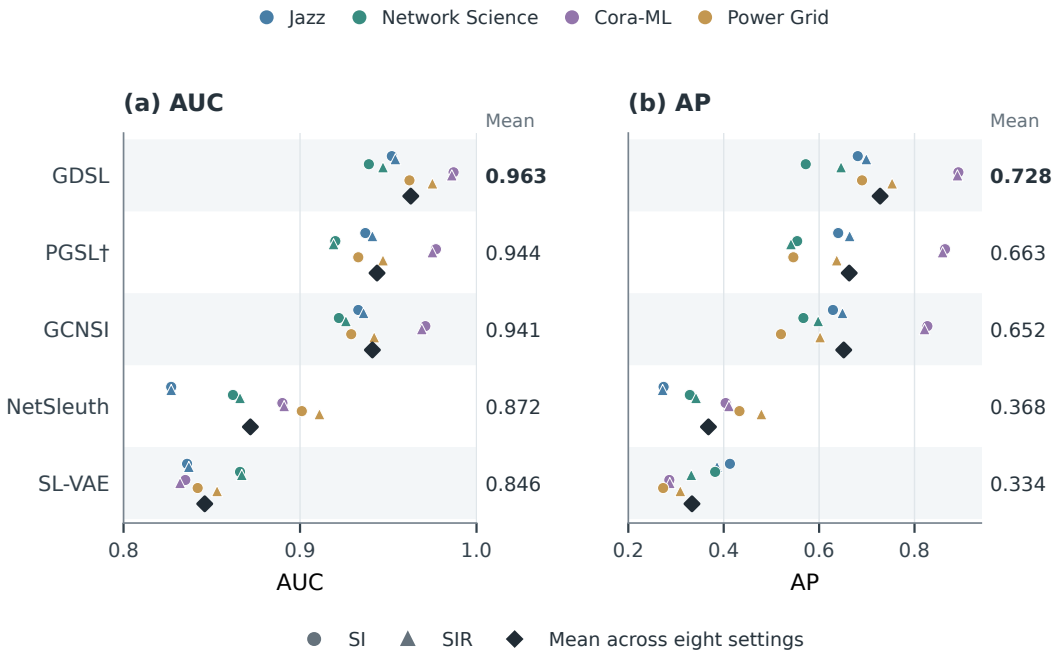

Figure 4: AUC and AP by network and diffusion setting. Colors identify networks; circles and triangles denote SI and SIR. Black diamonds mark equally weighted means. Vertical offsets only separate points. Appendix F lists all values.

Figure 4 complements this comparison with ranking metrics. GDSL achieves a mean AUC of 0.963 and a mean AP of 0.728, exceeding PGSL† by 1.91 and 6.48 percentage points, respectively. Its AUC, AP, and F1 exceed those of the evaluated baselines in all eight settings. The consistent gains cover overall node ranking, precision across recall levels, and final source-set recovery.

3.5 Structural Cues and Discriminative Correction

We study two ablations of the complete pipeline. Removing structural guidance disables the structural hint in generation, discrimination, and subsequent refinement, and retrains the corresponding generator and discriminator. The network structure used by graph convolutions is unchanged. Removing the discriminative correction stage retains the initial generator and forward model, bypasses node correction, and removes the graph-level compatibility term from the refinement objective. Table 2 reports the average metrics, and Table 3 gives F1 for each setting.

Table 2: Average ablation results, with equal weight given to each of the eight settings.

Method	AUC	AP	Top-K F1
GDSL	0.9627	0.7281	0.6834
Without structural cue	0.9547	0.6811	0.6441
Without discriminative stage	0.9104	0.5085	0.5010

Table 3: Ablation results for each network and diffusion setting. Δ is the F1 gain of the complete method over the corresponding ablation, in percentage points.

Setting	GDSL	No cue	Δ_{cue}	No disc.	Δ_{disc}
Jazz / SI	0.638	0.634	0.40	0.544	9.40
Jazz / SIR	0.656	0.655	0.10	0.567	8.90
Network Science / SI	0.555	0.507	4.80	0.361	19.40
Network Science / SIR	0.600	0.535	6.50	0.394	20.60
Cora-ML / SI	0.828	0.804	2.40	0.569	25.90
Cora-ML / SIR	0.821	0.799	2.20	0.564	25.70
Power Grid / SI	0.648	0.593	5.50	0.458	19.00
Power Grid / SIR	0.721	0.626	9.50	0.551	17.00

Removing structural guidance reduces mean F1 by 3.93 percentage points, with a decrease in every setting. Changes are small on Jazz but larger on Network Science and Power Grid. The contribution of the structural hint therefore varies across network and diffusion settings.

Removing the discriminative correction stage reduces mean F1 by 18.24 percentage points. The decreases range from 8.9 to 25.9 points, and AUC and AP also fall. This ablation evaluates the entire stage, including its source-label supervision. It shows that learning source-node decisions after the initial estimate is an important part of the complete method's localization performance.

Table 4: Top-K F1 at successive output stages. Δ_{corr} and Δ_{ref} denote the changes from the preceding stage, in percentage points.

Setting	Initial	Corrected	Refined	Δ_{corr}	Δ_{ref}
Jazz / SIR	0.5806	0.6558	0.6562	+7.517	+0.045
Network Science / SI	0.3633	0.5528	0.5547	+18.946	+0.196
Network Science / SIR	0.3928	0.6070	0.5999	+21.420	-0.709
Cora-ML / SI	0.5936	0.8290	0.8282	+23.541	-0.079
Cora-ML / SIR	0.5935	0.8223	0.8213	+22.874	-0.102
Power Grid / SI	0.4910	0.6513	0.6478	+16.030	-0.344
Power Grid / SIR	0.5666	0.7253	0.7208	+15.866	-0.448

Table 4 compares successive outputs of the same trained model, evaluated on the same test samples with the same Top-K F1 rule. These are not independently retrained ablations. Discriminative correction improves F1 in all seven settings with complete stage records. Subsequent refinement improves F1 in two settings and slightly reduces it in five. This agrees with Section 3.2: discriminative correction and subsequent fitting-based refinement should be evaluated separately, and lower fitting error should not be equated with better localization.

4 Discussion and conclusion

We presented GDSL, which combines initial source generation, diffusion prediction, and discriminative correction to recover diffusion sources from noisy observations. The method uses node-level differences between predicted diffusion and observations to inform source identification, and learns corrections from source labels. SI and SIR experiments over four real-world networks show improved localization in the evaluated comparisons. Ablations support the contributions of structural guidance and the complete discriminative correction stage.

Per-sample analysis shows that a closer fit to the observation can still produce a worse source set. Together with the overall results and stage analysis, this supports using diffusion differences to inform node decisions and evaluating the value of a correction through localization outcomes. Future work will examine inaccurate propagation parameters and incomplete network structure.

AI USE STATEMENT

AI tools were used for coding assistance, support in examining experimental results and related work, manuscript editing and translation, and figure preparation. The authors are responsible for verifying the methods, results, citations, and final manuscript.

REFERENCES

Hongyi Chen, Jingtao Ding, Xiaojun Liang, Yong Li, and Xiao-Ping Zhang. Structure-prior informed diffusion model for graph source localization with limited data. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, 2025. doi: 10.1145/3746252.3761146. URL https://arxiv.org/abs/2502.17928.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning*, pp. 233–240. ACM, 2006. doi: 10.1145/1143844.1143874. URL https://ftp.cs.wisc.edu/machine-learning/shavlik-group/davis.icml06.pdf.

Ming Dong, Bolong Zheng, Quoc Viet Hung Nguyen, Han Su, and Guohui Li. Multiple rumor source detection with graph convolutional networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 569–578. ACM, 2019. doi: 10.1145/3357384.3357994. URL https://doi.org/10.1145/3357384.3357994.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. doi: 10.1016/j.patrec.2005.10.010. URL https://www.sciencedirect.com/science/article/pii/S016786550500303X.

Pablo M. Gleiser and Leon Danon. Community structure in jazz. *Advances in Complex Systems*, 6(4):565–573, 2003. doi: 10.1142/S0219525903001067. URL https://arxiv.org/abs/cond-mat/0307434.

Dongpeng Hou, Yuchen Wang, Chao Gao, and Xianghua Li. A generalized diffusion framework with learnable propagation dynamics for source localization. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 2919–2927, 2025. doi: 10.24963/ijcai.2025/325. URL https://www.ijcai.org/proceedings/2025/325.

Bosong Huang, Weihao Yu, Ruzhong Xie, Jing Xiao, and Jin Huang. Two-stage denoising diffusion model for source localization in graph inverse problems. *arXiv preprint arXiv:2304.08841*, 2023. URL https://arxiv.org/abs/2304.08841.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6114.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1609.02907.

Chen Ling, Junji Jiang, Junxiang Wang, and Liang Zhao. Source localization of graph diffusion via variational autoencoders for graph inverse problems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1010–1020. ACM, 2022. doi: 10.1145/3534678.3539288. URL https://arxiv.org/abs/2206.12327.

Leyuan Liu, Zhangtao Cheng, Xovee Xu, and Fan Zhou. Denoising propagation uncertainty for information source localization. In *GLOBECOM 2024 – 2024 IEEE Global Communications Conference*, pp. 4244–4249, 2024. doi: 10.1109/GLOBECOM52923.2024.10901676. URL https://www.xoveexu.com/file/paper/24-12-GLOBECOM-DDSL.pdf.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000. doi: 10.1023/A:1009953814988. URL https://link.springer.com/article/10.1023/A:1009953814988.

M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104, 2006. doi: 10.1103/PhysRevE.74.036104. URL https://arxiv.org/abs/physics/0605087.

Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. Epidemic processes in complex networks. *Reviews of Modern Physics*, 87(3):925–979, 2015. doi: 10.1103/RevModPhys.87.925. URL https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.87.925.

B. Aditya Prakash, Jilles Vreeken, and Christos Faloutsos. Spotting culprits in epidemics: How many and which ones? In *2012 IEEE 12th International Conference on Data Mining*, pp. 11–20. IEEE, 2012. doi: 10.1109/ICDM.2012.136. URL https://faculty.cc.gatech.edu/~badityap/papers/netsleuth-icdm12.pdf.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):e0118432, 2015. doi: 10.1371/journal.pone.0118432. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0118432.

Li Sun, Suyang Zhou, Bowen Fang, Hechuan Zhang, Junda Ye, Yutong Ye, and Philip S. Yu. Trace: Structural riemannian bridge matching for transferable source localization in information propagation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 3308–3316, 2025. doi: 10.24963/ijcai.2025/368. URL https://www.ijcai.org/proceedings/2025/368.

Junxiang Wang, Junji Jiang, and Liang Zhao. An invertible graph diffusion neural network for source localization. In *Proceedings of the ACM Web Conference 2022*, pp. 1058–1069. ACM, 2022. doi: 10.1145/3485447.3512155. URL https://arxiv.org/abs/2206.09214.

Yansong Wang, Qisen Chai, Longlong Lin, and Tao Jia. PDSL: Propagation dynamics aware framework for source localization. *IEEE Transactions on Network Science and Engineering*, 2026. doi: 10.1109/TNSE.2026.3688551. URL https://arxiv.org/abs/2605.03550.

Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998. doi: 10.1038/30918. URL https://www.nature.com/articles/30918.

Xovee Xu, Tangjiang Qian, Zhe Xiao, Ni Zhang, Jin Wu, and Fan Zhou. PGSL: A probabilistic graph diffusion model for source localization. *Expert Systems with Applications*, 238:122028, 2024. doi: 10.1016/j.eswa.2023.122028. URL https://www.xoveexu.com/file/paper/23-10-ESWA-PGSL.pdf.

Xin Yan, Hui Fang, and Qiang He. Diffusion model for graph inverse problems: Towards effective source localization on complex networks. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/46ab9d9645b6975b947231ddb48da1ab-Abstract-Conference.html.

Zonghan Zhang, Zijian Zhang, and Zhiqian Chen. Multiple-source localization from a single-snapshot observation using graph bayesian optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 22538–22546, 2024. doi: 10.1609/aaai.v38i20.30262. URL https://ojs.aaai.org/index.php/AAAI/article/view/30262.

A FORWARD-MODEL TRAINING

The forward model learns from training pairs of true sources s and propagation states y, using

$$\begin{aligned}\mathcal{L}_F = \; & \mathrm{BCE}(F_\omega(s), y) + 0.5\,\mathrm{MSE}(F_\omega(s), y) \\ & + 0.25\ell_{\mathrm{Dice}}(F_\omega(s), y).\end{aligned} \tag{14}$$

BCE and MSE measure cross-entropy and mean squared error between node probabilities, respectively. The soft Dice loss measures overlap with the propagation state. Once the forward model has been trained, it maps x_p to $\widehat{y} = F_\omega(x_p)$. The discriminator receives $\widehat{y}$, $\widetilde{y} - \widehat{y}$, and $|\widetilde{y} - \widehat{y}|$, retaining the predicted spread, the direction of the discrepancy, and its magnitude.

B TRAINING LOSSES

B.1 SOURCE SUPERVISION AND LATENT REGULARIZATION

The source supervision loss for discriminative correction contains five terms:

$$\ell_{\mathrm{src}}(x, s) = w_1\ell_{\mathrm{WBCE}} + w_2\ell_{\mathrm{Dice}} + w_3\ell_{\mathrm{hard}} + w_4\ell_{\mathrm{list}} + w_5\ell_{\mathrm{count}} \tag{15}$$

The weighted node cross-entropy ℓ_{WBCE} gives a larger weight to positive labels. The soft Dice loss measures source-set overlap:

$$\ell_{\mathrm{Dice}}(x, s) = 1 - \frac{2\sum_i x_i s_i + \varepsilon_D}{\sum_i x_i + \sum_i s_i + \varepsilon_D} \tag{16}$$

The constant $\varepsilon_D = 10^{-6}$ provides numerical stability. The hard-negative loss ℓ_{hard} pairs true sources with the highest-scoring non-source nodes and applies softplus to "non-source score minus source score plus margin." The listwise loss ℓ_{list} applies softmax to temperature-scaled logits and computes cross-entropy against a uniform target distribution over true sources. The count loss ℓ_{count} applies smooth L1 to the ratio of total score mass to the true training source count, with target ratio 1. All terms are averaged over training samples, and the weights w control their contributions.

The regularizer for the global latent representation is averaged over its d_z dimensions:

$$\ell_{\mathrm{KL}} = -\frac{1}{2d_z}\sum_{j=1}^{d_z}(1 + \log\sigma_j^2 - \mu_j^2 - \sigma_j^2) \tag{17}$$

The forward model uses the same soft Dice expression, replacing the source target s with the propagation state y. The generator count constraint uses the configured training reference κ; the source supervision count loss uses the true training count. At inference, refinement uses the count reference R specified in Section 2.5.

B.2 SOURCE–OBSERVATION PAIR RANKING

During training, the true sources and their observation form a positive pair. We construct five types of negative pairs: soft candidates derived from generator scores; candidates that replace some true sources with non-source nodes; sources paired with another sample's observation; replacements constructed from the structural cue; and candidates with an incorrect source count. A graph-level ranking loss requires the positive pair to score above the negative pairs:

$$\mathcal{L}_{\mathrm{pair}} = \frac{1}{|\mathcal{N}|}\sum_{j\in\mathcal{N}}\max(0, m - d^+ + d_j^-) \tag{18}$$

Here, d^+ is the positive-pair score, d_j^- is the score of negative pair j, $\mathcal{N}$ is the set of negative pairs, and m is a margin. The loss is also averaged over training samples. Source labels also directly supervise the corrected node scores.

Equation 11 gives the combined objective.

C EXPERIMENTAL IMPLEMENTATION

C.1 DATA GENERATION AND EVALUATION

We sample locally clustered source sets containing approximately 10% of the nodes. Diffusion is resampled a limited number of times to approach the target infection scale. Node records represent cumulative infection, so recovered SIR nodes count as previously infected. Edge transmission probabilities combine a base transmission intensity, node degree, and edge heterogeneity. Table 5 gives the base intensity β and maximum number of propagation steps T. The SIR recovery probability is 0.05.

Table 5: Diffusion parameters.

Network	SI β	SI T	SIR β	SIR T
Jazz	0.018	30	0.020	35
Network Science	0.120	40	0.220	45
Cora-ML	0.075	40	0.090	50
Power Grid	0.160	50	0.200	60

Reporting noise first hides each previously infected node with probability 0.12. It then flips records using a mixture of random and locally correlated changes with a budget of $0.06N$. Noise is added separately to the two diffusion realizations before averaging. Training, validation, and test splits are grouped by source set. Learned models are fitted on the training set, and model selection uses the validation set. The test set is used only for final evaluation.

Figures 3 and 4 and Tables 8–9 use the main results rounded to three decimal places. Table 2 uses the four-decimal ablation summary, while Table 3 retains three-decimal per-setting F1 values. Means computed at different precisions can therefore differ in their last digit. Variation across settings is not presented as uncertainty across repeated training runs.

C.2 INFERENCE AND ABLATIONS

For the reported results, validation selects the discriminative correction strength. Inference then performs at most 20 source-score updates and selects the candidate with the lowest full refinement objective among steps 0 through 19. The fitting-only experiment in Section 3.2 evaluates the same candidate steps but uses only the diffusion-fit objective for both updates and selection. Neither procedure uses test source labels to select intermediate candidates.

The structural-guidance ablation starts from the same forward model and retrains the generator and discriminator. The correction-stage ablation reuses the initial generator and bypasses both the discriminator and its compatibility term in subsequent refinement. Tables 2 and 3 therefore compare the complete contributions of the respective components.

C.3 BASELINE IMPLEMENTATIONS

SL-VAE, GCNSI, and NetSleuth use local implementations under the same input data and evaluation rules. PGSL† is the inverse prediction branch of the local PGSL implementation: it takes diffusion observations and network structure and directly predicts source scores. The training program also trains a forward model and a flow prior, but these components are not used in this branch at test time.

C.4 ARCHITECTURE AND OPTIMIZATION SETTINGS

Architecture and optimization. The observation encoder uses three graph-convolution blocks. For graphs with more than 512 nodes, a learned projection reduces the node dimension to 64 before global encoding; smaller graphs use flattening. Table 6 gives the hidden width h, latent dimension z, batch size B, propagation depth T, edge-probability scale β, and epoch budgets for the forward, proposal, and discriminator stages (F/P/D). All stages use AdamW (Loshchilov & Hutter, 2019)

with weight decay 10^{-4}. The learning rates are 10^{-3} for F and 2.5×10^{-4} for P and D. Gradient norms are clipped at 5 for F/P and 2 for D.

Table 6: Run-level settings. Epoch counts are maximum budgets before early stopping.

Setting	h/z	B	T	β	Epochs F/P/D
Cora-ML / SI	112/56	8	40	0.075	70/85/45
Cora-ML / SIR	112/56	8	50	0.090	70/85/45
Jazz / SIR	128/64	64	35	0.020	60/80/50
Network Science / SI	128/64	16	40	0.120	70/90/50
Network Science / SIR	128/64	16	45	0.220	70/90/50
Power Grid / SI	80/40	4	50	0.160	70/80/40
Power Grid / SIR	80/40	4	60	0.200	70/80/40

Loss weights. Forward training weights binary cross-entropy, Brier error, and Dice loss by 1, 0.5, and 0.25. Proposal training weights propagation fitting, structural guidance, source-count regularization, and the KL term by 1, 0.2, 0.4, and 0.01; the proposal has no direct source-label loss. Discriminator training weights the pair-ranking and node-supervision terms by 0.45 and 1.2, with margin 1 and soft top-K temperature 0.08. The node loss uses weights $(1, 1, 1.2, 0.15, 0.35)$ for weighted binary cross-entropy, Dice, pairwise ranking, listwise ranking, and cardinality, respectively. The positive-class weight is 2, hard-negative ratio is 3, ranking margin is 0.12, and listwise temperature is 0.3. Supervision on corrected swap candidates has relative weight 0.35.

Validation and inference. Early stopping uses patience 15 and an improvement tolerance of 10^{-4}. Forward checkpoints minimize validation BCE plus 0.5 times Brier error; proposal and discriminator checkpoints maximize $0.4\,\mathrm{AUC} + 0.6\,\mathrm{Top}\text{-}K\,\mathrm{F1}$. The correction strength is 1.25 during discriminator training. Validation searches $\{0, 0.25, 0.5, 1, 1.5\}$: nonzero strengths must improve the composite score by at least 2×10^{-4} over zero correction, and the eligible strength with the highest F1 is selected (1.5 in the recorded runs). Inference decodes the latent mean once and performs 20 Adam refinement iterations at learning rate 0.01. The fitting, structural, cardinality, and discriminator-energy weights are 1, 0.05, 0.1, and 0.02; the lowest evaluated composite objective is retained per sample.

During refinement, the graph-score branch uses candidate-level features without the encoder-feature projection H used for the initial correction. The settings in Table 6 cover the seven runs with available configuration records.

D Additional phenomenon experiments

To test the phenomenon under different observation conditions, we use one diffusion realization, set the propagation horizon to eight steps, and set false-positive and false-negative rates to 0.03 and 0.15. We retrain the initial source predictor for each setting. For each sample, we select the candidate with the lowest reporting-noise-aware diffusion-fit error among steps 0 through 20 and compare it with the initial estimate. Every setting contains 450 test samples.

All eight settings contain samples with better fit but worse localization, although the average effects differ. In Jazz/SI, mean F1 increases by 0.244 percentage points while 31.11% of samples become worse. In Network Science/SIR, the mean increases by 0.053 points while 29.11% become worse. In both Cora-ML settings, more than half of the samples lose F1. These results show why both average performance and per-sample changes are needed to assess the effects of refinement.

Table 7: Supplementary phenomenon experiment. The localization-decrease count and rate include samples with both lower fitting error and lower F1. ΔF1 is the post-refinement mean minus the initial mean, in percentage points.

Setting	Initial F1	Updated F1	Lower F1	Fraction	ΔF1
Jazz / SI	0.1580	0.1604	140/450	31.11%	+0.244
Jazz / SIR	0.1612	0.1580	160/450	35.56%	-0.322
Network Science / SI	0.8151	0.8151	157/450	34.89%	-0.004
Network Science / SIR	0.8251	0.8256	131/450	29.11%	+0.053
Cora-ML / SI	0.4442	0.4399	263/450	58.44%	-0.429
Cora-ML / SIR	0.4433	0.4401	258/450	57.33%	-0.319
Power Grid / SI	0.8663	0.8656	213/450	47.33%	-0.070
Power Grid / SIR	0.8741	0.8739	174/450	38.67%	-0.020

E Additional related work

Learned inversion and source-set search offer different ways to recover sources from diffusion. IVGD makes graph diffusion models invertible through residual constructions and uses error compensation and validity-aware layers to correct source estimates (Wang et al., 2022). BOSouL searches over source sets with simulation-based Bayesian optimization and a surrogate model for uncertain objective values (Zhang et al., 2024). These approaches separate the modeling of diffusion from the rule used to select or correct a source estimate. Our experiments examine this distinction by comparing changes in diffusion fit with changes in the recovered source set.

Generative approaches also model uncertainty in source inference. DDMSL represents forward propagation as a discrete Markov process and uses a reversible residual network to reconstruct sources and diffusion paths (Yan et al., 2023). SL-Diff first obtains a coarse estimate from source-proximity supervision and then uses denoising diffusion for finer inference (Huang et al., 2023). DDSL combines an invertible graph network with a diffusion-based source generator to represent alternative sources consistent with an observation (Liu et al., 2024). Trace studies transfer between networks by modeling source and final-state distributions with a structural Riemannian bridge (Sun et al., 2025). These studies concern richer inverse models and transfer settings; our analysis focuses on how a given source estimate changes after correction.

F COMPLETE RANKING RESULTS

Tables 8 and 9 give the AUC and AP values shown in Figure 4. Each entry is the mean over the test samples in that setting. PGSL† is defined in Appendix C.

Table 8: AUC for each network and diffusion setting.

Setting	GDSL	PGSL†	GCNSI	NetSleuth	SL-VAE
Jazz / SI	**0.952**	0.937	0.933	0.827	0.836
Jazz / SIR	**0.954**	0.941	0.936	0.827	0.837
Network Science / SI	**0.939**	0.920	0.922	0.862	0.866
Network Science / SIR	**0.947**	0.919	0.926	0.866	0.867
Cora-ML / SI	**0.987**	0.977	0.971	0.890	0.835
Cora-ML / SIR	**0.986**	0.975	0.969	0.891	0.832
Power Grid / SI	**0.962**	0.933	0.929	0.901	0.842
Power Grid / SIR	**0.975**	0.947	0.942	0.911	0.853

Table 9: AP for each network and diffusion setting.

Setting	GDSL	PGSL†	GCNSI	NetSleuth	SL-VAE
Jazz / SI	**0.681**	0.640	0.629	0.274	0.413
Jazz / SIR	**0.699**	0.664	0.649	0.272	0.386
Network Science / SI	**0.572**	0.554	0.567	0.329	0.382
Network Science / SIR	**0.646**	0.541	0.598	0.342	0.332
Cora-ML / SI	**0.892**	0.864	0.827	0.404	0.286
Cora-ML / SIR	**0.890**	0.859	0.821	0.411	0.287
Power Grid / SI	**0.690**	0.546	0.520	0.433	0.273
Power Grid / SIR	**0.753**	0.637	0.602	0.479	0.309