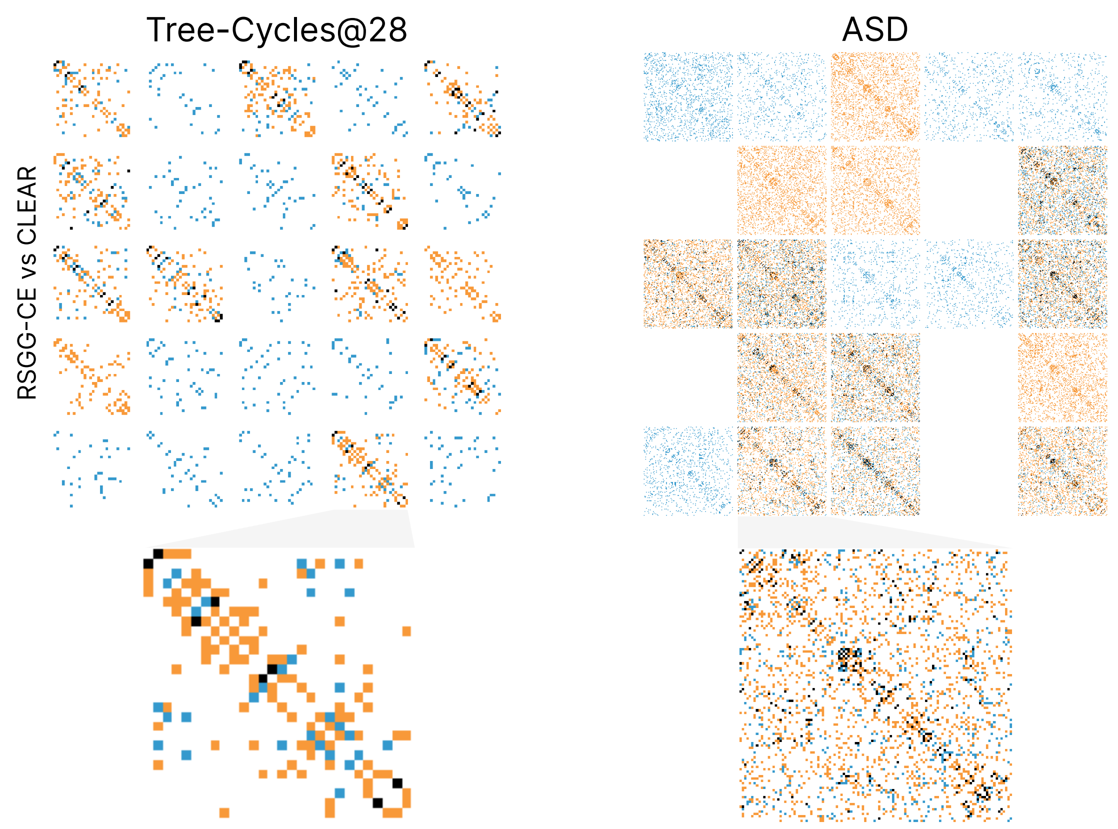
Robust Stochastic GrAph Generator for Counterfactual Explanations

Mario Alfonso Prado-Romero, Bardh Prenkaj, Giovanni Stilo

aiimlab.org

The 38th Annual AAAI Conference on Artificial Intelligence
FEBRUARY 20-27, 2024 | VANCOUVER, CANADA

TL;DR: We propose RSGG-CE that leverages graph-based GANs and the generator's learned latent space to generate plausible and valid counterfactual candidates.



1 Graph Counterfactual Explainability

- Generative Graph Counterfactual Explainability (**GCE**)
- SoA is generally **constrained** to the **input data** (search-based GCE) and relies on learned **perturbation masks** (learning-based GCE)
- Defaulting to factual-based explainers falters when dual classes clash (e.g., acyclic vs cyclic graphs)
- Crossing the decision boundary isn't enough; one must be close to the original instance

2 How the literature approached GCE

- Learning-based GCE [1-5] generate masks of relevant features given a graph G; combine this mask with G to derive G'; feed G' to the oracle Φ and update the mask
- CLEAR [5] uses a VAE to encode graphs into a latent representation which, at inference, is used to generate complete stochastic graphs.
- G-CounteRGAN [6,7] relies on 2D convolutions on the adjacency matrix of graphs

[1] Abrate, C.; and Bonchi, F. 2021. Counterfactual graphs for explainable classification of brain networks. In KDD'21

[2] Liu, Y.; Chen, C.; Liu, Y.; Zhang, X.; and Xie, S. 2021. Multi-objective Explanations of GNN Predictions. In ICDM'21

[3] Nguyen, T. M.; Quinn, T. P.; Nguyen, T.; and Tran, T. 2022. Explaining Black Box Drug Target Prediction through Model Agnostic Counterfactual Samples. IEEE/ACM Transactions on Computational Biology and Bioinformatics

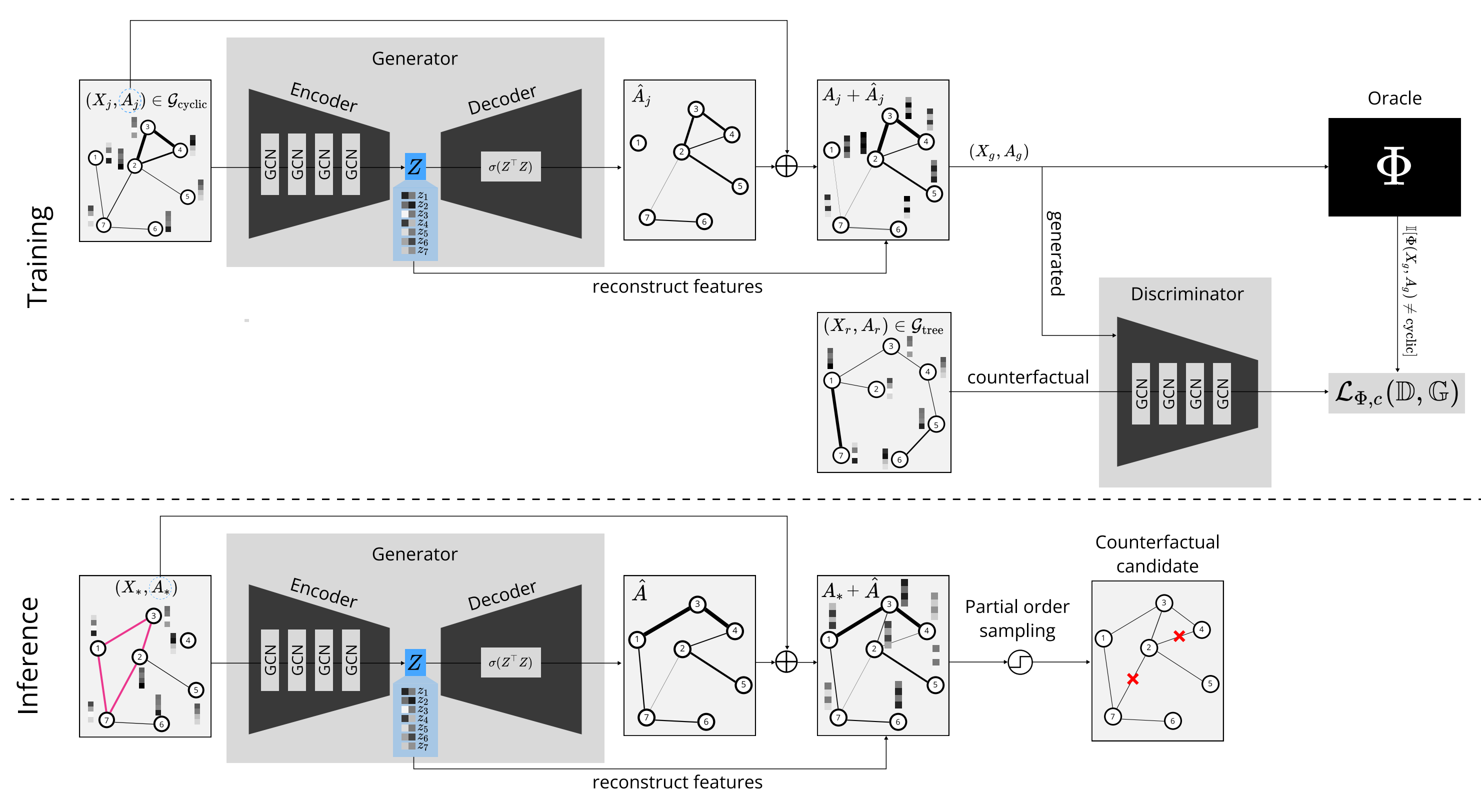
[4] Numeroso, D.; and Bacciu, D. 2021. Meg: Generating molecular counterfactual explanations for deep graph networks. In IJCNN'21

[5] Ma, J.; Guo, R.; Mishra, S.; Zhang, A.; and Li, J. 2022. CLEAR: Generative Counterfactual Explanations on Graphs. In NeurIPS'22

[6] Nemirovsky, D.; Thiebaut, N.; Xu, Y.; and Gupta, A. 2022. CounteRGAN: Generating counterfactuals for real-time recourse and interpretability using residual GANs. In UAI'22

[7] Prado-Romero, M. A.; Prenkaj, B.; and Stilo, G. 2023. Revisiting CounteRGAN for Counterfactual Explainability of Graphs. In ICLR'23 @ Tiny Paper Track

3 Proposed Approach



- RSGG-CE's **discriminator guides** the generator to learn the production of **counterfactuals** aligned with the opposite class
- **Training**: Modify the generator's optimization and include the oracle's predictions in the discriminator on the generated data

$$\mathcal{L}_{\Phi,c}(\mathbb{D},\mathbb{G}) = \sum_{(X_r,A_r)\in\mathcal{G}_{\neg c}} \Big(\log \mathbb{D}(Y \mid X_r, A_r)\Big)$$

discriminator optimisation on real data

$$+ \sum_{(X_g,A_g)\in\mathbb{G}(\mathcal{G}_c)} \Big(\mathbb{I}[\Phi(X_g, A_g) \neq c] \log \mathbb{D}(Y \mid X_g, A_g)\Big)$$

discriminator optimisation on generated data

$$+ \sum_{\substack{(X_j,A_j)\in\mathcal{G}_c,\\ \hat{X}_j, A_j+\hat{A}_j=\mathbb{G}(X_j,A_j)}} \log\Big(1 - \mathbb{D}(Y \mid \hat{X}, A_j + \hat{A}_j)\Big)$$

generator optimisation on the counterfactual data

- **Inference**: Sample edges with partial order guided by the learned probabilities from the generator's latent space to generate counterfactuals

Algorithm 1: Partial order sampling to produce a counterfactual.

Require: $G_* = (X_*, A_*), \mathbb{G} : \mathcal{G} \to \mathcal{G}, \Phi,$
$\hat{X}_*, A_* + \hat{A}_* = \mathbb{G}(X_*, A_*)$
$X_g, A_g \leftarrow \hat{X}_*, A_* + \hat{A}_*$
$\mathcal{P} \leftarrow$ `partial_order`(A_*)
$A' \leftarrow 0^{n\times n}$
for $\mathbb{O} \in \mathcal{P}$ **do**
  for $e = (u, v) \in \mathbb{O}.\mathcal{E}$ **do**
    $A'[u, v] \leftarrow$ `sample`$(e, A_g[u, v])$
    if $\mathbb{O}.o \wedge \Phi(X_g, A') \neq \Phi(X_*, A_*)$ **then**
      return (X_g, A')
    end if
  end for
end for
return (X_*, A_*)

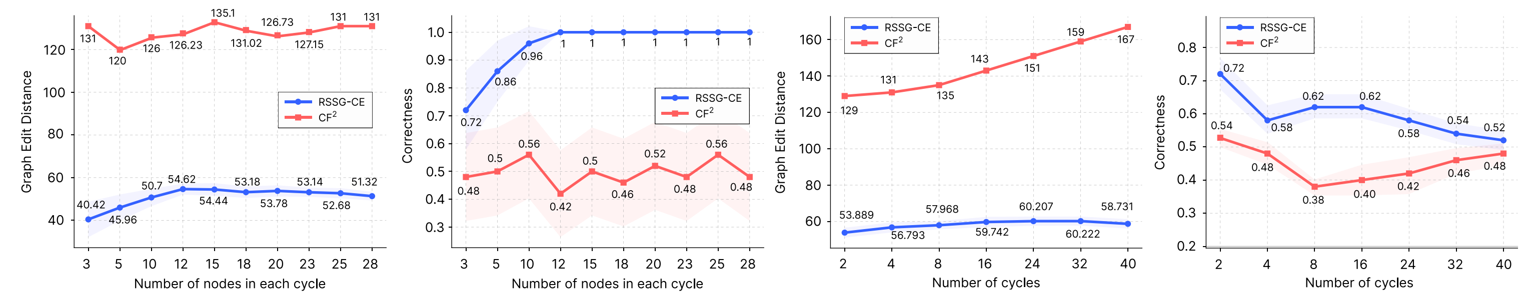
Algorithm 2: Example of `partial_order`

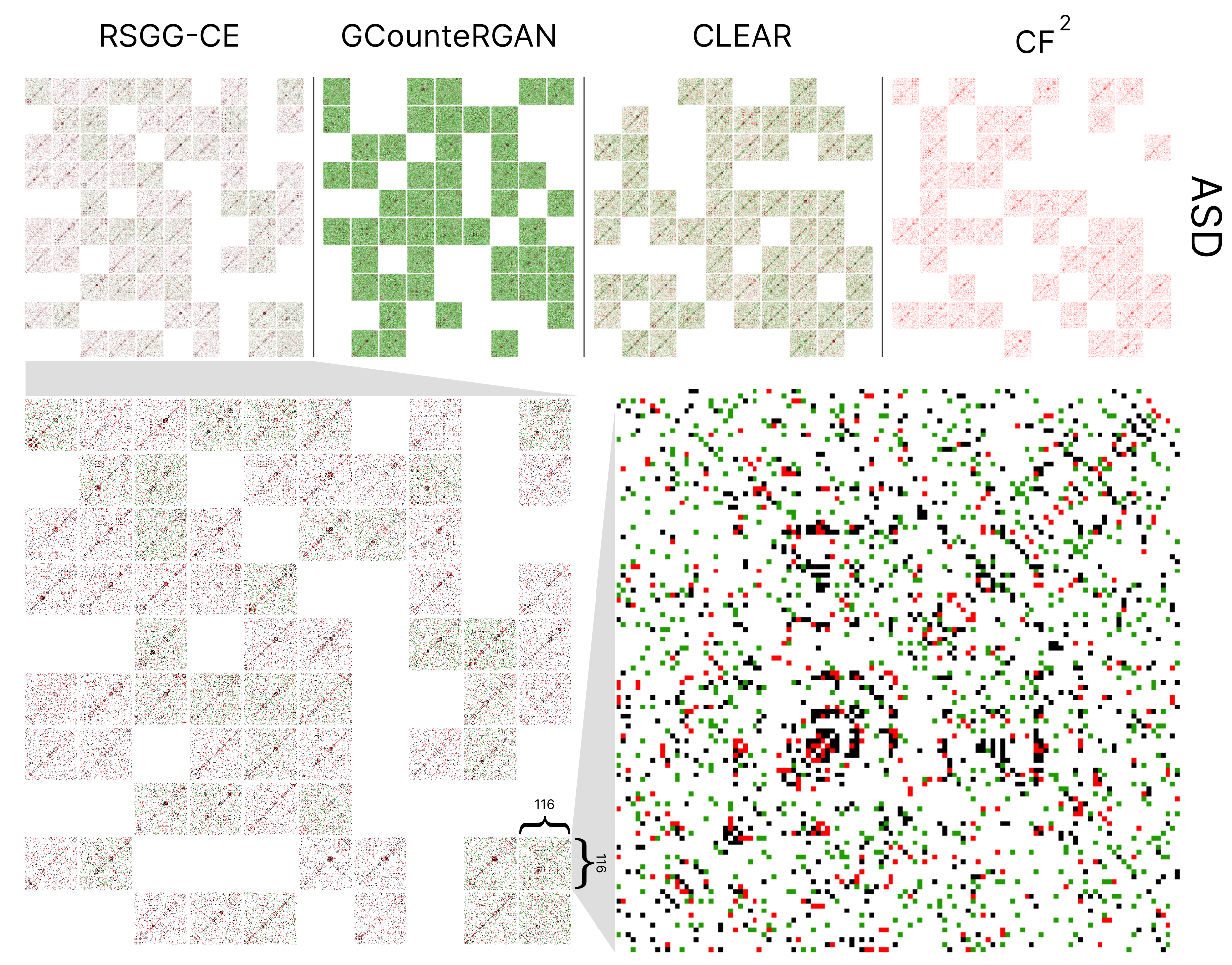
Require: $A \in \mathbb{R}^{n\times n}$
$E \leftarrow$ `positive_edges`(A) ▷ *Get the set of edges from the adjacency matrix A*
$\neg E \leftarrow$ `negative_edges`(A) ▷ *Get the set of non-existing edges from the adjacency matrix A*
$\mathcal{P} \leftarrow \{(\mathcal{E}{=}E, o{=}0), (\mathcal{E}{=}\neg E, o{=}1)\}$ ▷ *Build the partial order of the existing and non-existing edges with group tuples consisting of edge set $\mathcal{E}$, and oracle verification guard o.*
return $\mathcal{P}$

4 Take-away lesson

		Methods				
		MEG †	CF^2 †	CLEAR ‡	G-CounteRGAN ‡	**RSGG-CE ‡**
TC	Runtime (s) ↓	272.110	<u>4.811</u>	25.151	632.542	**0.083**
	GED ↓	159.700	<u>27.564</u>	61.686	182.414	**11.000**
	Oracle Calls ↓	**0.000**	**0.000**	4341.600	1321.000	<u>121.660</u>
	Correctness ↑	<u>0.530</u>	0.496	0.504	0.504	**0.885**
	Sparsity ↓	2.510	0.496	1.110	3.283	**0.199**
	Fidelity ↑	<u>0.530</u>	0.496	0.504	0.504	**0.885**
	Oracle Acc. ↑	1.000	1.000	1.000	1.000	1.000
ASD	Runtime (s) ↓	×	**15.313**	275.884	969.255	<u>80.000</u>
	GED ↓	×	<u>655.661</u>	1479.114	3183.729	**234.853**
	Oracle Calls ↓	×	**0.000**	5339.455	1182.818	<u>794.805</u>
	Correctness ↑	×	0.463	<u>0.554</u>	0.529	**0.603**
	Sparsity ↓	×	<u>0.850</u>	1.917	4.125	**0.304**
	Fidelity ↑	×	**0.287**	<u>0.319</u>	0.265	**0.287**
	Oracle Acc. ↑	×	0.773	0.773	0.773	0.773

- RSGG-CE is the best performer with a gain of **66.98%** and **19.65%** in Correctness over the second-performing method in TC and ASD





- RSGG-CE can do **both** edge additions and removals
- RSGG-CE **scales perfectly** when the number of nodes in a cycle increases since its **GED plateaus** reaching a perfect **correctness of 1**

GRAN SASSO SCIENCE INSTITUTE · SAPIENZA UNIVERSITÀ DI ROMA · UNIVERSITÀ DEGLI STUDI DELL'AQUILA · SOBIGDATA RESEARCH INFRASTRUCTURE · ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing · TERRITORI APERTI