Explaining Deep Graph Networks with Molecular Counterfactuals

Danilo Numeroso, Davide Bacciu
danilo.numeroso@phd.unipi.it, bacciu@di.unipi.it

@NeurIPS2020
Workshop on Machine Learning for Molecules

Motivations

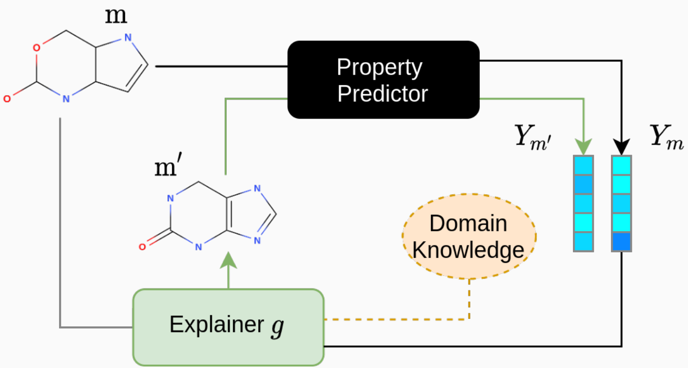
- Deep Graph Networks (DGN) are **ubiquitous**, even in safety-critical tasks, i.e, drug discovery.
- Need of explainability techniques
- So far, only a few DGN explainability methods in literature and no counterfactuals yet.

Why counterfactuals?

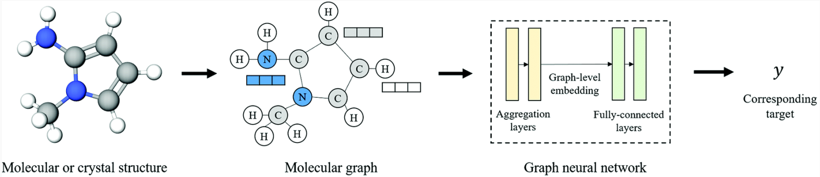
- **Easy to interpret** for domain experts.
- **Sanity check** of existing local explanation methods.

MEG: Molecular Explanation Generator

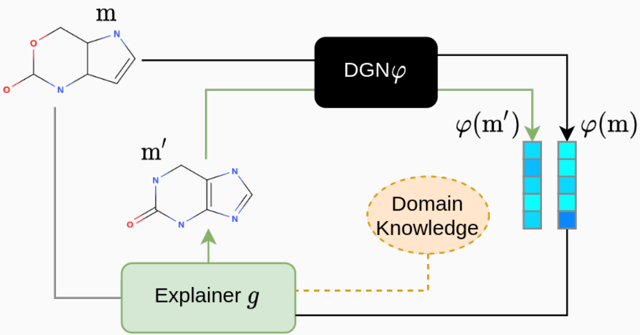
Generating (valid) molecular compounds acting as counterfactual explanations, through an RL-based agent.



Deep Graph Networks (DGNs) [1] are a variant of Deep Neural Networks that can learn patterns over graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ by aggregating node and neighbours information.



Davide Bacciu, Federico Errica, Alessio Micheli, and Marco Podda.
A gentle introduction to deep learning for graphs.
Neural Networks, 129:203–221, Sep 2020.


m
O
O
N
N
DGNφ
φ(m′)
φ(m)
m′
N
N
N
N
O
Domain Knowledge
Explainer g

Problem Formalisation

The learning problem for the RL agent takes the form of a MDP($\mathcal{S}$, $\mathcal{A}$, $\mathcal{Q}$, π, $\mathcal{R}$):

1. $\mathcal{S}$ is the molecular space.
2. $\mathcal{A}$ is the action state comprising molecule alteration actions preserving **chemical validity**.
3. $\mathcal{Q}$ and π are respectively the the learnt action value function and policy.
4. $\mathcal{R}$ is a multi-objective reward function.

$$\arg\max_{m'} \mathcal{R}(m, m^{'}) = \mathcal{L}\big(\varphi(m), \varphi(m')\big) + \mathcal{K}\big[m, m'\big].$$

- $\mathcal{K} \equiv$ **resemblance** with the molecule m under study.
- $\mathcal{L} \equiv$ **distance** between $\varphi(m')$ and $\varphi(m)$.

Similarity metrics

1. **Tanimoto similarity – Structural Similarity**:

$$\mathcal{T}(m, m') = \frac{f_m \cdot f_{m'}}{\|f_m\|^2 + \|f_{m'}\|^2 - f_m \cdot f_{m'}}$$

2. **Neural encoding similarity – Model Perception**:

$$\mathcal{K}[m, m'] = \frac{\mathsf{h}_m \cdot \mathsf{h}_{m'}}{\|\mathsf{h}_m\|\|\mathsf{h}_{m'}\|}$$

3. **Convex combination** of the two – **Trade-off**:

$$\alpha_1 \mathcal{T}(m, m') + \alpha_2 \mathcal{K}(m, m') \mid \sum_i \alpha_i = 1$$

Experiments on two different datasets:

1. **Tox21**; binary classification task on molecule toxicity, given m classified as c:

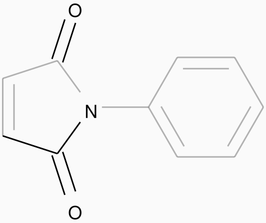
$$\arg\max_{m'} \mathcal{L}_g = \arg\max_{m'} \alpha(1 - y_c) + (1-\alpha)\mathcal{K}[m', m]$$

2. **ESOL**; regressive task on water solubility:

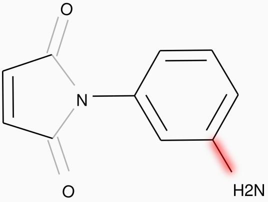
$$\arg\max_{m'} \alpha \,\mathrm{sgn}\left(\|s_{m'} - s\|_1 - \|s_m - s\|_1\right)\|s_{m'} - s_m\|_1 + (1-\alpha)\mathcal{K}[m', m]$$

We compare our method to **GNNExplainer**.

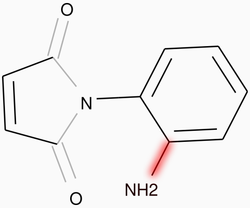
Some results (i)



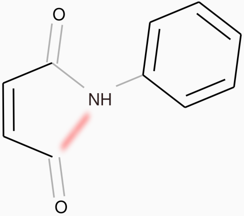
(a) A0 $\rightarrow$ NoTox 70%



(b) A1 $\rightarrow$ Tox 90% Sim 0.76

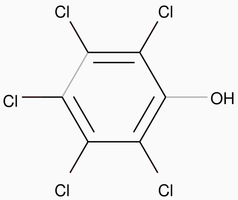


(c) A2 $\rightarrow$ Tox 83% Sim 0.79

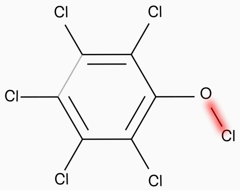


(d) A3 $\rightarrow$ Tox 80% Sim 0.68

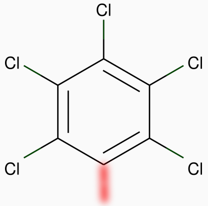
Some results (ii)



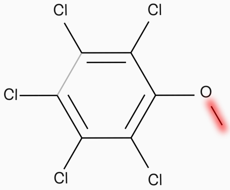
(e) B0 → Solubility -4.01 ≈ Target -4.28



(f) B1 → Solubility -6.11 Sim 0.29



(g) B2 → Solubility -5.93 Sim 0.31



(h) B3 → Solubility -5.07 Sim 0.28

Wrap-Up

1. **Generating counterfactual** explanations easy to understand for domain experts.
2. **Sanity check** on other local interpretability approaches.
3. Need of **experts supervision**.
4. **Optimise diversity** on the set of produced counterfactual.
5. Build further local explanation methods upon the detected counterfactuals for automatic explanations.

References

Check out our implementation and the paper for more details:

- **Code**: github.com/danilonumeroso/MEG
- **Paper**: arxiv.org/pdf/2011.05134.pdf

You can reach out to me at danilo.numeroso@phd.unipi.it , in case you have any doubts or questions.