ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1.
Available from http://www.i6doc.com/en/.

Improving Zorro Explanations for Sparse Observations with Dense Proxy Data

Andreas Mazur*, André Artelt† and Barbara Hammer ‡

CITEC – Cognitive Interaction Technology
Bielefeld University – Faculty of Technology
Inspiration 1, 33619 Bielefeld – Germany

Abstract. Explanation methods are considered the most prominent way of achieving the ubiquitous requirement of transparency. Ideally, in order to be useful, explanations should be "easy to understand" – i.e. being of low complexity. In this work, we empirically study explanations generated by *Zorro*, an explanation method for Graph Neural Networks. In the context of a standard reinforcement learning scenario, we propose a methodology to improve the quality of generated explanations in case of sparse observations.

1 Introduction

The ubiquitous requirement of transparency of machine learning (ML) based systems is usually realized by providing explanations of system's behavior. Nowadays, there exist a wide variety of different explanation methodologies for ML systems [1]. Also, for the relatively new Graph Neural Networks (GNNs), explanation methods have been developed, although these are still not as far developed as those for classic ML systems [2]. In order to be useful, an explanation should be of low-complexity – i.e. "easy to understand". Therefore it must balance between showing enough information but not too much. In this work, we consider explanations generated by Zorro [3], an explanation method for Graph Neural Networks. Our experiment shows that Zorro relies on the amount of information presented in its input. Sparse inputs cause Zorro to yield non-meaningful explanations. We therefore propose to substitute sparse inputs with dense proxy data. We evaluate our method on a standard reinforcement learning scenario, with sparse observations, where the policy is realized using a GNN.

2 Background

Reinforcement Learning. In reinforcement learning an agent interacts with an environment $\mathcal{E}$ by selecting appropriate actions a from an action set $\mathcal{A}$ for states $s \in \mathcal{E}$ [4]. Numerical rewards r_i, obtained for each chosen action, allow to assess the behavior of the agent via the *expected discounted return* $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$ with $\gamma \in [0, 1[$ [4]. This measure, however, indirectly depends on the agent's policy $\pi(a|s)$ which represents a conditional random distribution over the action

*Corresponding author: amazur@techfak.uni-bielefeld.de
†Affiliation with the University of Cyprus
‡We gratefully acknowledge funding from the VW-Foundation for the project *IMPACT* funded in the frame of the funding line *AI and its Implications for Future Society*.

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

space. $\mathbb{E}_\pi[G_t \mid s_t = s,\ a_t = a]$ yields the action-value function $q_\pi(s, a)$ [4]. Mnih et al. introduced the algorithm "*deep Q-learning with experience replay*" to estimate $q_\pi(s, a)$ with a deep Q-network $Q(s, a; \theta) \approx q_\pi(s, a)$ in [5]. Training the network improves the agent's policy. In essence, the sequence of loss functions $L_i(\theta_i) = \mathbb{E}_{(s_t,a_t)\sim\rho}\left[(y_i - Q(s_t, a_t; \theta_i))^2\right]$ with $y_i = r_{t+1}+\gamma \max_{a'} Q(s_{t+1}, a'; \theta_{i-1})$ is minimized via batch gradient descent with data from the *replay memory*. This data set stores experience tuples that are collected by the agent. Van Hasselt [6] pointed out that Q-Learning and therefore deep Q-Learning overestimates $q_\pi(s, a)$-values due to using only a single estimator. This significantly disrupts the training [4, 6]. Decoupling the action selection from the action's evaluation counteracts this issue [6]. Hence, in deep Q-learning we adjust the original y_i to $y_i = r_{t+1} + \gamma\ Q(s_{t+1}, \text{argmax}_{a'} Q(s_{t+1}, a'; \theta^{(1)}); \theta^{(2)})$.

Graph Convolutional Neural Networks. In this work we consider the graph convolutional neural networks (GCNs) which were introduced by Kipf and Welling in [7]. GCNs compute node-wise feature embeddings by propagating weighted messages among immediate neighbors $H^{l+1} = \sigma\left(D^{-1/2}(A + \mathbb{1})D^{-1/2}H^l W^l\right)$. This propagation rule was derived by approximating spectral graph convolutions with first-order Chebyshev polynomials [7, 8]. For an in-depth derivation we refer to [7] and [9].

Zorro. Zorro [3] yields a post-hoc, instance-level explanation algorithm for graph neural networks. It uses a rate-distortion framework to compute discrete masks which are supposed to explain node-level predictions. The algorithm searches in a pre-defined sub-graph around the target node for important nodes and node features. Those are marked in a discrete matrix M, i.e. the mask for which holds $M_{ij} = 1$ iff node i and feature j are included in the explanation, else $M_{ij} = 0$. Let X be the original feature matrix of the given graph and $\mathcal{R}$ a distribution over the feature space. Multiple $Y_M = X \odot M + Z \odot (\mathbb{1} - M)$ are computed with differently sampled $Z \sim \mathcal{R}$. Predictions of each Y_M are computed with the given graph neural network. Let m be the amount of all computed Y_M and n the amount of predictions of Y_M which match the prediction of X. The fidelity $\mathcal{F}(M) = n/m$ measures the quality of M. Overall, Zorro greedily computes a set of disjoint explanations $\mathcal{S} = \{M_1, M_2, ... \mid \forall i : F(M_i) \geq \tau,\ \cap_i M_i = \emptyset\}$ which yield a fidelity higher than a given threshold fidelity τ.

3 Case Study

3.1 Setup

We empirically evaluate our contribution by training a RL-agent on the Taxi-v3 OpenAI-Gym environment [10, 11]. In the Taxi-v3 environment a taxi has to move through a (5×5)-grid in order to pick-up a passenger, subsequently moving to- and eventually drop the passenger on a destination location. The default observations are given as integers. We convert those to graphs. In practice, the graph is represented with a binary feature matrix $F \in \mathbb{N}^{25\times4}$ and an adjacency matrix $A \in \mathbb{N}^{25\times25}$. All possible transitions in the environment are represented

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

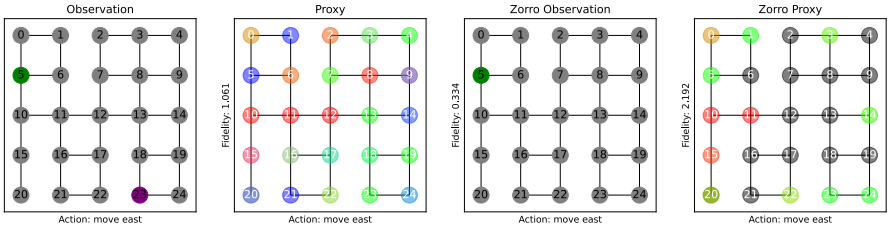

Fig. 1: [**very left**] taxi and passenger (picked up) located in node 5, destination on node 23 [**left**] proxy with feature vectors holding probabilities of the occurrence of each feature displayed in RGBA-encoding [**right**] Zorro applied on observation [**very right**] Zorro applied on proxy

as edges in the graph which never change. Each node is associated to a four-dimensional binary feature vector. The dimensions are described in Table 1. An entry is one if the corresponding attribute holds, otherwise zero. Dimensions zero and one are zero if dimension three is one. Note that the feature matrix has a total of 100 entries but at most three have a value different from zero. That is, we deem the observations to be sparse. An illustration of this observation-encoding is given in Fig. 1 on the very left. We use double deep Q-learning in order to train the agent on the modified environment. The deep Q-network we use is illustrated in the upper branch of the graph neural network represented in Fig. 2. It computes a graph embedding with general graph convolutions [12], reduces the amount of nodes to one via differential pooling [13] and predicts Q-values for the feature vector of the remaining node. Eventually, the index of the largest Q-value represents the chosen action. We use Zorro [3] to seek explanations for remaining node's classification. The code for this experiment will be made public[1] after publication.

Dim.	Meaning
0.	taxi located in this node
1.	passenger located in this node
2.	destination located in this node
3.	taxi and the passenger located in this node & passenger picked up by the taxi

Table 1: Feature vector

3.2 Applying Zorro on Sparse Data

In order to illustrate what regions of the observations are of particular interest for the agent we apply Zorro on the observation. As the hyper-parameters of Zorro we choose $\tau = 0.7$ and compute $\mathcal{F}(M)$ considering 300 sampled random matrices Z per M. If Zorro's result set $\mathcal{S}$ holds a subset $\widetilde{S}$ that contains masks for which the predictions match the original prediction, we select that $M \in \widetilde{S}$ which has the highest fidelity Eq.(2) to increase sparsity as a means to improve interpretability. If $\widetilde{S} = \emptyset$, we select that $M \in \mathcal{S}$ with lowest fidelity in $\mathcal{S}$. The final explanation E_X for an observation X is given by the Hadamard product $E_X = X \odot M$ ("Zorro Observation" in Fig. 1). We evaluate the obtained explanations by considering the following three evaluation metrics:

[1] `https://github.com/andreasMazur/ImprovedGNNExplanations`

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

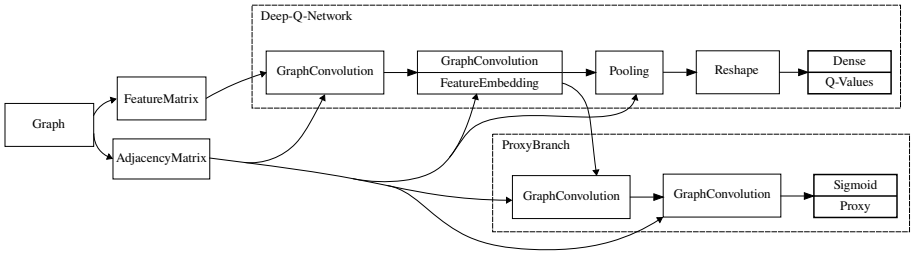

Fig. 2: The network architecture used throughout this work. The upper branch illustrates the network used to determine actions, the lower is used to compute proxies.

Metric 1 (Sparsity) *Let $Y \in \mathbb{R}^{m \times n}$ be a feature matrix or an explanation and y_{ij} an entry in Y. We calculate the sparsity of Y as follows:*

$$sparsity(Y) = \frac{\left|\{y_{ij} \mid i \in \{1, ..., m\},\ j \in \{1, ..., n\},\ y_{ij} = 0\}\right|}{m \cdot n} \tag{1}$$

Metric 2 (Fidelity) *In order to understand how similar a deep Q-network Φ assesses the explanation E_X to its observation X, we consider the mean-squared-error between Q-values $\Phi(X) = \vec{q}$ and $\Phi(E_X) = \hat{\vec{q}}$ as the fidelity of E_X:*

$$MSE(\vec{q}, \hat{\vec{q}}) = \frac{1}{n} \sum_{i=0}^{n-1} (q_i - \hat{q}_i)^2 \tag{2}$$

Metric 3 (Prediction Accuracy) *Let Φ be the deep Q-network, X_i a feature matrix of an observation and E_{X_i} the explanation of X_i. We compute the prediction accuracy as follows:*

$$acc\big(\{(X_i, E_{X_i})\}_{i=1...k}\big) = \frac{\left|\{E_{X_i} \mid arg\,max\,\Phi(E_{X_i}) = arg\,max\,\Phi(X_i)\}\right|}{k} \tag{3}$$

It turns out that for 1101 *observations, which we have considered during this work, the most prominent target action ("move south") appears* 303 *times. A baseline agent predicting "move south" for every observation would have an accuracy of* ≈ 0.275.

After computing the explanations (orange data in Fig. 3) we measured an average explanation sparsity of 0.983, a mean fidelity of 20.522 and a prediction accuracy of 0.64. However, we recognized that a lot of explanations were either entirely equal to the original observation (519 times) or completely empty (80 times). That leaves us 502 valid explanations. Hence, we have explanations for only about $\approx 46\%$ of all observations. While the explanation sparsity does not change significantly when leaving out the faulty explanations, the prediction accuracy drops to 0.092. This is worse then the baseline agent. Considering the poor performance and the fact that a lot of explanations cannot be considered valid stresses Zorro's problem with sparse data. This issue is addressed in the following section.

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

3.3 Predicting Dense Proxy Data

Since Zorro has difficulties dealing with sparse input data we propose to substitute the sparse original data with data the deep Q-network interprets similar to the original observations. We extend the deep Q-network architecture we used so far by connecting a new branch, called *proxy branch*, to the feature embedding layer as illustrated by the lower branch in Fig. 2. The proxy branch shall compute dense proxy data which we use as replacements for the original input observations (therefore the notion "*proxy*"). That is, the branch's output is a matrix with the same dimensions as the feature matrix from the observations. It contains floating point values in range zero to one, which can be interpreted as probabilities for the occurrence of a particular feature in a considered node. The Q-values predicted for the proxy inputs shall be similar to the Q-values predicted for the original feature matrix. Therefore, we use the MSE between the observation Q-values and the proxy Q-values as the loss function Eq. (2). During the training we freeze all weights in the deep Q-network besides the weights of the proxy branch. We can compare the proxy data to the previously computed explanations (green data in Fig. 3) by computing the evaluation metrics. We measured a mean sparsity of 0.037, a mean fidelity of 0.824 and a prediction accuracy of 0.763. Although the proxy inputs largely improve the fidelity and the prediction accuracy compared to Zorro's explanations for sparse observations, we cannot consider them as good explanations as their high density makes it very difficult to fell meaningful interpretations ("Proxy" in Fig. 1).

3.4 Applying Zorro on Proxy Data

In order to improve the sparsity we apply Zorro on the proxy data in the same manner as we did for the sparse observations in Section 3.2. As visible in Fig. 1 on the very right, applying Zorro on proxy data can yield explanations which distinguish from the previous explanations by balancing sparsity and information richness. When applying Zorro on the proxy data it benefits from a larger pool of selectable nodes and features since more nodes have feature vectors different from a zero vector in contrast to the original observation. Hence, it is not prone to mask too much or nothing unlike the first test series as reflected by the blue data in Fig. 3. Calculating the evaluation metrics for those explanations, we witness a mean sparsity of 0.444, a mean fidelity of 28.326 and a prediction accuracy of 0.41. That is, these explanations are not too dense like the proxy data and not too sparse like Zorro's explanations for the observations. Additionally, compared to the previous explanations the prediction accuracy im-

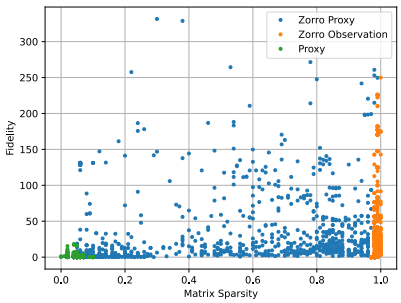

Fig. 3: For each observation in a given set of 1101 observations a proxy and two explanations, i.e. Zorro applied on observation and proxy, were computed.

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

proved by more than four times and therefore also outperforming the baseline agent.

4 Conclusion

Proxy data resembles original observations up to their fidelity while being a lot more dense, therefore representing reasonable substitutes. Zorro applied onto proxy data therefore benefits from a large pool of selectable nodes and features, making it less prone to mask too much or no information at all. The values given by our evaluation metrics emphasize that. We measured matrix sparsity values for proxy explanations which are not vanishingly low, compared to plain proxy data, or immoderately high, like Zorro's explanations for sparse observations, but rather values which are in a reasonable mid-range. Additionally, the prediction accuracy for proxy explanations is vastly better than the prediction accuracy for explanations of sparse observations. Considering all these findings, we deem Zorro's explanations for dense proxy data to be improved compared to Zorro's explanations for sparse observations.

References

[1] Christoph Molnar. *Interpretable Machine Learning*. 2019.

[2] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *arXiv preprint arXiv:2012.15445*, 2020.

[3] Thorben Funke, Megha Khosla, Mandeep Rathee, and Avishek Anand. Zorro: Valid, sparse, and stable explanations in graph neural networks. *arXiv:2105.08621*, 2021.

[4] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[5] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[6] Hado Hasselt. Double q-learning. *Advances in neural information processing systems*, 23, 2010.

[7] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2016.

[8] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

[9] Thomas N Kipf. *Deep learning with graph-structured representations*. 2020.

[10] Thomas G Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of artificial intelligence research*, 13:227–303, 2000.

[11] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[12] Jiaxuan You, Zhitao Ying, and Jure Leskovec. Design space for graph neural networks. *Advances in Neural Information Processing Systems*, 33:17009–17021, 2020.

[13] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31, 2018.