Distill n' Explain: explaining graph neural networks using simple surrogates

Tamara Pereira[1], Erik Nascimento[1], Lucas E. Resck[2], Diego Mesquita[2], Amauri Souza[1,3]

[1]Federal Institute of Ceará [2]Getulio Vargas Foundation [3]Aalto university

Overview

Generating explanations for GNN predictions usually implies back-propagating through the GNN or repeatedly learning local surrogate models.

Can we use less compute and explain a simple global GNN surrogate instead?

- We propose Distill n' Explain (DnX) a new framework for GNN explanations that hinges on explaining a simple surrogate model obtained through knowledge distillation;
- DnX comprises two steps: knowledge distillation and explanation extraction.
- We provide theoretical bounds on the quality of explanations based on these surrogates;
- The results achieved show that our methods outperform the prior art while running orders of magnitude faster.

DnX: Distill n' Explain

Knowledge distillation

In this step, we use a linear GNN Ψ to approximate the predictions from the GNN Φ we want to explain, for that we minimize the Kullback-Leibler divergence KL between the predictions of Φ and Ψ_Θ. Let $\hat{Y}_i^{(\Psi_\Theta)}$ and $\hat{Y}_i^{(\Phi)}$ denote the class predictions for node i from the Ψ_Θ and Φ models, respectively:

$$\min_{\Theta} \left\{ \mathrm{KL}\left(\hat{Y}^{(\Phi)}, \hat{Y}^{(\Psi_\Theta)}\right) \coloneqq \sum_{i \in V} \sum_{c} \hat{Y}_{ic}^{(\Phi)} \log \frac{\hat{Y}_{ic}^{(\Phi)}}{\hat{Y}_{ic}^{(\Psi_\Theta)}} \right\}. \tag{1}$$

Explanation extraction

To obtain an explanation to a given prediction $\hat{Y}_i^{(\Psi_\theta)}$, we want to identify a subgraph of $\mathcal{G}$ containing the nodes that influence the most that prediction. We denote an explanation $\mathcal{E}$ as an n-dimensional vector of importance scores. We introduce two strategies to compute $\mathcal{E}$.

1. **DnX: Optimizing for $\mathcal{E}$.** We can formulate the problem of finding the explanation $\mathcal{E}$ by treating it as a vector of 0-1 weights, and minimizing the squared L_2 norm between the logits associated with $\hat{Y}_i^{(\Psi_\Theta)}$ and those from the graph with node features masked by $\mathcal{E}$:

$$\min_{\mathcal{E} \in \{0,1\}^n} \| \widetilde{A}_i^L \mathrm{diag}(\mathcal{E}) X\Theta - \widetilde{A}_i^L X\Theta \|_2^2, \tag{2}$$

where $\widetilde{A}_i^L$ denotes the i-th row of the matrix $\widetilde{A}^L$. But this formulation in 2 admits the trivial solution $\mathcal{E} = [1, 1, \ldots, 1]$. To circumvent the issue and simultaneously avoid binary optimization, we replace the search space $\{0,1\}^n$ by the $(n-1)$-simplex $\Delta = \{r \in \mathbb{R}^n : \sum_i r_i = 1, \forall_i r_i \geq 0\}$:

$$\min_{\mathcal{E} \in \Delta} \left\| \widetilde{A}_i^L \left(\mathrm{diag}(\mathcal{E}) - I_n\right) X\Theta \right\|_2^2. \tag{3}$$

2. **FastDnX: Finding $\mathcal{E}$ via linear decomposition.** Let Z_i denote the logit vector associated with the prediction $\hat{Y}_i^{(\Psi_\Theta)}$. Due to the linear nature of Ψ, we can decompose Z_i into a sum of n terms, one for each node in V (plus the bias):

$$\widetilde{A}_{i1}^L X_1\Theta + \widetilde{A}_{i2}^L X_2\Theta + \ldots + \widetilde{A}_{in}^L X_n\Theta + b = Z_i. \tag{4}$$

Therefore, we can measure the contribution of each node to the prediction as its scalar projection onto $Z_i - b$:

$$\mathcal{E}_j \coloneqq \widetilde{A}_{ij}^L X_j \Theta (Z_i - b)^\intercal \tag{5}$$

Analysis

Definition (Faithfulness). Given a set $\mathcal{K}$ of perturbations of $\mathcal{G}_u$, an explanation $\mathcal{E}_u$ is faithful to a model f if

$$\frac{1}{|\mathcal{K}|+1} \sum_{\mathcal{G}'_u \in \mathcal{K} \cup \{\mathcal{G}_u\}} \left\| f(\mathcal{G}'_u) - f(t(\mathcal{G}'_u, \mathcal{E}_u)) \right\|_2 \leq \delta,$$

where $\mathcal{G}'_u$ is a possibly perturbed version of $\mathcal{G}_u$, t is a function that applies the explanation $\mathcal{E}_u$ to the graph $\mathcal{G}'_u$, and δ is a small constant.

Lemma 1 (Unfaithfulness with respect to Ψ). Given a node u and a set $\mathcal{K}$ of perturbations, the unfaithfulness of the explanation $\mathcal{E}_u$ with respect to the prediction $Y_u^{(\Psi_\Theta)}$ of node u is bounded as follows:

$$\frac{1}{|\mathcal{K}|+1} \sum_{\substack{\mathcal{G}'_u \in \\ \mathcal{K} \cup \{\mathcal{G}_u\}}} \left\| \Psi(\mathcal{G}'_u) - \Psi(t(\mathcal{G}'_u, \mathcal{E}_u)) \right\|_2 \leq \gamma \left\| \underset{\mathcal{E}_u}{\Delta} \widetilde{A}_u^L \right\|_2 .$$

Theorem 1 (Unfaithfulness with respect to Φ). Under the same assumptions of Lema 1 that provides an upper bound on the unfaithfulness of $\mathcal{E}_u$ with respect to the surrogate model Ψ and assuming the L_2 distillation error is bounded by α, the unfaithfulness of the explanation $\mathcal{E}_u$ for the original model Φ's node u prediction is bounded as follows:

$$\frac{1}{|\mathcal{K}|+1} \sum_{\substack{\mathcal{G}'_u \in \\ \mathcal{K} \cup \{\mathcal{G}_u\}}} \left\| \Phi(\mathcal{G}'_u) - \Phi(t(\mathcal{G}'_u, \mathcal{E}_u)) \right\|_2 \leq \gamma \left\| \underset{\mathcal{E}_u}{\Delta} \widetilde{A}_u^L \right\|_2 + 2\alpha.$$

Results

Table 1. Performance of node-level explanations for real-world datasets. For this dataset, we use average precision (AP) as an evaluation metric. Blue and Green numbers denote the best and second-best methods, respectively. DnX significantly outperforms the baselines (GNN-, PG-, and PGM-Explainers).

		Bitcoin-Alpha			Bitcoin-OTC		
Model	Explainer	top 3	top 4	top 5	top 3	top 4	top 5
GCN (3-hop)	GNNEx	80.1	74.9	70.9	82.4	79.6	70.6
	PGEx	81.5	78.1	69.5	78.5	74.5	67.4
	PGMEx	67.0	59.8	51.8	63.0	55.2	47.4
	DnX	**95.8**	**91.9**	**87.9**	**94.8**	**91.4**	**86.3**
	FastDnX	**89.8**	**85.2**	**80.2**	**88.0**	**83.0**	**78.8**

Table 2. Performance (average accuracy) of explanation methods for node-level explanations in the synthetic datasets. Blue and Green numbers denote the best and second-best methods, respectively. Overall, FastDnX is the best-performing method for all network architectures (GCN, ARMA, GATED, and GIN) on all datasets but Tree-Cycles and Tree-Grids.

Model	Explainer	BA-House	BA-Community	BA-Grids	Tree-Cycles	Tree-Grids	BA-Bottle
GCN	GNNExplainer	77.5 ± 1.2	64.7 ± 1.0	89.2 ± 2.0	77.2 ± 9.0	71.1 ± 1.0	73.3 ± 3.0
	PGExplainer	95.0 ± 1.1	70.6 ± 2.0	86.2 ± 9.0	**92.4 ± 5.2**	76.7 ± 1.2	98.2 ± 3.0
	PGMExplainer	**97.9 ± 0.9**	92.2 ± 0.2	88.6 ± 0.9	**94.1 ± 0.8**	**86.8 ± 2.0**	97.5 ± 1.5
	DnX	97.7 ± 0.2	**94.6 ± 0.1**	**89.8 ± 0.1**	83.3 ± 0.4	80.2 ± 0.1	**99.6 ± 0.1**
	FastDnX	**99.6** ± NA	**95.4** ± NA	**93.9** ± NA	87.3 ± NA	**85.0** ± NA	**99.8** ± NA
ARMA	GNNExplainer	80.9 ± 1.2	78.5 ± 1.0	87.3 ± 1.3	77.7 ± 1.0	79.3 ± 1.1	84.3 ± 1.3
	PGExplainer	91.4 ± 0.1	72.1 ± 0.1	83.8 ± 1.0	**92.6 ± 2.1**	85.1 ± 0.1	97.0 ± 1.1
	PGMExplainer	**99.3 ± 0.2**	67.5 ± 0.8	86.8 ± 0.3	**95.0 ± 0.2**	**90.6 ± 0.3**	**99.7 ± 0.1**
	DnX	98.1 ± 0.2	**92.7 ± 0.2**	**90.8 ± 0.1**	83.5 ± 0.4	79.6 ± 0.3	96.9 ± 0.2
	FastDnX	**100.0** ± NA	**95.2** ± NA	**94.7** ± NA	87.1 ± NA	**87.7** ± NA	**99.9** ± NA
GATED	GNNExplainer	79.7 ± 1.0	68.8 ± 1.0	**91.4 ± 3.0**	85.2 ± 2.0	73.2 ± 4.0	70.0 ± 2.0
	PGExplainer	96.1 ± 4.1	70.9 ± 3.0	90.7 ± 1.0	**91.7 ± 7.0**	83.7 ± 1.5	**98.7 ± 0.1**
	PGMExplainer	**98.6 ± 0.1**	69.4 ± 0.5	86.8 ± 0.3	**94.1 ± 0.2**	**90.1 ± 0.2**	**98.3 ± 0.2**
	DnX	98.3 ± 0.1	**91.1 ± 0.1**	90.8 ± 0.1	85.0 ± 0.3	82.1 ± 0.2	98.0 ± 0.2
	FastDnX	**99.6** ± NA	**93.5** ± NA	**94.0** ± NA	76.8 ± NA	**86.8** ± NA	98.0 ± NA
GIN	PGMExplainer	60.2 ± 0.2	84.5 ± 0.3	68.4 ± 0.2	**89.3 ± 0.2**	**85.0 ± 0.5**	55.7 ± 0.4
	DnX	**99.0 ± 0.1**	**94.0 ± 0.2**	**91.1 ± 0.1**	**84.1 ± 0.3**	**77.3 ± 0.2**	**95.3 ± 0.2**
	FastDnX	**99.6** ± NA	**94.7** ± NA	**93.9** ± NA	75.2 ± NA	76.5 ± NA	**99.1** ± NA

Time comparison.

To demonstrate the computational efficiency of DnX/FastDnX, Figure 1 shows the time each method takes to explain a single GCN prediction. For a fair comparison, we also take into account the distillation step in DnX/FastDnX.

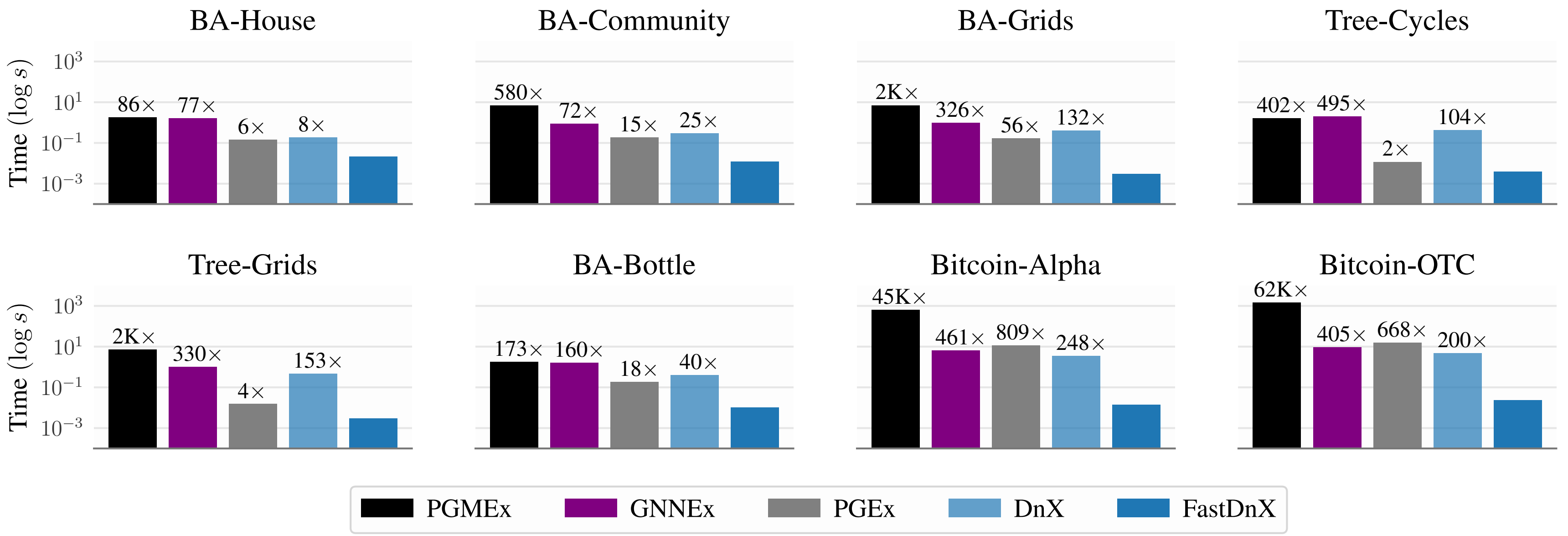

Figure 1. Time comparison. The bar plots show the average time each method takes to explain a prediction from GCN.

Are benchmarks too simple?

Given that DnX/FastDnX often achieve remarkable performance by explaining simple surrogates, a natural questions arises: are these popular benchmarks for GNN explanations too simple? Since these benchmarks rely on model-agnostic ground-truth explanations, we now investigate inductive biases behind these explanations, and show that they can be easily captured.

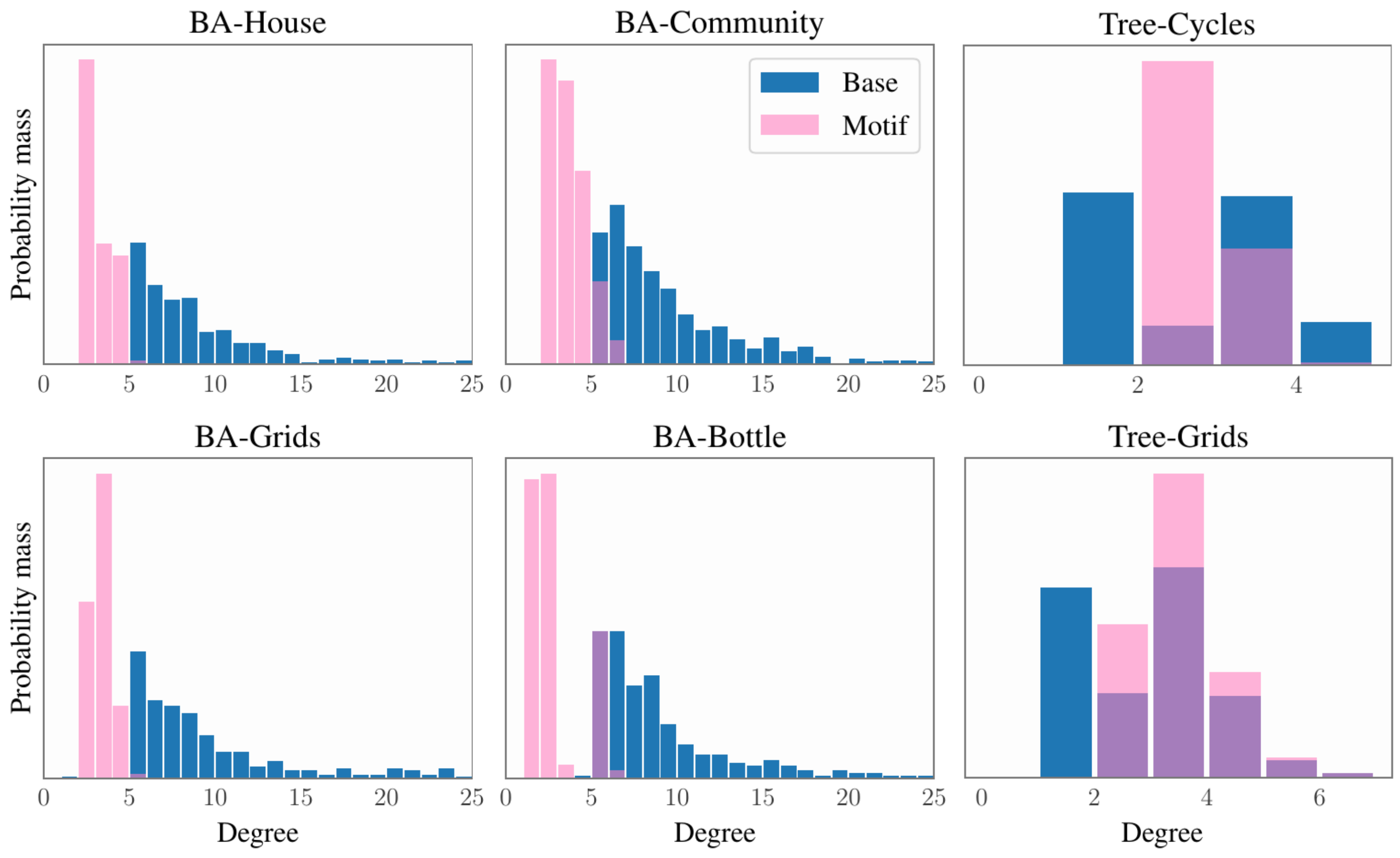

Figure 2. Degree distribution of motif and base nodes.

What you'll also find in the manuscript

- Distillation results;
- Results for edge-level predictions;
- Experiments evaluating the fidelity on real datasets;
- Experiments on additional datasets.