

Report from project SAD2

1 Datasets and Ground Truth

Datasets (simulated trajectories) and their corresponding ground-truth Boolean networks are generated by sweeping the following parameter grid:

Parameter	Values
<i>Network parameters</i>	
n_nodes	{5, 8, 11, 16}
network_seed	{0, 1}
<i>Trajectory parameters</i>	
n_trajectories	{1, 5, 20}
sync transition	{True, False}
trajectory_len	{5, 20, 100}
sampling_frequency	{1, 3}

Table 1: Parameter grid used for generating datasets and ground-truth networks.

Larger networks (e.g., `n_nodes` > 12) significantly increased runtime for both dataset generation and BNFiner inference, so instead of exhaustively testing all sizes we used a representative set spanning the range [5, 16]. In total, the sweep contains $4 \times 2 \times 2 \times 3 \times 3 \times 2 = 288$ parameter combinations, which already requires substantial computational effort during evaluation. In addition, we tracked the `attractor_state_percentage`, which varies between 0 and 1 depending on the trajectory. Overall, the chosen grid allowed us to assess which conditions favor accurate graph-structure inference.

2 Evaluation

- **Reconstruction setup.** We used the simplest BNFiner invocation: `bnf -e input1.txt -n output1.sif -l 3`. The key adjustment was `-l 3`, which limits each Boolean function to at most three parents; this greatly reduces the search space and speeds up reconstruction.
- **Scoring functions.** We evaluated candidate network structures using two BNFiner-recommended scores: *Minimal Description Length (MDL)* and *BDe (Bayesian–Dirichlet equivalence)*.
- **Loss functions (accuracy metrics).** We measured structural prediction error using `edge jaccard distance` and `graph edit distance`: together they capture both *local edge overlap* (exact wiring agreement) and *global topological discrepancy* (how many edits are needed to transform one graph into the other). We chose these metrics because they are widely used in graph-structure evaluation and provide complementary views of reconstruction quality.

- **Final conclusions.** We ran a wide range of parameter configurations, and presenting all plots here would be unreadable, so we selected only the most informative ones. The results varied substantially—from near-perfect reconstruction to almost no correct edges. Overall, the most influential factor was dataset size: more trajectories and longer trajectories consistently improved performance (Figure 3). This matches intuition, since larger datasets provide more information for inference. We also observed a slight advantage for synchronous transitions, which is intuitive because synchronous updates are deterministic and, at each step, constrain the Boolean transition functions of all nodes. In contrast, the choice of evaluation metric (`edge_jaccard_distance` vs. `graph_edit_distance`) did not affect the qualitative conclusions, and we did not observe a consistent difference between the MDL and BDe scoring methods. Finally, we observed a slight negative correlation between network size and reconstruction accuracy (Figure 4), which is expected since larger networks are generally harder to infer accurately.

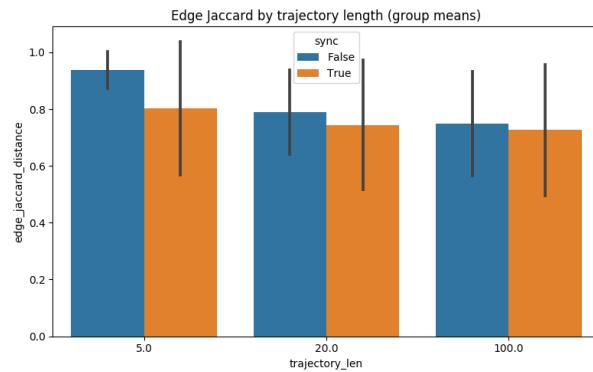


Figure 1: *
Jaccard vs. trajectory length (MDL)

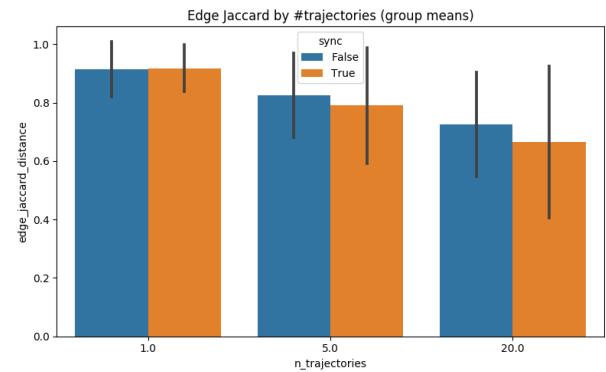


Figure 2: *
Jaccard vs. number of trajectories (MDL)

Figure 3: Edge Jaccard distance under MDL for two dataset axes.

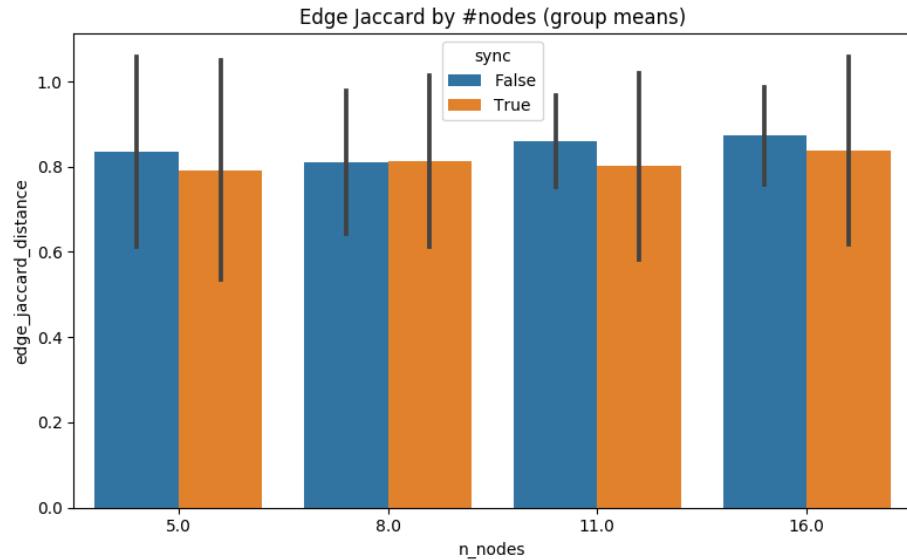


Figure 4: Edge Jaccard distance under MDL as a function of network size (n_nodes).

3 Challenges encountered

- **Legacy environment constraints.** One of the biggest challenges was setting up a fully compatible Python 2.7 environment (libraries, type checking, and tooling), since BNFinder does not support Python 3+. This turned out to be a valuable lesson in reproducibility: some of us managed the setup via `pyenv`, while others had to rely on Docker due to limited compatibility of their local machines with that early of a Python version.
- **Computational cost of the sweep.** To meaningfully assess how inference quality depends on key parameters (e.g., transition mode, network size etc.), we needed to run evaluations for a few hours, which limited how many additional combinations we could explore. To reduce runtime, we parallelized runs where possible and used BNFinder’s `-1` option to cap the maximum number of parents per node, substantially decreasing computational burden.