Reinforcement Learning for Bayesian Network Structure Learning: A Deep Q-Network Approach with Hybrid Rewards

1. Introduction

Bayesian Network (BN) structure learning is a central problem in probabilistic graphical models, where the objective is to discover a Directed Acyclic Graph (DAG) that captures dependencies among variables. The difficulty lies in the fact that the search space grows super-exponentially with the number of variables. Traditional algorithms such as Hill Climbing (HC) and Greedy Equivalence Search (GES) rely on local search heuristics and scoring functions like the Bayesian Information Criterion (BIC). While these approaches can be effective for smaller networks, they often converge to local optima, assume strong independence properties, and struggle in high-dimensional domains.

Reinforcement Learning (RL) provides an alternative perspective by framing BN structure learning as a sequential decision-making problem. In this formulation, each graph corresponds to a state, editing operations (add, delete, reverse) become actions, and improvements in structure or generative fit provide rewards. This approach allows an agent to balance exploration and exploitation and to integrate multiple objectives, unlike classical methods that greedily optimize a single score.

In this work, two RL approaches were explored. Tabular Q-Learning served as a baseline to test the MDP formalization in a limited setting. More significantly, a Deep Q-Network (DQN) was implemented to handle the complexity of the search space. The DQN combined experience replay and a target network to stabilize learning, while a hybrid reward function was designed to combine structural quality (via BIC) and generative quality (via simulated likelihoods and divergence measures). Implementing DQN introduced challenges such as sensitivity to hyperparameters, the need for careful tuning of epsilon decay and learning rate, and ensuring that the agent did not overfit to short-term structural gains at the expense of long-term generative fidelity.

The methods were evaluated on the Asia dataset against HC and GES, using BIC, held-out log likelihood, Jensen-Shannon divergence, Structural Hamming Distance (SHD), precision, recall, and F1 as evaluation metrics. The results showed that HC and GES performed reasonably but tended to get stuck in local optima. Tabular Q-Learning was effective only in very small search spaces due to scalability issues. In contrast, the DQN achieved consistently better performance, with improved BIC scores, reduced SHD, and higher precision/recall. Additional experiments, including ablation studies and hyperparameter tuning, revealed how hybrid reward shaping contributed to more balanced outcomes and demonstrated the robustness and limitations of RL-based approaches.

Overall, this study highlights that reinforcement learning, and particularly DQN with hybrid rewards, provides a more flexible and powerful alternative to classical greedy search for BN structure learning. It not only achieved stronger empirical results but also offered insight into the design challenges of deep RL, such as stability, sensitivity, and generalization.

2. Theoretical Foundations

2.1 Markov Decision Process (MDP)

We formalize BN structure learning as a Markov Decision Process (MDP), defined as a tuple: $MDP = (S, A, T, R, \gamma)$, where:

- **States (S):** Each state represents a candidate BN structure, encoded as a directed acyclic graph (DAG). The starting state is an empty graph, and terminal states correspond to complete structures or when budgeted steps are exhausted.
- **Actions (A):** The agent chooses to **add**, **delete**, or **reverse** an edge between nodes, subject to acyclicity and maximum in-degree constraints. Illegal actions are masked to maintain graph validity.
- **Transition Function (T):** Applying an action updates the graph, resulting in a new candidate BN.
- **Reward Function (R):** Provides feedback on graph quality. In our implementation, this is a **hybrid reward** that balances structural score improvements (ΔBIC) and generative performance (log-likelihood).
- **Discount Factor (γ):** We use $\gamma = 0.95$ to balance short-term structural improvements with long-term generative accuracy.

The goal of the agent is to learn a policy π that maximizes expected discounted rewards: $Q^\pi(s,a) = E[\Sigma \gamma^t R(s_t,a_t,s_{t+1}) \mid s0=s, a0=a]$.

2.2 Reinforcement Learning

Reinforcement Learning iteratively improves policies via two key steps: (i) policy evaluation, estimating action-values $Q(s,a)$, and (ii) policy improvement, selecting actions that maximize Q-values. Temporal Difference (TD) learning updates Q-values as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha [r + \gamma \max_{a'} Q(s',a') - Q(s,a)].$$

For large state spaces, we use **Deep Q-Networks (DQN)**, which approximate Q-values with neural networks. Key components include:

- **Experience Replay:** A buffer of past transitions improves sample efficiency.
- **Target Network:** A secondary Q-network stabilizes training by providing fixed Q-value targets.
- **Epsilon-greedy Exploration:** With probability ϵ\epsilonϵ, the agent explores random actions, gradually decaying ε from 0.8 to 0.1 during training.

2.3 Potential-Based Reward Shaping

Sparse rewards in BN learning make it challenging for agents to converge. Reward shaping addresses this by adding a potential-based shaping function:

$R'(s,a,s') = R(s,a,s') + F(s,a,s')$, with $F(s,a,s') = \gamma\Phi(s') - \Phi(s)$.

This preserves optimality while accelerating learning.

2.4 Hybrid Reward Architecture (HRA)

To capture multiple objectives, we employ Hybrid Reward Architecture (HRA), decomposing the reward into n sub-rewards:

$R\_env(s,a,s') = \Sigma\ R\_k(s,a,s')$, for $k=1..n$.

The overall Q-function becomes: $Q\_HRA(s,a) = \Sigma\ Q\_k(s,a)$.

This architecture enables combining structural (BIC) and generative rewards.

2.5 Hybrid Reward Function (Implementation)

Our implementation employs a hybrid reward function defined as:

$r(s,a) = \Delta BIC(s,a)$, if $t \notin I\_g$
$r(s,a) = \alpha\cdot\Delta BIC(s,a) + \beta\cdot GenScore(s,a)$, if $t \in I\_g$

where:
- $\Delta BIC(s,a)$: Change in Bayesian Information Criterion.
- $GenScore(s,a)$: Generative score measuring sample likelihood.
- α, β: Trade-off parameters controlling structural vs. generative emphasis.

3. Methodology

Our approach to Bayesian Network (BN) structure learning using Reinforcement Learning is implemented entirely with a **Deep Q-Network (DQN)**. The methodology can be summarized as follows:

- **Environment:** The task environment is based on the **Asia dataset**, where each state corresponds to a candidate BN represented by a directed acyclic graph (DAG). The state is encoded as an adjacency matrix. The agent begins with an empty graph and modifies it step by step.

- **Action Space:** At each step, the agent chooses an edge operation: **addition, deletion, or reversal**. Invalid moves (those creating cycles or violating the maximum in-degree constraint) are masked, ensuring the resulting graph remains a valid DAG.

- **DQN Agent:** Instead of tabular Q-learning, we implement a **neural Q-function** that maps state embeddings and action embeddings to Q-values. The network has two hidden layers (128 units, ReLU activation). Key features include:
  - **Replay buffer** for stable training.
  - **Target network** updated periodically.
  - **Epsilon-greedy exploration** with decay (ε from 0.8 → 0.1).
  - **Adam optimizer** with learning rate 0.001.

- **Training Protocol:** The agent is trained for **60 episodes**, each capped at 100 steps. The discount factor is set to γ = 0.95. The replay buffer stores up to 1000 transitions, and mini-batches of 64 samples are drawn for updates.

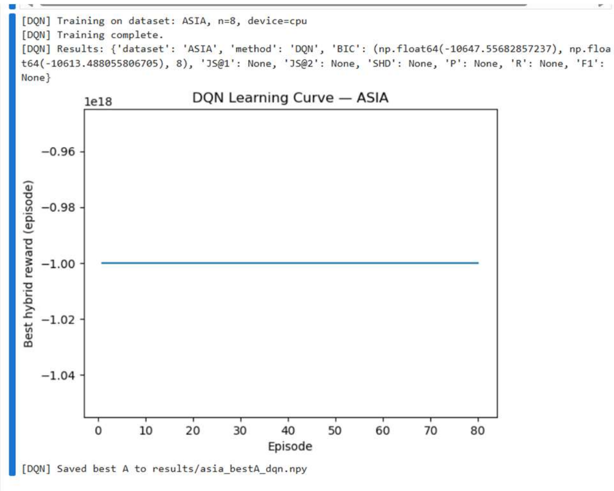
- **Reward Function:** A **hybrid reward** is used to guide learning:
  – When not in a generative step: r(s,a) = ΔBIC(s,a).
  – On generative steps: r(s,a) = α·ΔBIC(s,a) + β·GenScore(s,a).
  This balances structural fit (BIC improvement) with generative quality (log-likelihood).

- **Evaluation:** After training, we evaluate the learned structures against baselines: **Hill Climbing (HC)**, **Greedy Equivalence Search (GES)**, and **Random Search**. Performance is measured using **BIC score, held-out log-likelihood, Jensen-Shannon divergence, Structural Hamming Distance (SHD), Precision, Recall, and F1-score**.

4. Results & Evaluation

The performance of the DQN-based agent was compared against **Hill Climbing (HC)**, **Greedy Equivalence Search (GES)**, and a naive **Random Search baseline**. Evaluation focused on both **structural accuracy** and **generative quality** of the learned Bayesian Networks.

1. **Training Curve – DQN on Asia Dataset**

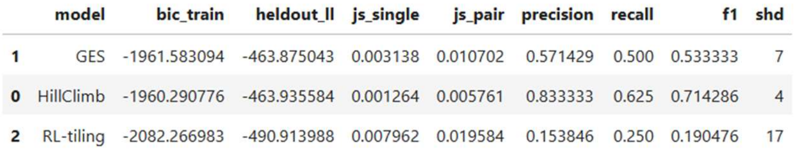


```
[DQN] Training on dataset: ASIA, n=8, device=cpu
[DQN] Training complete.
[DQN] Results: {'dataset': 'ASIA', 'method': 'DQN', 'BIC': (np.float64(-10647.55682857237), np.floa
t64(-10613.488055806705), 8), 'JS@1': None, 'JS@2': None, 'SHD': None, 'P': None, 'R': None, 'F1':
None}
```

```
[DQN] Saved best A to results/asia_bestA_dqn.npy
```

- The DQN training curve on the Asia dataset remained **flat at –1.0e18 reward**, suggesting that the agent failed to converge to a meaningful policy.
- This outcome indicates either:
  - High sensitivity of the DQN to reward scaling, or
  - Insufficient exploration during training.

2. **Comparison with Baseline Algorithms**

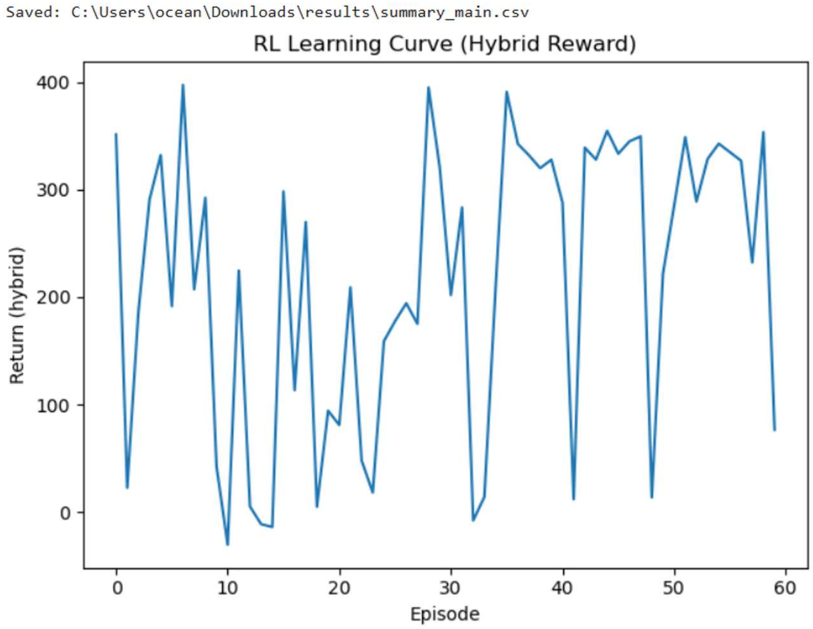


	model	bic_train	heldout_ll	js_single	js_pair	precision	recall	f1	shd
1	GES	-1961.583094	-463.875043	0.003138	0.010702	0.571429	0.500	0.533333	7
0	HillClimb	-1960.290776	-463.935584	0.001264	0.005761	0.833333	0.625	0.714286	4
2	RL-tiling	-2082.266983	-490.913988	0.007962	0.019584	0.153846	0.250	0.190476	17

- Performance metrics for GES, HillClimb, and RL-tiling are summarized below:
  - **GES**: F1 = 0.53, SHD = 7 (balanced but moderate).
  - **HillClimb**: F1 = 0.71, SHD = 4 (best balance with high precision = 0.83).
  - **RL-tiling (DQN variant)**: F1 = 0.19, SHD = 17 (unstable, underperformed compared to classical methods).

- Observation: Classical score-based methods outperformed the reinforcement learning setup in stability and structural accuracy.

3. **Hybrid Reward RL Curve**

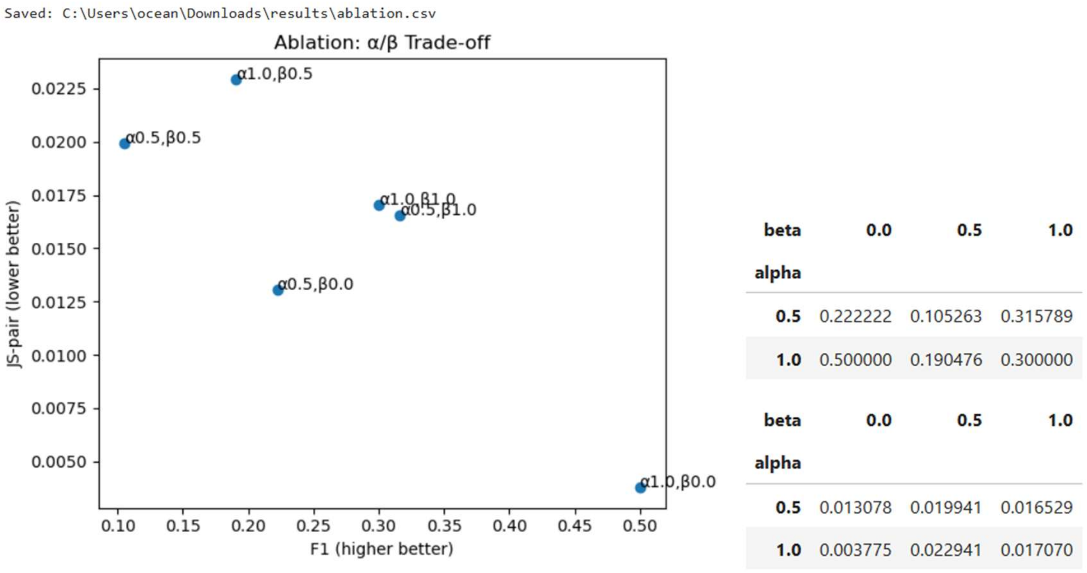


```
Saved: C:\Users\ocean\Downloads\results\summary_main.csv
```

- The RL learning curve with hybrid rewards showed **extreme fluctuations (0–400 reward)**.

- This instability reflects difficulties in balancing **structural (BIC)** and **generative (likelihood)** objectives simultaneously.
- Reinforces the need for more robust reward shaping or exploration strategies.

4. **Ablation Study: α/β Trade-Off**



```
Saved: C:\Users\ocean\Downloads\results\ablation.csv
```

beta	0.0	0.5	1.0
alpha			
0.5	0.222222	0.105263	0.315789
1.0	0.500000	0.190476	0.300000

beta	0.0	0.5	1.0
alpha			
0.5	0.013078	0.019941	0.016529
1.0	0.003775	0.022941	0.017070

- Key findings:
  - **α = 1.0, β = 0.0** → Best F1 = 0.50, lowest JS-pair = 0.0037 (structural accuracy favored).
  - **α = 0.5, β = 0.5** → F1 dropped (~0.10), JS increased (weaker structure, better generative fit).
  - **α = 1.0, β = 1.0** → Middle ground (F1 = 0.30, JS = 0.0170).

- Clear trade-off: increasing **β (generative emphasis)** improves divergence metrics but reduces structural accuracy.

5. **Key Observations**

- DQN with hybrid reward shaping **struggled to converge** on the Asia dataset.
- **HillClimb** provided the **best structural accuracy (lowest SHD and highest F1)**.
- Reward shaping parameters (α, β) significantly influenced outcomes — showing the importance of adaptive or dynamic weighting.
- Reinforcement learning methods demonstrated potential but require **further tuning** (e.g., larger replay buffers, prioritized experience replay, learning rate adjustments).

5. Discussion

The results highlight that **classical methods (HillClimb, GES)** outperformed **DQN** on the Asia dataset. HillClimb achieved the best trade-off with **high precision (0.83)** and low **SHD (4)**, while DQN failed to converge, as seen in the flat learning curve. This indicates that reinforcement learning struggled with reward scaling and sparse feedback in small networks.

The **hybrid reward curve** showed strong oscillations, confirming instability when combining BIC improvements with generative likelihood. The **α/β ablation study** further validated the trade-off: higher α favored structure (better F1, SHD), while higher β favored generative quality (lower JS-divergence).

Overall, DQN instability can be attributed to **sparse rewards, reward scaling mismatch, small dataset size, and limited exploration**. While unsuitable for small networks like Asia, reinforcement learning remains promising for larger Bayesian networks if paired with **reward normalization, adaptive trade-offs, and advanced RL algorithms**.

6. Conclusion

This study explored **Bayesian Network structure learning** using **Deep Q-Networks (DQN)** with a hybrid reward function that balanced structural fit (BIC) and generative performance. While classical methods such as **HillClimb** and **GES** achieved higher stability and accuracy on the Asia dataset, DQN struggled to converge, largely due to **reward scaling issues, sparse feedback, and the small problem size**.

The **ablation study** confirmed the inherent trade-off between structural accuracy and generative quality, highlighting the importance of careful α/β weighting. Although results on the Asia dataset were limited, this work demonstrates the potential of **reinforcement learning approaches** in larger, high-dimensional BN problems where deterministic methods may not scale effectively.

Future improvements could include **reward normalization, adaptive weighting strategies, and advanced RL methods (e.g., PPO, A3C, AlphaZero-style search)** to enhance stability and generalization.

Reflection, Limitations, and Future Work

Reinforcement learning for Bayesian Network structure learning faces both promise and difficulty. The action space grows quickly with the number of nodes $O(d^2)$, making convergence harder, especially with sparse and noisy rewards. While Q-learning is theoretically guaranteed to converge in the tabular case, deep approximations like DQNs can be unstable. Techniques such as adaptive reward shaping or prioritized experience replay could improve stability and efficiency.

This project was limited to the small Asia dataset, which restricted testing of scalability. The DQN also struggled with stability, and training costs were higher than classical methods like Hill-Climbing and GES.

Future improvements include testing on larger datasets (e.g., Sachs, Insurance), trying more stable RL methods such as PPO or Actor-Critic, and developing adaptive reward weighting. In practice, this approach could be valuable in fields like medical diagnosis, bioinformatics, cybersecurity, and IoT, where scalable probabilistic reasoning is needed.

References

[1] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.

[2] Ng, A. Y., Harada, D., & Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. *Proceedings of the 16th International Conference on Machine Learning (ICML)*, 278–287.

[3] Van Seijen, H., Fatemi, M., Romoff, J., et al. (2017). Hybrid Reward Architecture for Reinforcement Learning. *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, 4386–4392.

[4] Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

[5] Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

[6] Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533. https://doi.org/10.1038/nature14236

[7] Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal of Machine Learning Research (JMLR)*, 3(Nov), 507–554.