

SIT320 Advanced Algorithms

High Distinction Task, Option A: RLiG, Reinforcement Learning for Bayesian Network Structure and Generation

Samridh Limbu

All the code is in `task-11.ipynb` and its 55 tests pass. Every table and figure below is written by that notebook's last cell, which is the repro entry point, and every number in the prose is copied off its output.

Two conventions, because both go the other way in parts of the literature. BIC here is $LL(\hat{\theta}, G) - \frac{\log N}{2}|\theta|$, so **higher is better**, in nats. Structural Hamming distance is counted on the CPDAG, which is how Tsamardinos et al. define it [9], so a reversal the data cannot distinguish costs nothing. The DAG version sits beside it as SHD_{DAG} because a quick implementation computes that one, and the two differ a lot.

1 Problem and Formulation

Bayesian network structure learning is normally a greedy search over DAGs scored by something like BIC, and the score is the only thing the search looks at. RLiG changes two things [1]. It makes the search a sequential decision problem, so a move can be worth taking even when it lowers the score right now. And it adds a second term asking how good the learned network is at the job it usually gets built for, which is generating data. That second term is expensive. So RLiG only pays for it at chosen points and falls back on the cheap structural score everywhere else, those points are the tiles.

1.1 The decision process

Given N rows of discrete data over d variables, variable i taking r_i values:

State $s_t = (G_t, t \bmod L, \phi_t)$, where G_t is the current DAG as a boolean adjacency matrix, L is the tile length, and ϕ_t holds the BIC per row, the edge count, t/T , the mean and maximum in-degree, the stall counter, the last measured generative score, a flag for whether this step is a generative one and the share of the generative budget already spent. The DQN sees those ten numbers and a constant, and nothing else. Tabular keys on the graph and the tile position only, and linear sees the state through the action features of Section 2.3. The tile position has to reach all three, or the agent cant tell a cheap step from an expensive one.

Actions $a = (\text{op}, i, j)$ with op one of add, delete or reverse, so $3d^2$ in the raw encoding. Feasible means the result stays acyclic and every in-degree stays at or below k . **Transitions** are deterministic. **Termination** is after T edits or L_{stall} steps with no BIC improvement.

Reward. Write $BIC_N(G) = BIC(G)/N$, let $\text{Gen}(G) \in [0, 1]$ be the generation similarity of Section 2.1, and let the hybrid score be $H(G) = \alpha BIC_N(G) + \beta \text{Gen}(G)$. At a non-generative step Gen is not measured, so $r_t = \alpha(BIC_N(G_{t+1}) - BIC_N(G_t))$. At a generative step $r_t = \alpha \Delta BIC_N + \beta(\text{Gen}(G_{t+1}) - g_{\text{last}})$, where g_{last} is the score from the previous generative state, because $\text{Gen}(G_t)$ was never measured. On the first one of an episode there is no previous score, so that reward carries no generative part. The task sheet also allows the level form $\alpha \Delta BIC + \beta \text{Gen}(G')$, which is implemented and ablated in Section 4.5.

1.2 Why the difference form is the default

With $\gamma = 1$ the difference reward telescopes. The structural half collapses to $\alpha(BIC_N(G_T) - BIC_N(G_0))$ across the whole episode, and the generative half collapses between the first and the last graph that was measured. So maximising return is close to maximising the hybrid score of the final graph, and that is why RL is here. An edit that lowers H is not forbidden, it only has to be paid back before the episode ends, and a greedy hill climber cant do that at all. `test_reward_telescopes_to_the_hybrid_gain` checks that identity to six decimals. Two things stop it being the clean $H(G_T) - H(G_0)$. $\text{Gen}(G_0)$ is never evaluated, so the baseline is the first measured graph and the agents own first edit chooses it. And with tiles the generative half only telescopes between consecutive generative states.

1.3 Tiling, and what it is worth

Step t is generative when $t \bmod L \in I_g$. With $L = 5$, $I_g = \{0\}$ and $T = 40$ an episode schedules eight simulate-and-compare calls instead of forty. Table 1 measures what that saves per `env.step`, which is the cost that matters.

2 Method

2.1 The two halves of the objective

BIC decomposes over families, which is the only reason any of this runs:

$$BIC(G) = \sum_{i=1}^d \left[\sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \frac{N_{ijk}}{N_{ij}} - \frac{\log N}{2} q_i (r_i - 1) \right].$$

network	d	pairs	cheap step (ms)	generative step (ms)	ratio
synthetic	10	45	0.06	0.71	12.0×
asia	8	28	0.05	0.52	10.5×
sachs	11	55	0.08	0.81	10.5×
child	20	190	0.16	2.02	12.9×
insurance	27	200	0.29	2.44	8.3×
alarm	37	200	0.77	3.51	4.5×

Table 1: Milliseconds per environment step at $N_s = 1000$, which is the cheapest setting the tiling argument has to survive. The runs in Section 4 pay five times that many simulated rows. A cheap step is a cached BIC delta per legal action, a generative step adds a CPT fit, N_s simulated rows and the marginal comparison. At $N_s = 5000$ the ratio reaches 36× on Child and is 25 to 34× on the three smaller ones. It falls away at the wide end, because scanning every legal action costs more when there are more of them, so tiling pays best at or below about twenty variables and at large N_s .

The penalty is Schwarz’s $\frac{k}{2} \log n$ [2], the decomposable discrete form with this notation is de Campos [3]. An edit touches one family, two for a reversal, so a delta costs at most four local scores and all are cached. The likelihood term uses plain MLE counts, which the BIC derivation assumes. Dirichlet smoothing [5, 4] stays out of the score and goes into the fitted CPTs, because unsmoothed the held-out likelihood goes to minus infinity the first time a test row hits a parent configuration training never showed.

Gen(G) fits every CPT on the training split with a Dirichlet $\alpha = 1$ prior, draws N_s rows by ancestral sampling, compares one-way and two-way marginals against a reference split using Jensen-Shannon divergence in bits [22], and returns $1 - (\frac{1}{2}\text{JS}_1 + \frac{1}{2}\text{JS}_2)$. A JS divergence in bits is bounded by 1, so $\text{Gen} \in [0, 1]$ whatever d is and β keeps meaning the same thing. Two-way marginals are all $\binom{d}{2}$ pairs when there are few and a fixed seeded subset of 200 otherwise, fixed rather than resampled so two scores compare the same quantity. The reference split is held out, because scoring a simulation against the rows its own CPTs were fitted to rewards a graph for reproducing noise.

2.2 Calibrating β , which is not optional

The two halves are about two orders of magnitude apart, which I did not expect. On ASIA at 500 rows the empty graph against hill climbing’s graph moves BIC_N by about 0.65 and Gen by about 0.0062, so at $\alpha = \beta = 1$ the generative term contributes nothing and a sweep over $\beta \in \{0, 1, 2\}$ would measure nothing while looking like it had. For each dataset I take

$$\beta^* = \frac{\text{BIC}_N(G_{\text{hc}}) - \text{BIC}_N(\emptyset)}{\text{Gen}(G_{\text{hc}}) - \text{Gen}(\emptyset)},$$

the β at which one unit of movement in each half is worth the same, and quote every β as a multiple ρ of it. $\rho = 1$ is therefore balanced by construction rather than by tuning, and it was fixed before I looked at any result. The main table also carries a $\rho = 0$ row per learner, added after the sweep of Section 4.4 because $\rho = 0$ does better over most of the grid, Child being the exception, which is reporting the negative result rather than tuning on it. It costs two generative evaluations and grows with d , about 104 on ASIA and about 1837 on ALARM, which is why a raw β cannot transfer across networks but ρ can.

2.3 The dual policy and the three agents

The policy is ε -greedy [16, 18] on a learned Q . In a non-generative state the reward it trained on is the structural score alone, in a generative state its the full hybrid, and the tile position is in the state so the agent can tell which regime applies.

Tabular keys Q on the graph bytes and the tile position [17], so only visited states cost anything. Its exact, and it is the honest way to show the ceiling. **Linear** is $Q(s, a) = w^\top \phi(s, a)$ over fifteen features. Anything constant across the actions of one state cannot change an argmax, so state variables appear only multiplied by something that does vary, either the normalised BIC delta or the operator. So one weight vector can learn “trust the BIC delta less on a generative step”. **The DQN**, HD extension 1, is a network over the state encoding concatenated with learned embeddings of i, j and the operator plus five scalar action features, with a replay buffer and target network [19]. Because the action is embedded rather than indexed, one architecture covers every (i, j, op) at every width. Infeasible actions are masked before the argmax and before the bootstrap max [20], so the agent never has to learn acyclicity, only which legal edits are worth taking. Storing every next-state action matrix in the buffer is not affordable at $d = 37$, so the target maxes over a 24 action shortlist, half best by BIC delta and half random. That is an approximation. Section 5 counts it against the result.

The BIC delta is an action feature in both learners on purpose, so RLIG only has to learn when following the score is the wrong move. Section 4.5 zeroes it.

2.4 Exploration, and learned tiling

Uniform ε -greedy over the legal actions does not work here. A sparse ALARM graph offers roughly thirteen hundred legal edits, so a uniform draw almost never lands on a useful one. RLIG instead explores with a Boltzmann draw over the BIC delta at temperature 0.3. The DQN draws over the delta normalised by the

largest one in that state, tabular and linear draw over the raw delta in nats, which at this temperature puts nearly all the mass on the top action or two. Narrower than the name sounds, for those two.

The fixed pattern I_g is a guess. HD extension 2 replaces it with a decision. A second head $Q_{\text{gen}}(s, \text{spend})$ chooses at each step whether to spend one of the episode's B generative evaluations, masked off once the budget is gone. Both heads bootstrap on the same return, which makes this a factored action rather than a separate problem, and they never coordinate explicitly. The comparison is at matched budget: a fixed tile with $L = 5$, $I_g = \{0\}$ schedules $\lfloor T/5 \rfloor$ per episode, so the learned tiler gets a cap of exactly $\lfloor T/5 \rfloor$. It is free to spend fewer and it does. Anything it gains has to come from spending them at better moments or from not spending them at all.

```

G <- init()                # empty, random, or a hill-climb warm start
for t = 0 .. T-1:
  A <- feasible_edits(G)    # cycle and in-degree masks off one closure
  D <- bic_deltas(G, A)    # cached local scores, two for a reversal
  a <- policy(s_t, A, D)    # argmax on Q, Boltzmann on D when exploring
  G' <- apply(a, G)
  if (t mod L) in I_g:      # or the meta-head says spend, and budget remains
    r <- alpha*dBIC_N + beta*(Gen(G')-g_last) # N_s rows from G', compare
    remember (H(G'), G')    # only measured graphs can be the answer
  else:
    r <- alpha * (BIC_N(G') - BIC_N(G))
  update(s_t, a, r, s_{t+1}); G <- G'
return the graph with the best measured H

```

Figure 1: RLIG, one episode. The last line matters: only graphs seen at a generative state are candidates, so with $L = 5$ the answer comes from one in five of the graphs visited. A run with no generative states falls back to the best graph by BIC, and both are tracked.

2.5 HD extension 3, what bounded noise can do

Gen is estimated from simulated rows, so the agent sees $\hat{g} = g + \xi$ and not g . Assume only that $|\xi| \leq \varepsilon$, and note that the per-graph cache measures each graph once a run, so its ξ is fixed.

The return. The telescoping of Section 1.2 leaves only the first and the last measured graph in the generative part of the return, $\beta(\hat{g}_m - \hat{g}_1)$ over m generative steps. So $|\hat{R} - R| \leq 2\beta\varepsilon$, or 0 when $m \leq 1$. The level form adds $\beta\hat{g}_k$ at every generative step, so its error reaches $m\beta\varepsilon$ under a constant shift. At $L = 5$ and $T = 40$, $m = 8$, so four times worse than the difference form. Two tests check both on the real environment with made-up noise.

The policy and the returned graph. If every return is off by at most c , the policy that is best on noisy returns loses at most $2c$ against the truly best one, so $J(\pi^*) - J(\hat{\pi}) \leq 4\beta\varepsilon$, against $2m\beta\varepsilon$ for the level form. That's a bound on the optimum of the noisy objective and says nothing about whether training gets there. The usual tabular convergence argument needs a reward that depends only on the state and the action, and g_{last} is not in my tabular key, so I have no convergence result. The same argument bounds what gets handed back. If G_0 is the measured graph with the best true H , the returned $\hat{G}$ has $H(G_0) - H(\hat{G}) \leq 2\beta\varepsilon$. Its recorded score is off by $\beta\xi(\hat{G}) \in [-\beta\varepsilon, \beta\varepsilon]$, which cannot be negative on average for a fixed set of graphs with independent mean-zero noise, because a maximum favours the graphs whose noise came out high. Section 4.6 measures that optimism.

How big ε is. The marginal score has no useful worst-case bound, so Section 4.6 measures its noise instead. The discriminator score of Section 2.6 has one for part of its noise. Given the trained classifier $f \in [0, 1]$ and the real rows it is tested on, the score moves no more than the mean of f over the m_{te} held-out simulated rows does, and those rows are independent draws from the graph. Hoeffding's Theorem 1 [24] and a union bound over the K graphs a run measures put that part of every score within $\varepsilon = \sqrt{\ln(2K/\delta)/(2m_{\text{te}})}$ with probability $1 - \delta$, so $m_{\text{te}} \geq 2\beta^2 \ln(2K/\delta)/\Delta^2$ is enough to keep its share of the gap to $H(G_0)$ under Δ . At $m_{\text{te}} = 1000$, $K = 2000$ and $\delta = 0.05$ the run prints $\varepsilon = 0.0751$, and 0.01 would take 56449 rows. On ASIA $2\beta\varepsilon$ is then 0.461, most of the 0.65 of BIC_N between hill climbing's graph and the empty one, so as a guarantee it says little at this size. It also leaves out the noise from training the classifier and from which real rows it is tested on.

2.6 HD extension 4, a learned discriminator score

MMD-GAN [26] swaps a fixed kernel for one learned adversarially, so the test looks wherever two samples differ most. I take the simplest form of that idea, a classifier two-sample test [25]. If a classifier can't tell simulated rows from real ones on rows it never trained on, the graph generates well. For a graph, Gen_{disc} fits the CPTs as before, simulates $m = \min(N_s, 2000)$ rows, one-hot encodes them with m reference rows and splits both in half. A network with one hidden layer of 32 units trains on one half for 150 full-batch Adam steps at learning rate 0.03. With f its probability that a row is real, the other half gives $\text{Gen}_{\text{disc}}(G) = 1 - |\bar{f}_{\text{real}} - \bar{f}_{\text{sim}}|$. Lopez-Paz and Oquab score held-out accuracy. On a balanced test set, crediting each row with f instead of the hard call $f > \frac{1}{2}$ makes that accuracy $\frac{1}{2} + \frac{1}{2}(\bar{f}_{\text{real}} - \bar{f}_{\text{sim}})$, so this is their statistic with soft calls, which their Remark 2 says the argument extends to. The hard version is in the code too. I picked the soft one on ASIA and then checked it on the other two networks. It takes the marginal scores slot and cache and gets its own β^* , about

3 on ASIA against 104, because its range between hill climbing’s graph and the empty one is 0.21 where the marginal score’s is 0.0062.

3 Experimental Setup

3.1 Data, baselines and metrics

Five discrete benchmarks from the bnlearn Bayesian Network Repository [11], plus one synthetic network with known ground truth. Nothing here is real-world data. The repository publishes node, arc and parameter counts, and my parser reproduces all fifteen of those numbers exactly. That’s `test_counts_match_the_repository`.

Reinforcement learning over DAGs is not new, and neither is searching from the empty graph on a score difference. What my search across arXiv, DBLP, OpenAlex and the general web did not turn up was anything putting *simulation fidelity* in the reward. So the hybrid objective is the part I treat as new, modestly, since a search that finds nothing is weak evidence. Paying for an expensive reward only sometimes is studied [23], though not in this setting.

network	d	arcs	max in-deg	β^*	GES SHD
synthetic	10	12	3	166	4
asia	8	8	2	104	5
sachs	11	17	3	145	13
child	20	25	2	523	13
insurance	27	52	3	1016	36
alarm	37	46	4	1837	31

Table 2: The five repository networks, ASIA [12], Sachs [15], Child, Insurance [14] and ALARM [13], plus the synthetic one. β^* is the calibrated scale of Section 2.2. It grows with d because BIC_N sums over variables while Gen is an average bounded by 1, the synthetic network sitting out of order.

Data is drawn by ancestral sampling and split three ways: 500 rows to fit the CPTs and the BIC, 2000 rows as the reference the generative score compares simulations against, and 5000 rows held out for the final metrics only. The synthetic network is 10 variables, 12 arcs, ternary, with Dirichlet(0.6) conditionals so the dependencies are findable at this sample size. 500 training rows is deliberate, at 5000 BIC is close to consistent on these networks and a second objective has little left to add.

ALARM has a node with four parents, so at $k = 3$ the true ALARM graph is outside the search space of every capped method, which is hill climbing, tabu, GES and all of RLIG, and a nonzero SHD is forced for those. PC is not capped, so it does not inherit that excuse.

Baselines are hill climbing on BIC with add, delete and reverse, the same with a tabu list of length ten and a fifteen step stall limit, GES, and PC with a G^2 test at $\alpha = 0.01$ and conditioning sets up to size two [10]. All four are mine. Hill climbing, tabu and GES share one scorer and cache, every agent run builds its own. Structure is reported as SHD on the CPDAG [9], SHD on the DAG, and edge precision, recall and F1 with an edge correct only when its orientation is right. Generation is held-out log-likelihood per row, mean JS divergence over one-way and two-way marginals, an unbiased MMD² with a Hamming kernel [21] which `task-11.ipynb` computes but no table below reports, and two posterior predictive checks.

Everything is numpy, scipy, torch and matplotlib. No structure learning library, because the searchers are the thing being compared. GES is the one most likely to be wrong if written from memory, so it follows Chickering’s definitions, validity conditions and closed form score changes over completed PDAGs [6, 7], completed by Meek’s rules R1 to R3 [8]. Two deviations are mine. I enumerate **T** and **H** only up to size three where Chickering quantifies over every subset, and my delete refuses an operator whose scored family would exceed the in-degree cap, which is stricter than his validity condition. No backward deletion was ever accepted on these runs, so the second changed no answer here, and whether the size three cap left a better forward operator behind is not something I tested. The notebook has 55 tests and they run in 2.4 seconds. They include the action masks against applying every edit for real over every distinct ordered pair and operator, and the CPDAG code against every DAG on three, four and five nodes, 29281 of them at five in 8782 classes.

3.2 Hyper-parameters

$k = 3$, $\alpha = 1$, $\rho = 1$ so $\beta = \beta^*$, $L = 5$ with $I_g = \{0\}$, $N_s = 5000$, $\gamma = 1$, Dirichlet smoothing 1.0, ε decaying linearly from 1.0 to 0.05, Boltzmann exploration at temperature 0.3, $L_{\text{stall}} = 15$, and an edit budget and episode count that scale with width, $T = 40$ and 250 episodes on the synthetic network and ASIA, $T = 110$ and 120 on ALARM. The rest are in the YAML block in cell 1. `config_from_yaml` parses it into a `Config`, but the grid builds its own through `base_config` and cell 13 sets N_s and the seeds, so the YAML still reads 1000 and three seeds. The DQN sizes, embedding 12, hidden 96, batch 32, buffer 4000, a target copy every 200 steps and a 24 action shortlist, are `DqnRLIG` defaults. Nothing was tuned. The main, ρ , tiling, learning curve and extension 4 reward runs are over ten seeds, the ablation and sample size runs over two, and the baselines are deterministic given the data so they run once. One dataset is drawn per network at a fixed seed and every method sees that draw, so a seed varies the agent and not the data. A printed spread is one standard deviation, and BIC, simulation counts and seconds are means on their own.

network	method	SHD	SHD _{DAG}	prec	rec	F1	BIC	held-out LL	$\overline{JS}_1$	$\overline{JS}_2$
synthetic	HC	5.0	6.0	0.64	0.58	0.61	-4262	-8.2559	0.0013	0.0034
	Tabu	5.0	6.0	0.64	0.58	0.61	-4262	-8.2559	0.0013	0.0034
	GES	4.0	5.0	0.78	0.58	0.67	-4267	-8.3446	0.0011	0.0035
	PC	11.0	8.0	0.36	0.33	0.35	-4426	-8.3871	0.0013	0.0040
	RLiG greedy	5.0	6.6 ± 0.5	0.58 ± 0.04	0.53 ± 0.04	0.56 ± 0.04	-4262	-8.2585 ± 0.0010	0.0014 ± 0.0000	0.0039 ± 0.0002
	RLiG linear	2.7 ± 2.3	3.4 ± 1.8	0.78 ± 0.12	0.75 ± 0.12	0.76 ± 0.12	-4398	-8.1092 ± 0.0775	0.0011 ± 0.0001	0.0031 ± 0.0003
	RLiG linear, $\rho = 0$	1.0	3.0 ± 0.9	0.82 ± 0.08	0.75 ± 0.07	0.78 ± 0.08	-4261	-8.1079 ± 0.0014	0.0012 ± 0.0002	0.0033 ± 0.0003
	RLiG DQN	2.8 ± 1.3	4.4 ± 1.0	0.71 ± 0.08	0.68 ± 0.09	0.70 ± 0.08	-4271	-8.1340 ± 0.0557	0.0012 ± 0.0001	0.0034 ± 0.0003
	RLiG DQN, $\rho = 0$	1.0	3.2 ± 0.6	0.80 ± 0.05	0.73 ± 0.05	0.77 ± 0.05	-4261	-8.1086 ± 0.0015	0.0012 ± 0.0001	0.0032 ± 0.0003
	true DAG	0.0	0.0	1.00	1.00	1.00	-4283	-7.9567	0.0009	0.0025
asia	HC	4.0	5.0	0.57	0.50	0.53	-1110	-2.2895	0.0006	0.0016
	Tabu	4.0	5.0	0.57	0.50	0.53	-1110	-2.2895	0.0006	0.0016
	GES	5.0	5.0	0.57	0.50	0.53	-1111	-2.2920	0.0005	0.0014
	PC	8.0	8.0	0.00	0.00	0.00	-1171	-2.4050	0.0007	0.0014
	RLiG greedy	4.3 ± 0.5	5.0 ± 0.8	0.59 ± 0.08	0.50 ± 0.10	0.54 ± 0.09	-1112	-2.2931 ± 0.0059	0.0006 ± 0.0001	0.0016 ± 0.0001
	RLiG linear	3.4 ± 2.2	3.1 ± 2.0	0.79 ± 0.18	0.74 ± 0.14	0.76 ± 0.15	-1110	-2.2654 ± 0.0168	0.0006 ± 0.0001	0.0014 ± 0.0002
	RLiG linear, $\rho = 0$	1.0	1.6 ± 0.5	0.91 ± 0.07	0.80 ± 0.06	0.85 ± 0.07	-1107	-2.2540 ± 0.0002	0.0006 ± 0.0001	0.0014 ± 0.0002
	RLiG DQN	6.7 ± 1.8	6.4 ± 1.7	0.53 ± 0.15	0.56 ± 0.12	0.54 ± 0.13	-1120	-2.2692 ± 0.0107	0.0005 ± 0.0001	0.0013 ± 0.0003
	RLiG DQN, $\rho = 0$	1.0	1.7 ± 0.5	0.90 ± 0.07	0.79 ± 0.06	0.84 ± 0.06	-1107	-2.2540 ± 0.0002	0.0006 ± 0.0001	0.0013 ± 0.0002
	true DAG	0.0	0.0	1.00	1.00	1.00	-1110	-2.2520	0.0007	0.0015
sachs	HC	5.0	8.0	0.75	0.53	0.62	-4066	-7.5548	0.0011	0.0051
	Tabu	8.0	14.0	0.27	0.18	0.21	-4063	-7.5701	0.0014	0.0058
	GES	13.0	14.0	0.40	0.24	0.30	-4062	-7.6182	0.0013	0.0059
	PC	7.0	14.0	0.25	0.18	0.21	-4141	-7.6644	0.0013	0.0078
	RLiG greedy	5.3 ± 0.5	7.8 ± 0.4	0.79 ± 0.04	0.54 ± 0.02	0.64 ± 0.03	-4066	-7.5669 ± 0.0164	0.0011 ± 0.0000	0.0052 ± 0.0001
	RLiG linear	7.2 ± 2.6	11.1 ± 2.7	0.49 ± 0.17	0.37 ± 0.14	0.42 ± 0.15	-4068	-7.5041 ± 0.0509	0.0014 ± 0.0002	0.0054 ± 0.0005
	RLiG linear, $\rho = 0$	13.0	14.4 ± 0.7	0.36 ± 0.07	0.21 ± 0.04	0.27 ± 0.05	-4062	-7.6181 ± 0.0004	0.0016 ± 0.0001	0.0062 ± 0.0002
	RLiG DQN	8.4 ± 3.2	12.0 ± 1.9	0.46 ± 0.12	0.33 ± 0.10	0.38 ± 0.11	-4071	-7.5327 ± 0.0039	0.0014 ± 0.0002	0.0054 ± 0.0005
	RLiG DQN, $\rho = 0$	13.0	14.6 ± 0.7	0.35 ± 0.07	0.21 ± 0.04	0.26 ± 0.05	-4062	-7.6180 ± 0.0003	0.0014 ± 0.0002	0.0061 ± 0.0002
	true DAG	0.0	0.0	1.00	1.00	1.00	-4184	-7.3462	0.0016	0.0053
child	HC	18.0	15.0	0.63	0.48	0.55	-6834	-12.8420	0.0011	0.0042
	Tabu	18.0	15.0	0.63	0.48	0.55	-6834	-12.8420	0.0011	0.0042
	GES	13.0	10.0	0.80	0.64	0.71	-6697	-12.4589	0.0011	0.0031
	PC	21.0	17.0	0.62	0.32	0.42	-7435	-13.9031	0.0012	0.0109
	RLiG greedy	18.0	13.6 ± 0.5	0.71 ± 0.03	0.54 ± 0.02	0.61 ± 0.02	-6820	-12.8504 ± 0.0009	0.0012 ± 0.0000	0.0042 ± 0.0000
	RLiG linear	12.4 ± 1.6	9.8 ± 1.7	0.79 ± 0.06	0.65 ± 0.06	0.71 ± 0.06	-6700	-12.4436 ± 0.0158	0.0011 ± 0.0001	0.0032 ± 0.0002
	RLiG linear, $\rho = 0$	10.1 ± 0.8	10.1 ± 0.8	0.80 ± 0.00	0.64 ± 0.03	0.71 ± 0.04	-6697	-12.4587 ± 0.0002	0.0011 ± 0.0001	0.0032 ± 0.0002
	RLiG DQN	23.4 ± 3.9	19.8 ± 3.7	0.57 ± 0.14	0.41 ± 0.06	0.47 ± 0.08	-6946	-13.0230 ± 0.1037	0.0012 ± 0.0001	0.0051 ± 0.0004
	RLiG DQN, $\rho = 0$	20.3 ± 3.7	17.6 ± 2.9	0.63 ± 0.14	0.40 ± 0.06	0.49 ± 0.08	-6887	-12.9718 ± 0.1110	0.0012 ± 0.0002	0.0054 ± 0.0005
	true DAG	0.0	0.0	1.00	1.00	1.00	-6777	-12.3526	0.0012	0.0032
insurance	HC	40.0	39.0	0.60	0.35	0.44	-7816	-14.4189	0.0014	0.0051
	Tabu	41.0	40.0	0.59	0.33	0.42	-7780	-14.4306	0.0013	0.0048
	GES	36.0	33.0	0.75	0.40	0.53	-7656	-14.1711	0.0013	0.0045
	PC	46.0	40.0	0.62	0.25	0.36	-8459	-15.7651	0.0012	0.0151
	RLiG greedy	41.0	37.0	0.66	0.37	0.47 ± 0.00	-7699	-14.2535 ± 0.0011	0.0012 ± 0.0000	0.0045 ± 0.0000
	RLiG linear	37.7 ± 4.5	33.9 ± 3.1	0.70 ± 0.07	0.38 ± 0.04	0.50 ± 0.05	-7681	-14.2013 ± 0.0480	0.0013 ± 0.0001	0.0046 ± 0.0001
	RLiG linear, $\rho = 0$	35.2 ± 1.0	32.1 ± 0.9	0.77 ± 0.02	0.41 ± 0.01	0.54 ± 0.02	-7656	-14.1720 ± 0.0028	0.0013 ± 0.0001	0.0046 ± 0.0001
	RLiG DQN	58.2 ± 5.7	55.2 ± 5.7	0.32 ± 0.08	0.17 ± 0.03	0.22 ± 0.05	-8122	-15.1329 ± 0.2042	0.0013 ± 0.0002	0.0063 ± 0.0006
	RLiG DQN, $\rho = 0$	51.2 ± 3.6	47.3 ± 4.6	0.46 ± 0.11	0.21 ± 0.05	0.29 ± 0.06	-7904	-14.8658 ± 0.1227	0.0012 ± 0.0001	0.0060 ± 0.0003
	true DAG	0.0	0.0	1.00	1.00	1.00	-9516	-13.9895	0.0072	0.0184
alarm	HC	41.0	37.0	0.44	0.37	0.40	-6103	-11.4264	0.0010	0.0028
	Tabu	45.0	38.0	0.42	0.37	0.40	-6100	-11.4255	0.0010	0.0027
	GES	31.0	30.0	0.59	0.50	0.54	-6070	-11.3882	0.0009	0.0026
	PC	34.0	31.0	0.71	0.33	0.45	-7794	-15.1033	0.0010	0.0098
	RLiG greedy	40.3 ± 0.9	35.9 ± 1.3	0.45 ± 0.02	0.37 ± 0.01	0.41 ± 0.01	-6127	-11.5137 ± 0.0146	0.0010 ± 0.0000	0.0029 ± 0.0001
	RLiG linear	41.5 ± 3.8	37.3 ± 2.9	0.43 ± 0.07	0.38 ± 0.06	0.40 ± 0.06	-6114	-11.4199 ± 0.0360	0.0010 ± 0.0001	0.0028 ± 0.0001
	RLiG linear, $\rho = 0$	36.4 ± 2.7	33.1 ± 1.8	0.51 ± 0.04	0.44 ± 0.04	0.47 ± 0.04	-6079	-11.3824 ± 0.0125	0.0009 ± 0.0000	0.0026 ± 0.0001
	RLiG DQN	63.2 ± 7.2	61.8 ± 7.2	0.19 ± 0.07	0.15 ± 0.06	0.17 ± 0.06	-7006	-13.3424 ± 0.3180	0.0010 ± 0.0001	0.0043 ± 0.0006
	RLiG DQN, $\rho = 0$	47.8 ± 2.6	43.0 ± 4.1	0.38 ± 0.08	0.24 ± 0.06	0.29 ± 0.07	-6408	-12.4038 ± 0.3018	0.0009 ± 0.0001	0.0038 ± 0.0005
	true DAG	0.0	0.0	1.00	1.00	1.00	-6517	-10.8442	0.0010	0.0027

Table 3: All six networks at 500 training rows. The capped methods run at $k = 3$, PC is uncapped. BIC is on the training rows, LL is held out per row on 5000 test rows, and $\overline{JS}_2$ compares two-way marginals of the test rows against 5000 rows simulated from the learned network. Higher BIC, LL, precision, recall and F1 are better, SHD and $\overline{JS}$ lower. The true DAG row is only a reference.

4 Results

4.1 RLIG against the classical learners

$\rho = 1$ is the hybrid the sheet specifies and $\rho = 0$ switches the generative half off. I lead with $\rho = 0$ because that is where the result is.

At $\rho = 0$ the linear agent is the best structure learner in the table on three of the six networks. The synthetic network and ASIA both hit SHD 1.0 on every seed, F1 0.78 and 0.85, against 5.0 and 4.0 for hill climbing and 4.0 and 5.0 for GES. ASIA’s BIC of -1107 is better than the true graph’s -1110 and its held-out log-likelihood of -2.2540 is about two thousandths off the true graph’s. Insurance beats GES on structure, 35.2 against 36.0, at the same BIC to the precision the table prints, though that margin is smaller than its own seed spread of ± 1.0 .

It loses on the other three, and the losses are not the same kind. On ALARM GES is clearly ahead, 31.0 against 36.4, and that is the one clean win for equivalence-class search here. On Child the ablation lands in GES’s equivalence class, same BIC -6697 and same SHD 13.0, and it is the one network where the hybrid at $\rho = 1$ edges it, 12.4 against 13.0. That is inside its spread of ± 1.6 and beats 13 on only four seeds in ten, so I would not lean on it. On Sachs hill climbing gets SHD 5.0 at BIC -4066 , while GES and RLIG at $\rho = 0$ both find the better BIC of -4062 and both land on 13.0. Two unrelated searches reaching the better score and the same worse SHD points at the criterion at this sample size, though it does not prove they returned the same graph.

Greedy does worse on structure than the hill climber it imitates on three of six, and ties it on two more, never by more than one edge. It takes the same best-delta move rule but returns the best graph whose *hybrid* score was measured, and with $L = 5$ that is one visited graph in five.

The DQN is the weakest variant and it is the wide networks where it loses. At $\rho = 0$ it matches the linear agent exactly on the three smallest, trails it on Child, 20.3 against 13.0, then falls to 51.2 on Insurance and 47.8 on ALARM where the linear agent gets 35.2 and 36.4.

4.2 Learning curves

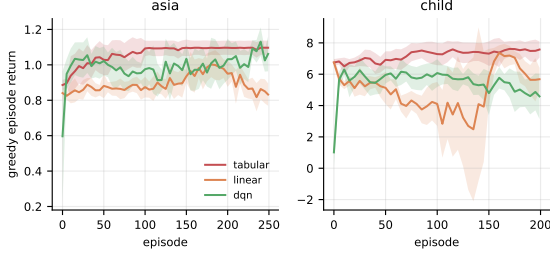


Figure 2: Greedy episode return against training episode, mean and one standard deviation over ten seeds. The curve comes from a separate greedy evaluation episode, so decaying ε cannot pass for learning.

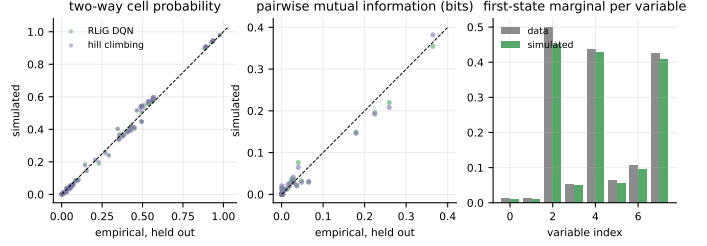


Figure 3: Posterior predictive checks on ASIA, 5000 rows simulated from CPTs fitted on the 500 training rows. Left, every two-way cell probability against held out. Middle, pairwise mutual information. Right, the first-state marginal of each variable.

On ASIA the tabular curve climbs to about 1.10 by episode a hundred and then barely moves, finishing highest of the three. That only holds while the table stays small, about 490 distinct states on ASIA against about 4800 on Child, roughly one new state per three steps there. On best hybrid score the three finish close rather than equal, 101.87 linear, 101.85 tabular, 101.84 DQN.

4.3 Tiling, with and without

network	tiling	sims	SHD	held-out LL	$\overline{JS}_2$	s
asia	no tiles	0	2.7 ± 1.7	-2.2695 ± 0.0216	0.0013 ± 0.0002	7.2
asia	$L=10, I_g=\{0\}$	867	7.6 ± 2.5	-2.2784 ± 0.0174	0.0014 ± 0.0003	8.3
asia	$L=5, I_g=\{0\}$	1929	6.7 ± 1.8	-2.2692 ± 0.0107	0.0013 ± 0.0003	9.7
asia	$L=5, I_g=\{0,2\}$	3716	6.9 ± 2.3	-2.2781 ± 0.0161	0.0013 ± 0.0002	12.1
asia	$L=2, I_g=\{0\}$	4249	5.6 ± 2.8	-2.2632 ± 0.0123	0.0013 ± 0.0003	12.8
asia	every step	8689	4.5 ± 1.6	-2.2674 ± 0.0115	0.0013 ± 0.0002	18.6
asia	learned	1538	8.5 ± 2.2	-2.2951 ± 0.0208	0.0015 ± 0.0002	11.2
child	no tiles	0	19.1 ± 2.5	-13.0764 ± 0.1449	0.0056 ± 0.0007	12.1
child	$L=10, I_g=\{0\}$	1197	23.0 ± 4.0	-13.0893 ± 0.1762	0.0055 ± 0.0005	18.0
child	$L=5, I_g=\{0\}$	2479	23.4 ± 3.9	-13.0230 ± 0.1037	0.0051 ± 0.0004	24.2
child	$L=5, I_g=\{0,2\}$	4822	19.2 ± 3.3	-12.7724 ± 0.1056	0.0042 ± 0.0004	35.5
child	$L=2, I_g=\{0\}$	5642	20.4 ± 4.4	-12.8004 ± 0.0929	0.0044 ± 0.0003	39.4
child	every step	11326	19.5 ± 2.0	-12.7769 ± 0.0999	0.0046 ± 0.0004	67.5
child	learned	2056	23.0 ± 3.4	-13.1779 ± 0.1428	0.0056 ± 0.0003	25.8
synthetic	no tiles	0	2.0 ± 1.3	-8.1387 ± 0.0453	0.0032 ± 0.0004	7.5
synthetic	$L=10, I_g=\{0\}$	876	4.1 ± 1.9	-8.1796 ± 0.0878	0.0035 ± 0.0004	9.0
synthetic	$L=5, I_g=\{0\}$	1933	2.8 ± 1.3	-8.1340 ± 0.0557	0.0034 ± 0.0003	10.7
synthetic	$L=5, I_g=\{0,2\}$	3662	3.2 ± 2.0	-8.1508 ± 0.0970	0.0031 ± 0.0002	13.6
synthetic	$L=2, I_g=\{0\}$	4202	4.5 ± 0.7	-8.1891 ± 0.0730	0.0033 ± 0.0004	14.5
synthetic	every step	8267	3.1 ± 2.0	-8.0948 ± 0.0833	0.0032 ± 0.0004	21.2
synthetic	learned	1577	5.7 ± 2.3	-8.2517 ± 0.1093	0.0037 ± 0.0005	12.1

Table 4: Tile pattern against cost. *sims* counts the simulate-and-compare calls executed after the per-graph cache. The learned tiler is capped at $\lfloor T/5 \rfloor$ per episode, matching the $L = 5, I_g = \{0\}$ row, so the two share a budget. It spends about 80 percent of that cap, which the *sims* column shows. *no tiles* is the BIC-only control.

Tiling behaves as designed on the cost side, spending 22 to 23 percent at $L = 5$ with $I_g = \{0\}$ of what evaluating every step spends. On structure it comes out ahead on one of the three, 2.8 against 3.1 on the synthetic network, while ASIA and Child prefer the full schedule, 6.7 against 4.5 and 23.4 against 19.5.

It still cannot make the generative term worth having on structure. At zero cost the BIC-only control wins on all three, 2.7 against 6.7 on ASIA, 19.1 against 23.4 on Child and 2.0 against 2.8 on the synthetic network. Against the best tiled row it is two wins and a tie, and per seed even the wins are close, ASIA six seeds of ten and the synthetic network four with five level. On held-out likelihood it goes the other way on all three, and on Child that gap is real, -12.77 against -13.08 .

The learned tiler also fails to beat the fixed tile at matched budget. It is better on Child by 0.4 SHD, worse on the other two, and worse on held-out likelihood on all three. I read that as a result about my meta-controller rather than about learned tiling in general, because the two heads share a return and never coordinate.

4.4 The weight on the generative half

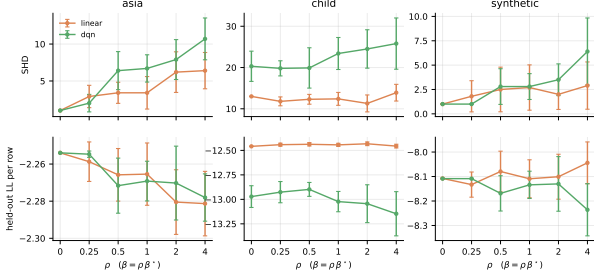


Figure 4: Sweeping ρ , where $\beta = \rho \beta^*$. Top row SHD, bottom row held-out log-likelihood, with one standard deviation over ten seeds. At $\rho = 0$ the generative term drops out of the objective, though the run still pays for the simulations the tile schedules.

variant	asia		child		synthetic	
	SHD	LL	SHD	LL	SHD	LL
reference	6.0	-2.274	25.5	-13.092	2.5	-8.110
uniform exploration	4.5	-2.256	19.0	-12.915	6.0	-8.168
no action mask	6.0	-2.274	25.5	-13.092	2.5	-8.110
Gen vs train split	6.0	-2.276	20.0	-12.932	3.5	-8.186
no BIC in features	9.0	-2.294	37.5	-13.758	9.5	-8.429
no Gen cache	7.0	-2.276	21.5	-12.966	6.5	-8.146
reward = sheet form	9.0	-2.309	38.0	-14.202	6.0	-8.161
init = random	9.0	-2.263	21.5	-13.023	2.5	-8.137
init = hill climb	2.5	-2.254	15.0	-12.677	5.5	-8.145
gamma = 0.9	6.5	-2.275	15.0	-12.538	1.0	-8.107
in-degree k = 2	6.0	-2.276	26.0	-12.926	7.5	-8.248

Table 5: One design choice changed at a time against the reference run, two seeds each, every row a DQN run. LL is held-out log-likelihood per row.

I ran this sweep to test whether the generative half helps. Mostly it does not. Four of the six curves have their best SHD at $\rho = 0$, and both exceptions are Child, where the linear agent is flat and noisy across the range, 11.3 to 13.9, and the DQN’s minimum at $\rho = 0.25$ sits inside its own $\rho = 0$ spread. Worse, what it buys on two-way JS sits inside the noise. Five of the six curves have their best JS above zero but no gap clears the seed spread.

I expected a trade-off, better generation bought with worse structure, and one seed at a shorter budget seemed to show it. Ten seeds did not. The one positive result is Child with the linear agent, 13.0 down to 11.3 with held-out likelihood better on eight seeds of ten.

4.5 Ablations

I had expected exploration to dominate Table 5 and it does not. The two rows that move it most are dropping the BIC delta from the action features, which costs 3.0, 12.0 and 7.0 SHD on the three networks, and switching the reward from the difference form to the task sheet’s level form, which costs 3.0, 12.5 and 3.5. Section 2.5 gives one reason for the second, the level form’s return can carry $m/2$ times the noise, four at $T = 40$ and six on Child, though it also changes what is optimised and this table cannot separate the two. Warm starting from hill climbing and dropping γ to 0.9 are the two biggest improvements and they tie on Child, both at 15.0 against the reference’s 25.5. Warm starting gains 3.5 on ASIA but it loses 3.0 on the synthetic network, so its not the free win it was. Boltzmann against uniform exploration is mixed, and it runs the other way to what I expected. Uniform is ahead on ASIA and Child, 4.5 against 6.0 and 19.0 against 25.5, and behind on the synthetic network, 6.0 against 2.5.

Turning off action masking returns the reference numbers exactly because the flag is not wired to anything, the environment masks regardless. So that row only checks that nothing else depends on it.

Posterior predictive checks on ASIA are in Figure 3. The sample size study runs in `task-11.ipynb` under `samples` and is not tabled here.

4.6 HD extensions 3 and 4, measured

Table 6 asks whether a score can tell apart the graphs a search has to choose between. The witness gap ranks them closer to held-out likelihood than the marginal score on all three networks, r_s of 0.90, 0.93 and 0.95 against 0.80, 0.77 and 0.86, and it stands further above its own noise on all three. It does not always put the true DAG first though. On ASIA it beats every neighbour where the marginal score beats 69 percent, but on the synthetic network and Child it does worse at that, 0.80 against 0.95 and 0.82 against 0.88. The hard accuracy is noisier than the gap on every network and ranks worse on ASIA and Child, so the choice made on ASIA held up, though it puts the synthetic true DAG first more often. None of it is cheap, one disc score costs about 30 ms against 1.3 to 5.0 for the marginal one, because each disc score is a small training run.

network	score	ms	noise sd	spread	spread/noise	r_s with LL	true wins
synthetic	marginal JS	1.7	0.00019	0.00054	2.9	0.80	0.95
	disc, witness gap	29.4	0.0102	0.0465	4.6	0.90	0.80
	disc, accuracy	29.4	0.0184	0.0576	3.1	0.90	0.90
asia	marginal JS	1.3	0.00012	0.00068	5.9	0.77	0.69
	disc, witness gap	28.4	0.0043	0.0348	8.1	0.93	1.00
	disc, accuracy	28.4	0.0122	0.0666	5.5	0.70	0.62
child	marginal JS	5.0	0.00011	0.00051	4.5	0.86	0.88
	disc, witness gap	30.9	0.0153	0.0737	4.8	0.95	0.82
	disc, accuracy	30.8	0.0195	0.0888	4.6	0.92	0.67

Table 6: Can each score tell the true DAG from its neighbours. Per network the true DAG, its one-edge deletions, eight legal one-edge additions and the empty graph, each scored with the same five fresh seeds and no cache. *noise sd* is the mean standard deviation over the five for every graph, *spread* the standard deviation of the means, r_s their rank correlation with held-out LL, and *true wins* the share of neighbours the true DAG beats. Spread, r_s and true wins leave out the empty graph, which would flatter every score.

Against the marginal score at the same ρ the learned one gets the better mean SHD in 8 of the 12 cells of Table 7 and the better held-out likelihood in 9. The clearest case is the synthetic network with the linear agent, which at $\rho = 1$ lands in the true equivalence class on all ten seeds and beats $\rho = 0$ on likelihood on every seed. Child with the linear agent edges $\rho = 0$ too, 12.0 against 13.0 at $\rho = 4$, seven seeds better and one worse. On ASIA and for the DQN $\rho = 0$ still wins on structure, and the DQN on Child falls apart at $\rho = 4$. The optimism column is positive on average in every row with weight on Gen and within its own spread of zero in every $\rho = 0$ row. That's what Section 2.5 predicts, since only a score being maximised gets picked for its lucky noise.

network	agent	Gen	$\rho = 1, \text{ or } 0 \text{ for none}$			$\rho = 4$	
			SHD	held-out LL	optimism %	SHD	held-out LL
synthetic	linear	none	1.0	-8.1079 ± 0.0014	0.6 ± 2.1		
		marginal	2.7 ± 2.3	-8.1092 ± 0.0775	7.4 ± 1.3	2.9 ± 2.4	-8.0441 ± 0.0854
		disc	0.0	-7.9584 ± 0.0016	5.7 ± 2.7	1.6 ± 2.5	-7.9690 ± 0.0302
	dqn	none	1.0	-8.1086 ± 0.0015	0.5 ± 5.1		
		marginal	2.8 ± 1.3	-8.1340 ± 0.0557	6.2 ± 1.3	6.4 ± 3.4	-8.2358 ± 0.1065
		disc	2.3 ± 2.1	-8.0424 ± 0.0688	4.9 ± 2.9	6.6 ± 2.7	-8.0722 ± 0.0688
asia	linear	none	1.0	-2.2540 ± 0.0002	3.2 ± 4.5		
		marginal	3.4 ± 2.2	-2.2654 ± 0.0168	11.4 ± 2.3	6.4 ± 2.5	-2.2813 ± 0.0174
		disc	2.6 ± 1.4	-2.2559 ± 0.0026	2.9 ± 0.8	5.0 ± 2.0	-2.2571 ± 0.0031
	dqn	none	1.0	-2.2540 ± 0.0002	0.9 ± 2.6		
		marginal	6.7 ± 1.8	-2.2692 ± 0.0107	9.0 ± 3.1	10.7 ± 2.8	-2.2780 ± 0.0127
		disc	5.3 ± 2.6	-2.2604 ± 0.0050	2.1 ± 1.3	8.5 ± 2.6	-2.2601 ± 0.0037
child	linear	none	13.0	-12.4587 ± 0.0002	0.4 ± 3.8		
		marginal	12.4 ± 1.6	-12.4436 ± 0.0158	4.9 ± 1.6	13.9 ± 2.0	-12.4561 ± 0.0189
		disc	12.8 ± 3.0	-12.4545 ± 0.0374	7.1 ± 2.7	12.0 ± 1.0	-12.4566 ± 0.0579
	dqn	none	20.3 ± 3.7	-12.9718 ± 0.1110	-0.3 ± 3.0		
		marginal	23.4 ± 3.9	-13.0230 ± 0.1037	2.4 ± 1.9	25.8 ± 6.2	-13.1470 ± 0.2257
		disc	23.8 ± 1.6	-13.0163 ± 0.1713	9.2 ± 6.7	37.9 ± 9.9	-13.7432 ± 0.5912

Table 7: The discriminator score in the reward against the marginal score, each with its own β^* , ten seeds. *none* is $\rho = 0$, the marginal score measured at zero weight. *optimism* is the recorded Gen of the returned graph minus the mean of five fresh rescoring, as a percent of that score's spread between hill climbing's graph and the empty one.

5 Failure Modes

With the marginal score the hypothesis failed on five of the six agent and network pairs, all but Child linear.

The marginal score is too flat to steer with. Gen averages JS divergences over low-order marginals, and many different structures reproduce those equally well. On ASIA at 500 rows its whole range between the empty graph and hill climbing's graph is about 0.0062 against 0.65 for the structural half. Re-running the whole grid at $N_s = 5000$, five times the rows, only widened it from 0.0045, while a generative step on ASIA went from 10.5 to 32.0 times the cost of a cheap one. So the flatness points at what Gen measures more than at how well it is measured, and extension 4 tests that directly. The learned score does better in most cells, so the flatness was partly the score, but $\rho = 0$ still wins on ASIA and for the DQN.

The DQN loses to the linear agent, and it loses on the wide networks. Three things could be doing it. On ALARM it gets about 13000 gradient steps for 15000 parameters, and its bootstrap target maxes over 24 of about 1300 actions, a bias I did not measure. The linear agents fifteen features are a strong inductive bias that lands close to right. The first two are fixable with compute. I do not know about the third.

The BIC feature is doing more of the work than the learning is. Dropping it costs the most in Section 4.5 summed over the three networks, so the DQN is learning how to weigh the structural score rather than to do without it. Boltzmann against uniform can't be judged on three widths and two seeds. On Child the two seeds even disagree in sign, +17 and -4. And Boltzmann explores where BIC already says to look.

Things that belong to the setting rather than the method. ALARM's numbers are relative, since $k = 3$ puts its true graph out of reach. Seed variance is large. On 16 of the 20 rows that move at all the standard deviation over ten seeds is wider than the distance to the nearest other method, and RLIG linear on ALARM runs 32 to 47 across the ten. The ablation and sample size tables are two seeds each and read as sketches. Episode counts were chosen to fit a runtime budget rather than to convergence. A tuned DQN might close some of the gap in Section 4.1 and I cannot rule that out.

6 Conclusion

With the generative half switched off, the linear agent is the best structure learner in the comparison on three of the six networks, and on the synthetic network and ASIA it lands within one edge of the true equivalence class on every seed. An agent that keeps editing past the first move that fails to improve can climb out the other side of where hill climbing stops. Nothing in the grid varies T , so that is a reading and not a measured effect. Its still the part of the RL framing worth keeping, and it has nothing to do with the generative reward.

The generative reward mostly does not pay for itself. Raising ρ degrades structure recovery on four of the six curves and does not reliably improve the metric it optimises, Child with the linear agent at 12.4 against 13.0 being the one place the marginal score's hybrid wins, and that inside its spread. Tiling works as a cost control, at about a fifth of what evaluating every step costs, but only comes out ahead on structure on the synthetic network. Section 2.5 gives the difference from a noise bound $m/2$ times tighter than the level form's. A learned discriminator in place of the marginal score does better in most cells, and with the linear agent at $\rho = 1$ finds the synthetic network's true class on every seed. So part of the problem was the score, but $\rho = 0$ still wins on ASIA and for the DQN.

References

- [1] SIT320 High Distinction task sheet, *Building your own Module*, Option A. The environment, the tiling scheme, the required baselines and the metric list are quoted from it.
- [2] G. Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, 1978. The $\frac{k}{2} \log n$ penalty, stated in maximisation form. Says nothing about Bayesian networks, so it is cited only for the penalty.
- [3] L. M. de Campos. A Scoring Function for Learning Bayesian Networks based on Mutual Information and Conditional Independence Tests. *Journal of Machine Learning Research*, 7:2149–2187, 2006.
- [4] D. Heckerman, D. Geiger and D. M. Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, 20:197–243, 1995. Dirichlet parameter priors. Abstract and metadata only.
- [5] G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992. Dirichlet counts for CPT estimation. Abstract only.
- [6] D. M. Chickering. Optimal Structure Identification With Greedy Search. *Journal of Machine Learning Research*, 3:507–554, 2002.
- [7] D. M. Chickering. Learning Equivalence Classes of Bayesian-Network Structures. *Journal of Machine Learning Research*, 2:445–498, 2002.
- [8] C. Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 403–410, 1995.
- [9] I. Tsamardinos, L. E. Brown and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 2006. Structural Hamming distance, defined on PDAGs, an undirected against directed mismatch counting as one error. The copy I read prints no volume or pages, so none is quoted.
- [10] P. Spirtes, C. Glymour and R. Scheines. *Causation, Prediction, and Search*, second edition. The MIT Press, 2000.
- [11] M. Scutari. *bnlearn Bayesian Network Repository*, discrete networks, <https://www.bnlearn.com/bnrepository/>. Source of the five `.bif` files embedded in the notebook, and of the node, arc and parameter counts the notebook checks against. The maximum in-degree column in Table 2 is counted off those files rather than quoted. See also M. Scutari, Learning Bayesian Networks with the bnlearn R Package, *Journal of Statistical Software*, 35(3):1–22, 2010.
- [12] S. L. Lauritzen and D. J. Spiegelhalter. Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems. *Journal of the Royal Statistical Society, Series B*, 50(2):157–224, 1988. ASIA. The repository credits Child to Spiegelhalter and Cowell, *Bayesian Statistics 4*, 1992, which I did not open.
- [13] I. A. Beinlich, H. J. Suermondt, R. M. Chavez and G. F. Cooper. The ALARM Monitoring System: A Case Study with two Probabilistic Inference Techniques for Belief Networks. In *AIME 89*, pages 247–256, Springer, 1989. ALARM. Abstract and metadata only.
- [14] J. Binder, D. Koller, S. Russell and K. Kanazawa. Adaptive Probabilistic Networks with Hidden Variables. *Machine Learning*, 29(2-3):213–244, 1997. Insurance. Abstract and metadata only.
- [15] K. Sachs, O. Perez, D. Pe’er, D. A. Lauffenburger and G. P. Nolan. Causal protein-signaling networks derived from multi-parameter single-cell data. *Science*, 308(5721):523–529, 2005. Abstract only. The network here is the repository’s discretised version and the data is sampled from it, not the original measurements.
- [16] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*, second edition. The MIT Press, 2018.
- [17] C. J. C. H. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3), 1992. Metadata and abstract only.
- [18] S. Singh, T. Jaakkola, M. L. Littman and C. Szepesvári. Convergence Results for Single-Step On-Policy Reinforcement-Learning Algorithms. *Machine Learning*, 38(3):287–308, 2000.
- [19] V. Mnih et al. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.
- [20] S. Huang and S. Ontañón. A Closer Look at Invalid Action Masking in Policy Gradient Algorithms. *FLAIRS*, volume 35, 2022.
- [21] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf and A. Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13:723–773, 2012.
- [22] J. Lin. Divergence Measures Based on the Shannon Entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.
- [23] D. Krueger, J. Leike, O. Evans and J. Salvatier. Active Reinforcement Learning: Observing Rewards at a Cost. arXiv:2011.06709, 2020.
- [24] W. Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. Sections 1 and 2. Theorem 1 is the bound used in Section 2.5.
- [25] D. Lopez-Paz and M. Oquab. Revisiting Classifier Two-Sample Tests. In *ICLR*, 2017. arXiv:1610.06545. Section 3, the test and its accuracy statistic, Remark 2, and Section 5 on judging generative models with it.
- [26] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang and B. Póczos. MMD GAN: Towards Deeper Understanding of Moment Matching Network. In *NIPS*, 2017. arXiv:1705.08584. Abstract and Section 2, the adversarially learned kernel.