Supplemental Material for Black-Box Policy Search with Probabilistic Programs

Jan-Willem van de Meent **Brooks Paige** **David Tolpin** **Frank Wood**
Department of Engineering Science, University of Oxford

A Anglican

All case studies are implemented in Anglican, a probabilistic programming language that is closely integrated into the Clojure language. In Anglican, the macro `defquery` is used to define a probabilistic model. Programs may make use of user-written Clojure functions (defined with `defn`) as well as user-written Anglican functions (defined with `defm`). The difference between the two is that in Anglican functions may make use of the model special forms `sample`, `observe`, and `predict`, which interrupt execution and require action by the inference back end. In Clojure functions, `sample` is a primitive procedure that generates a random value, `observe` returns a log probability, and `predict` is not available.

Full documentation for Anglican can be found at

```
http://www.robots.ox.ac.uk/~fwood/anglican
```

The complete source code for the case studies can be found at

```
https://bitbucket.org/probprog/black-box-policy-search
```

B Canadian Traveler Problem

The complete results for the Canadian traveler problem, showing the performance and convergence for the learned policies for multiple graphs of different sizes and topologies, are presented in Figures 1 and 2.

C RockSample

The RockSample problem was formulated as a benchmark for value iteration algorithms and is normally evaluated in an infinite horizon setting where the discount factor penalizes sensing and movement. In the original formulation of the problem, movement and sensing incur no cost. The agent gets a reward of 10 for each good rock, as well as for reaching the right edge, but incurs a penalty of -10 when sampling a bad rock.

Here we consider an adaptation of RockSample to a finite horizon setting. We assume sensing is free, and movement incurs a cost of -1. We structure the policy by moving along rocks in a left-to-right order. At each rock the agent sense the closest next rock and chooses to move to it, or discard it and consider the next closest rock. When the agent gets to a rock, it only samples the rock if the rock is good. The parameters describe the prior over the probability of moving to a rock conditioned on the current location and the sensor reading.

D Guess Who

In Table 1 we provide as reference the complete ontology for the Guess Who domain. At each turn, the player asks whether the unknown individual has a particular value of a single attribute.

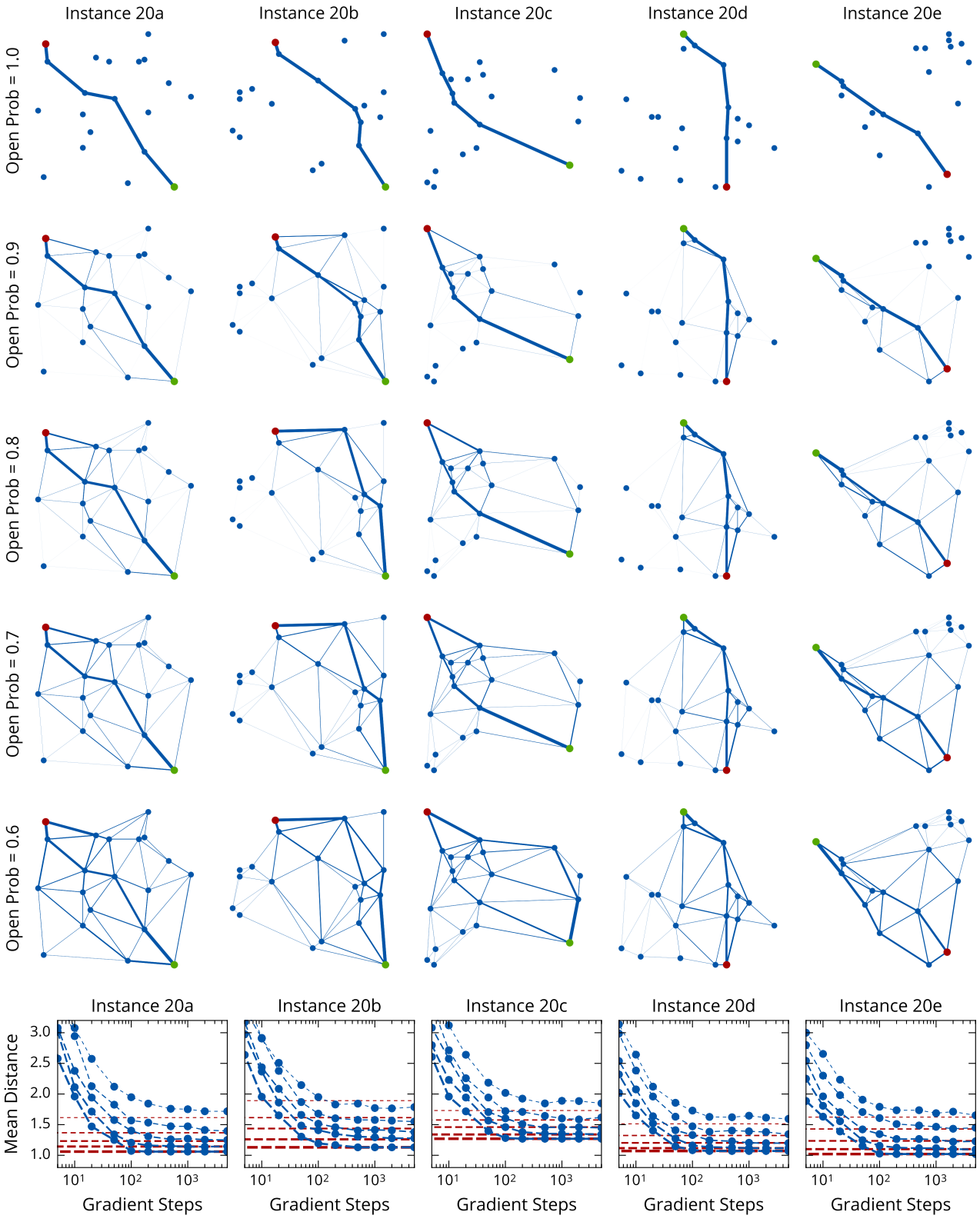

Figure 1: Canadian traveler problem: edge weights, indicating average travel frequency under the learned policy, and convergence for individual instances with 20 nodes.

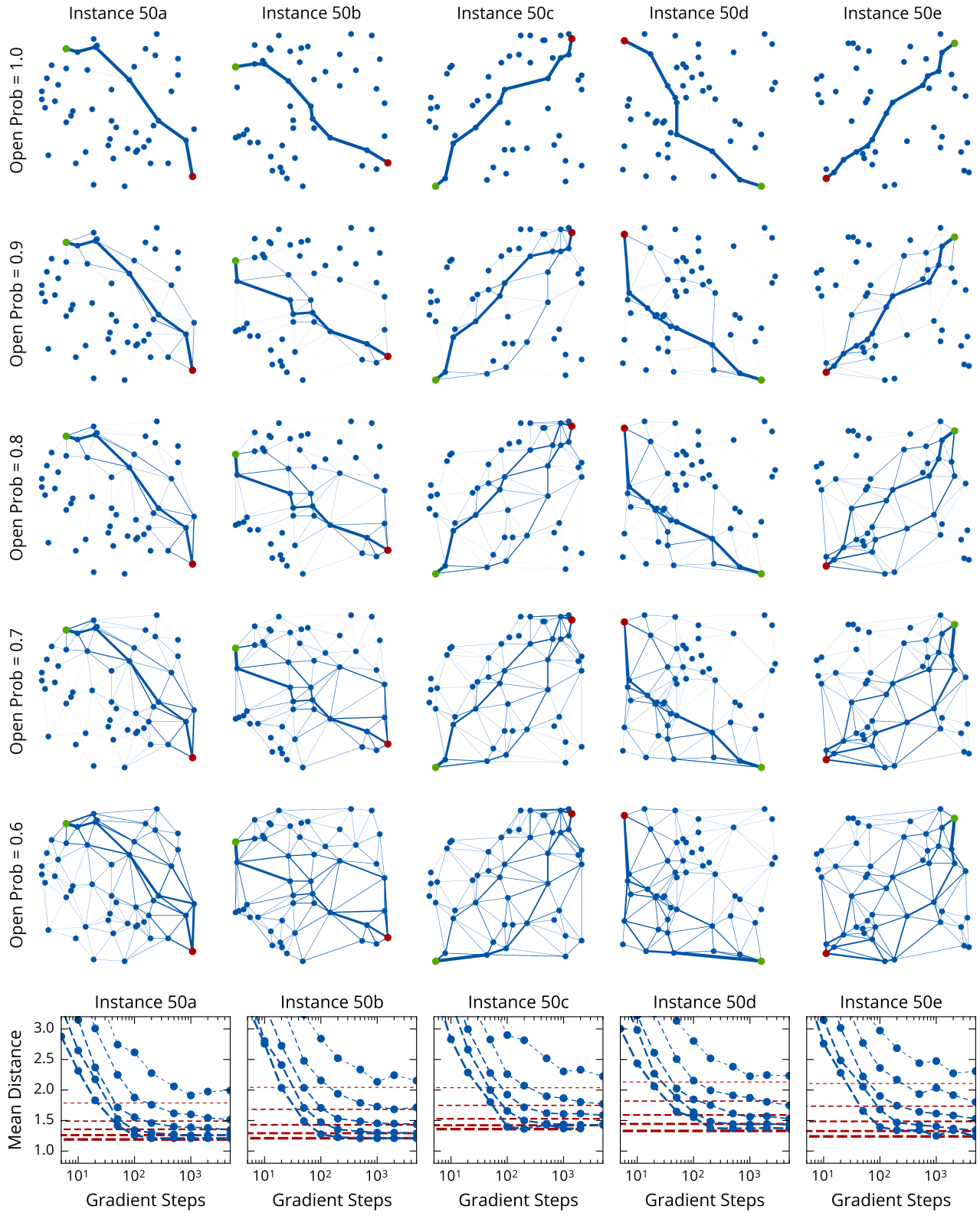

Figure 2: Canadian traveler problem: edge weights, indicating average travel frequency under the learned policy, and convergence for individual instances with 50 nodes.

id	beard	ear-rings	eye-color	gender	glasses	hair-color	hair-length	hair-type	hat	moustache	mouth-size	nose-size	red-cheeks
alex	false	false	brown	male	false	black	short	straight	false	true	large	small	false
alfred	false	false	blue	male	false	ginger	long	straight	false	true	small	small	false
anita	false	false	blue	female	false	blonde	long	straight	false	false	small	small	true
anne	false	true	brown	female	false	black	short	curly	false	false	small	large	false
bernard	false	false	brown	male	false	brown	short	straight	true	false	small	large	false
bill	true	false	brown	male	false	ginger	bald	straight	false	false	small	small	true
charles	false	false	brown	male	false	blonde	short	straight	false	true	large	small	false
claire	false	false	brown	female	true	ginger	short	straight	true	false	small	small	false
david	true	false	brown	male	false	blonde	short	straight	false	false	large	small	false
eric	false	false	brown	male	false	blonde	short	straight	true	false	large	small	false
frans	false	false	brown	male	false	ginger	short	curly	false	false	small	small	false
george	false	false	brown	male	false	white	short	straight	true	false	large	small	false
herman	false	false	brown	male	false	ginger	bald	curly	false	false	small	large	false
joe	false	false	brown	male	true	blonde	short	curly	false	false	small	small	false
maria	false	true	brown	female	false	brown	long	straight	true	false	small	small	false
max	false	false	brown	male	false	black	short	curly	false	true	large	large	false
paul	false	false	brown	male	true	white	short	straight	false	false	small	small	false
peter	false	false	blue	male	false	white	short	straight	false	false	large	large	false
philip	true	false	brown	male	false	black	short	curly	false	false	large	small	true
richard	true	false	brown	male	false	brown	bald	straight	false	true	small	small	false
robert	false	false	blue	male	false	brown	short	straight	false	false	small	large	true
sam	false	false	brown	male	true	white	bald	straight	false	false	small	small	false
susan	false	false	brown	female	false	white	long	straight	false	false	large	small	true
tom	false	false	blue	male	true	black	bald	straight	false	false	small	small	false

Table 1: Ontology for the Guess Who domain, consisting of 24 individuals, characterized by 11 binary attributes and two multi-class attributes.