Neuroevolution-Based Inverse Reinforcement Learning

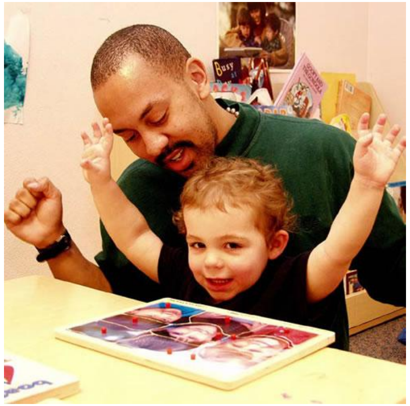
Karan K. Budhraja
Committee: Tim Oates (Chair), Cynthia Matuszek, Tim Finin

Motivation

➔ Thesis overview

awwwwwwwwww
wwwwwwwwwwww
wwwwwwwwwwww

- ◆ Infant learning alone vs Infant learning with assistance

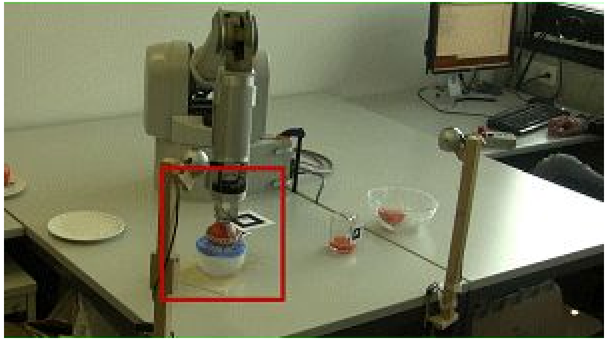


- ◆ Summary: Use genetic algorithms + neural networks to assist in learning

Motivation

- ➔ Reinforcement Learning (RL)
  - Model of learning from experience
  - Inspired by human learning

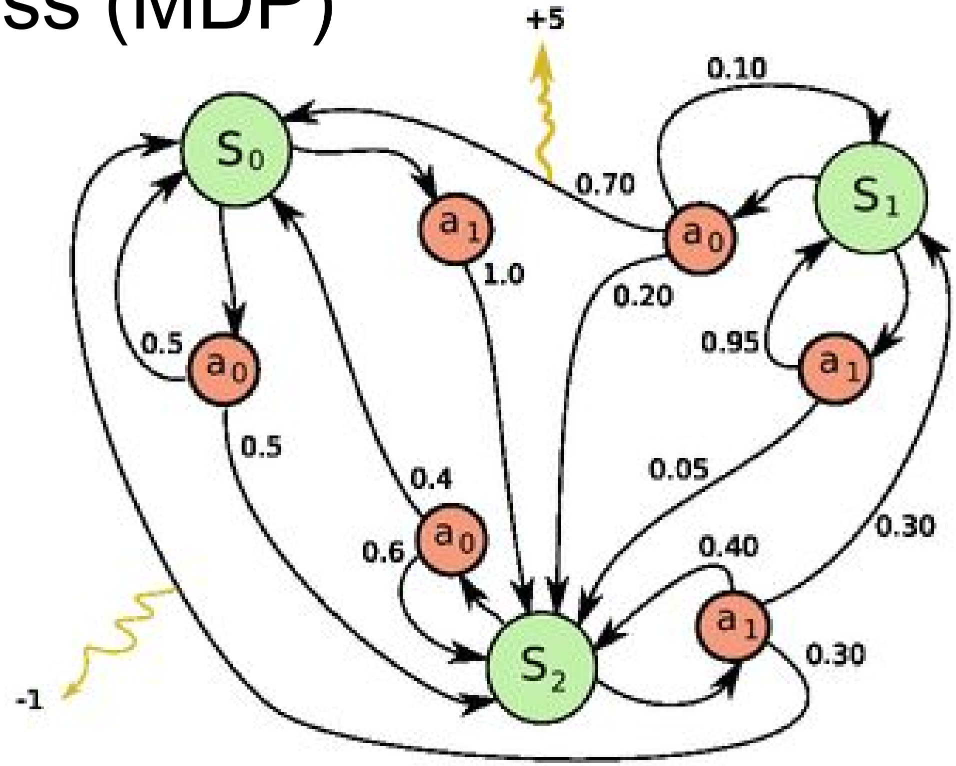
- ➔ Inverse Reinforcement Learning (IRL)
  - Model of learning from example
  - Also inspired by human learning
  - Aligned with Learning from Demonstration (LfD)



Markov Decision Process (MDP)

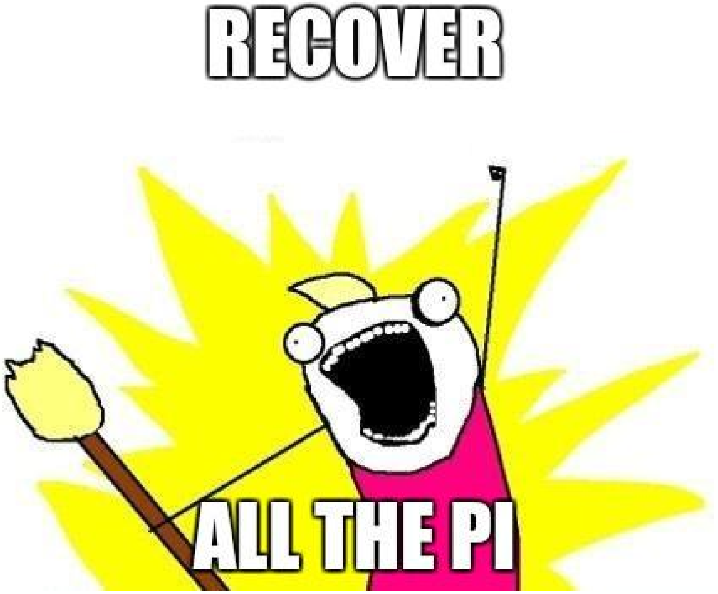
➔ Comprises of (S,A,θ,R,γ)

- States have *features*
- V: State *values* based on aggregated rewards
- π, π^*: *Policy*, optimal policy



Inverse Reinforcement Learning (IRL)

- ➔ Given: (a piece of) some π
  - ◆ Demonstration / Examples
  - ◆ In case of perfect demonstration, $\pi = \pi^*$
- ➔ Goal: Recover R (and all the π)
- ➔ *Focus of this work*
  - ◆ Recover all the π
  - ◆ Assume R is a function of S



Related Work

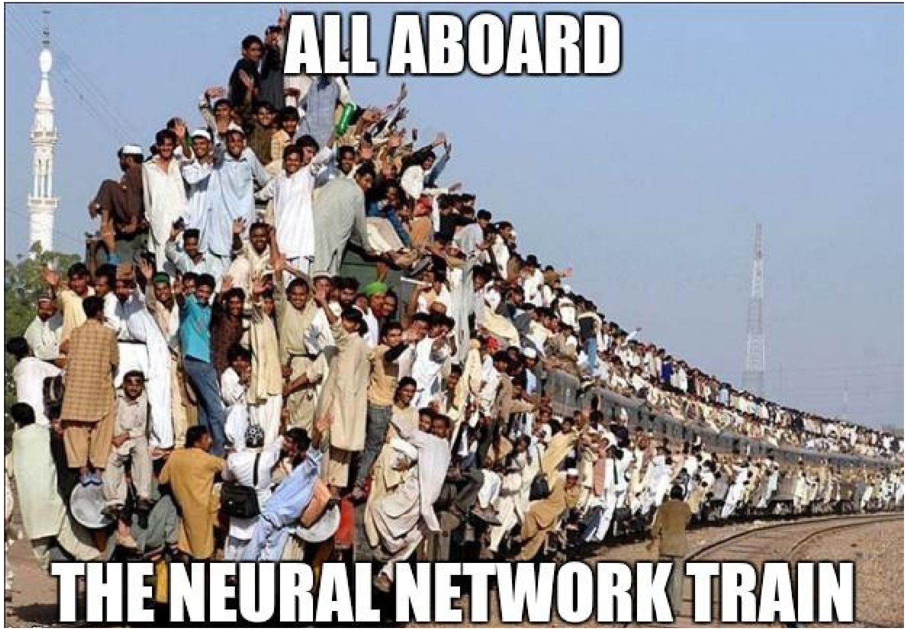
- ➔ Feature construction for IRL (FIRL) (2010)
- ➔ Gaussian Process IRL (GPIRL) (2011)
- ➔ Bayesian Non-Parametric (BNP) approaches (2012, 2013)
- ➔ Expectation Maximization (2015)
- ➔ Maximum Likelihood IRL (2014)

Why not try neural networks (NNs)?

- ➔ Regression trees targeted at linear functions
- ➔ Regression trees / GP regression overfit more
- ➔ NNs generate more abstract features
- ➔ NNs are universal approximators
- ➔ *Why neuroevolution?*
  - ◆ Complexity of function is unknown
  - ◆ Avoid unnecessary neurons / connections

Safe plan

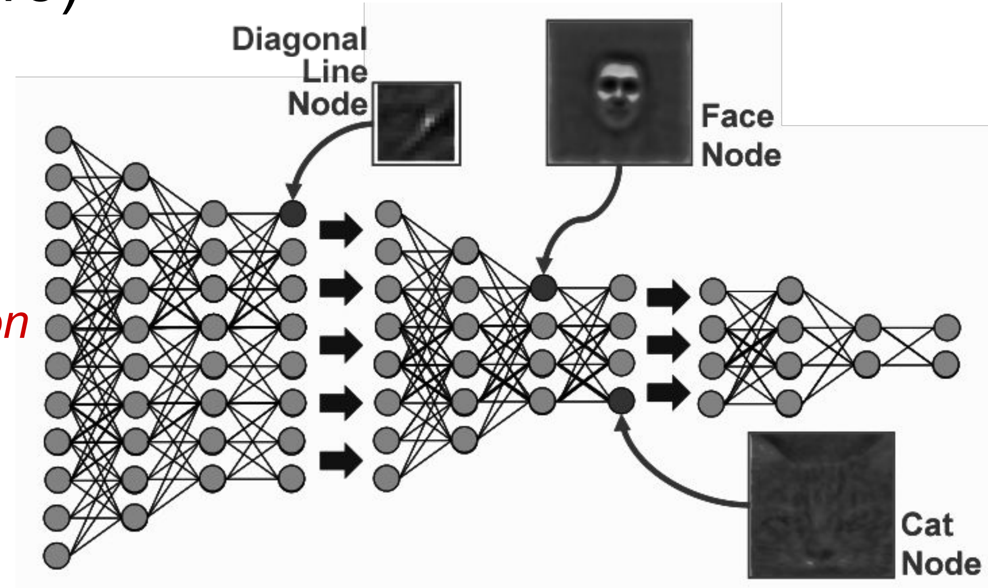
➔ Let’s throw neural networks at it and pray :)



Related Work

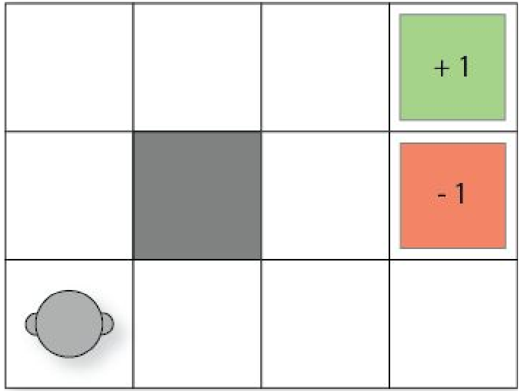
➔ Deep Learning for IRL (2015)

- Similar to this work
- Surpasses existing algorithms
- Intuitively competitive
  - Less efficient
- *Not yet available for comparison*



Problem Definition

- ➔ Grid world MDP
- ➔ Possible actions (5): ← ↑ → ↓ ∅
- ➔ State features: *defined by tooklit*
  - ◆ We use GPIRL MATLAB toolkit

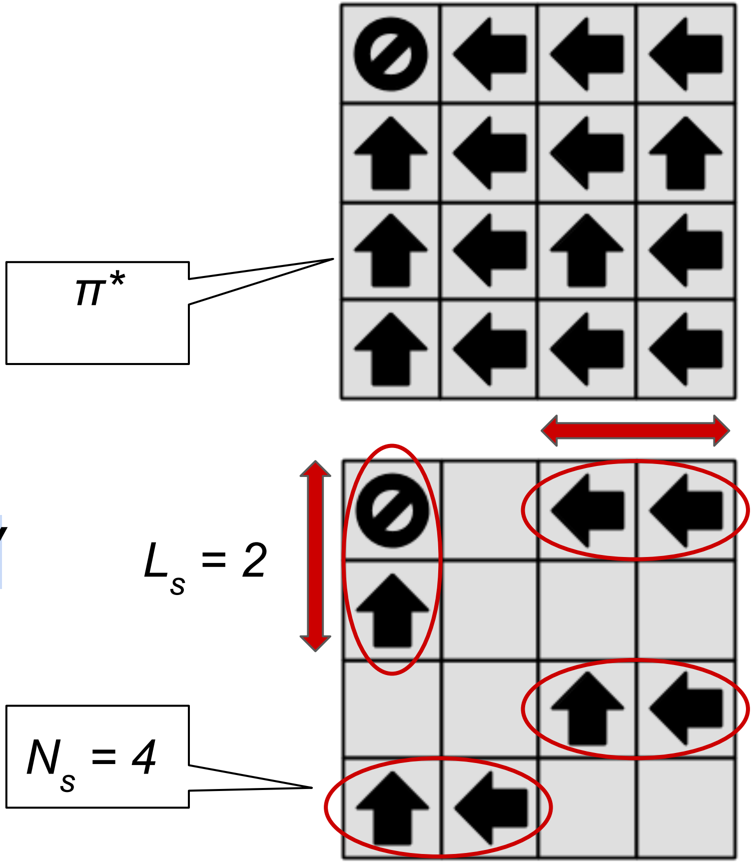


1 1	0 1
1 0	0 0

1 1 1 1	0 1 1 1	0 0 1 1
1 1 0 1	0 1 0 1	0 0 0 1
1 1 0 0	0 1 0 0	0 0 0 0

Problem Definition

- N_s: number of examples
- L_s: length of each example
- Random (optimal) example
  - Start at random state
  - Follow π^* for L_s
- Goal: state features ⇒ R or V
  - Use R or V to recover π^*



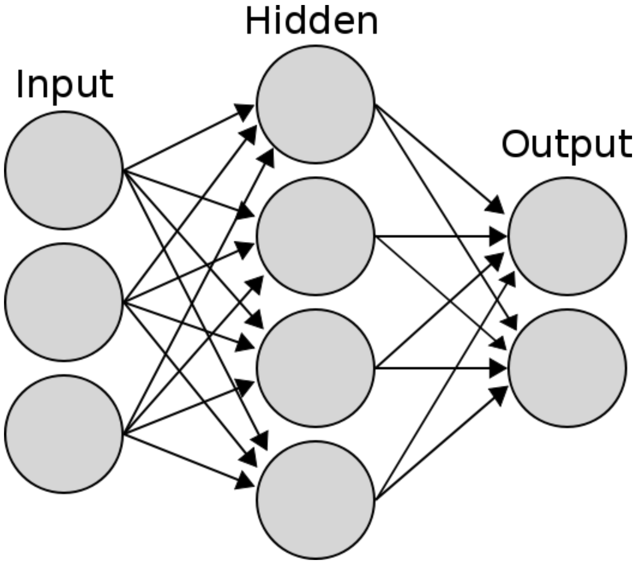
Genetic Algorithms (GA)

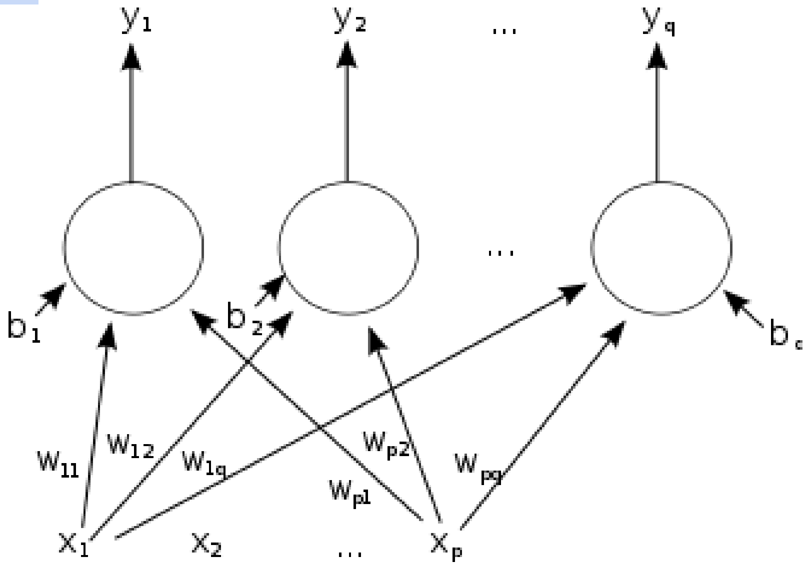
how good is this gene?

- ➔ Inspired by the biological genomes
- ➔ Evaluate genomes based on *fitness function*

(Artificial) Neural network (ANN / NN)

- ➔ Inspired by biological neural networks
- ➔ Function approximation model





Neuroevolution of Augmented Topologies (NEAT)

- ➔ Encode an *NN* explicitly (*w*,*b*) as a genome
- ➔ Use *GA* to tweak weights and connections
- ➔ Begins with relatively simple NN
- ➔ *NN gains complexity* based on fitness requirement

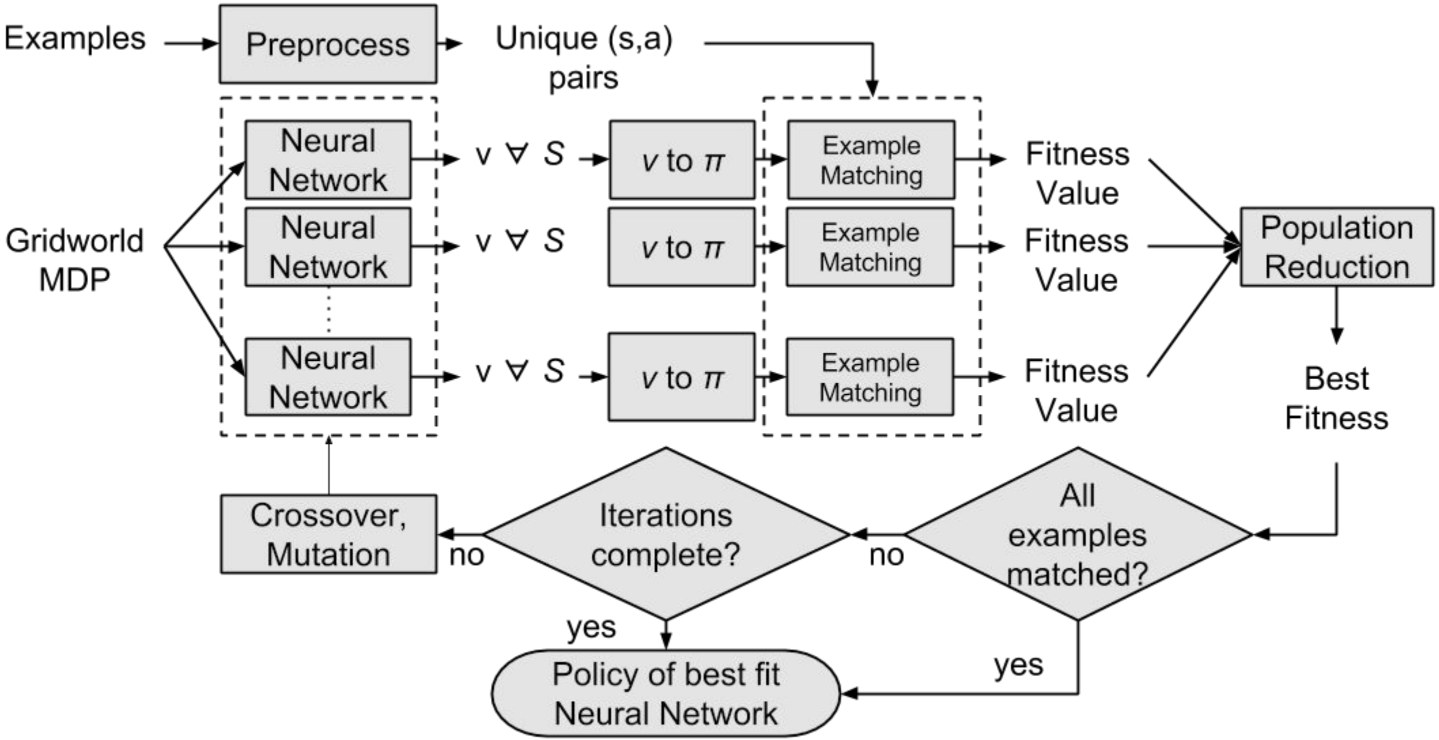
what fitness function do we use?

Proposed Method: *NEAT-IRL*

contribution I

- ➔ NN input: state features
- ➔ NN output: state value
- ➔ Use policy match based fitness function
  - ◆ Cosine of angle between policy action and optimal action directions
  - ◆ Accumulate over all example states
- ➔ Algorithm terminates when all examples matched
  - ◆ Also limited by N_G
- ➔ *Existing work uses rule-based learning*
  - ◆ Excluded from comparison

Proposed Method: *NEAT-IRL*

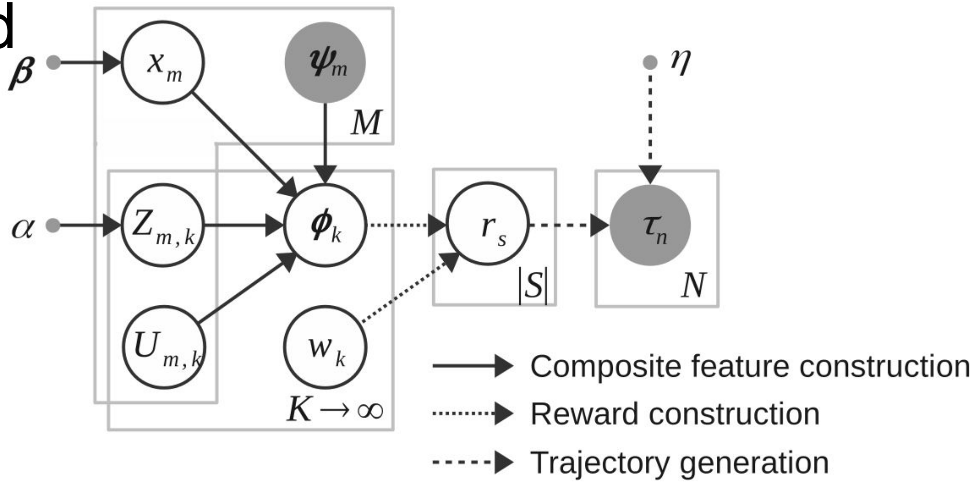


FIRL, GPIRL vs *NEAT-IRL*

- ➔ FIRL, GPIRL focus on reward matching
- ➔ FIRL assumes linear combination of features
- ➔ NEAT-IRL generates state values (not state rewards)
  - ◆ State values surface smoother than state rewards
  - ◆ Values to policy easy: greedy action selection
- ➔ GP models can be optimized to fit data exactly
  - ◆ Easier to overfit

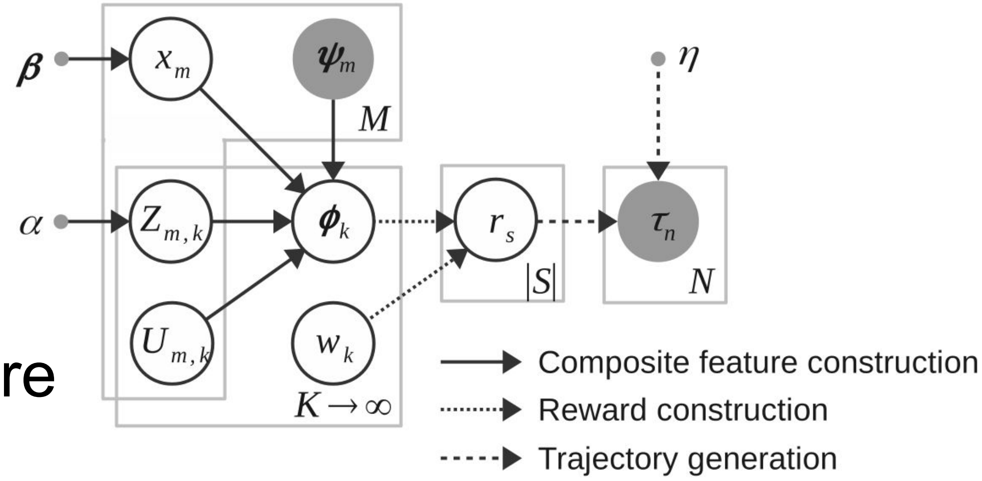
BNP-FIRL(MAP, mean)

- ➔ Reward (r) = product of composite features (ϕ) and associated weights (w)
- ➔ ψ: Original state features
- ➔ X: atomic features used to form composite features



BNP-FIRL(MAP, mean)

- ➔ *Z*: atomic features used to build composite features
- ➔ IBP used to estimate number and values of ϕ
- ➔ *U*: negation of boolean feature in composite feature
- ➔ *τ*: demonstrations



BNP-FIRL(MAP, mean)

- ➔ $P(r|\tau)$ is iteratively maximized
- ➔ Store data over all iterations
- ➔ MAP based result
  - ◆ Iterative results used to compute Maximum A-Posteriori (MAP)
  - ◆ Use empirical observations to estimate unobserved quantity
  - ◆ *Inferior to mean based result in our setting*
- ➔ Mean based result
  - ◆ Compute $r = w.\phi$ per iteration
  - ◆ Result = sum of r over all iterations

what we do with this data matters

Proposed Method: *BNP-FIRL(NEAT)*

contribution II

- ➔ Dimensions of w and ϕ vary across iterations
- ➔ Dimensions of r are constant across iterations
  - ◆ Use as input to NN
  - ◆ Output of NN is state reward

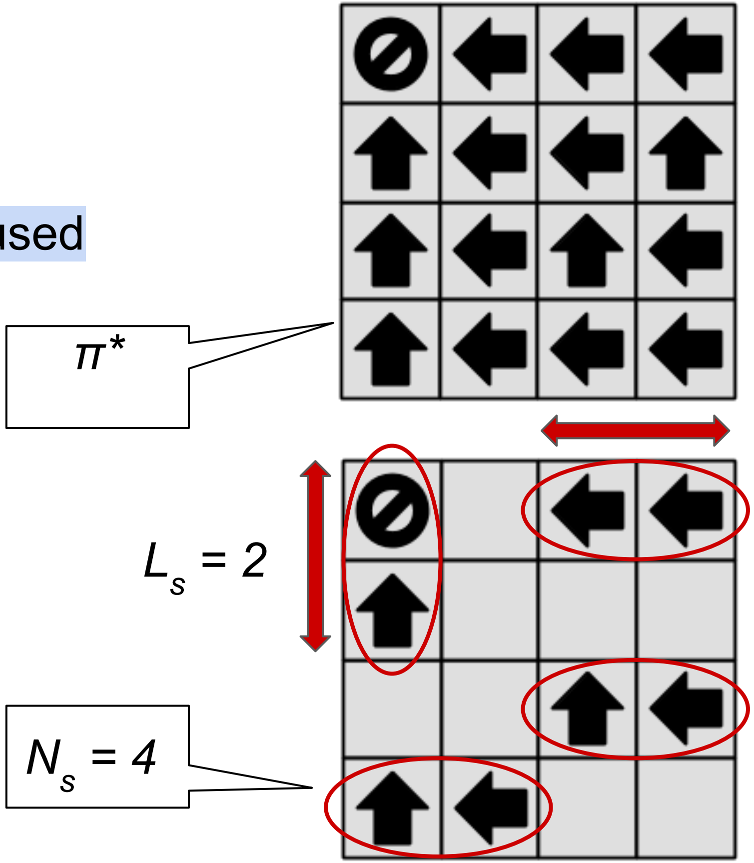
BNP-FIRL(mean) vs *BNP-FIRL(NEAT)*

- ➔ BNP-FIRL(mean) uses linear combination of *r* values
- ➔ Non-linear combination expected to perform better
  - ◆ More powerful expression of variable relationships
- ➔ BNP-FIRL(NEAT) increases algorithm parameters
  - ◆ N_P, N_G

Evaluation

➔ Experimental setup

- Scaled values of IRL toolkit setup used
- *16x16* grid world: $N_S = 8$, $L_S = 4$
  - Used for primary experiments
- *4x4* grid world: $N_S = 4$, $L_S = 1$
  - Used for MDP based analysis
- Averages over *25* executions



Evaluation

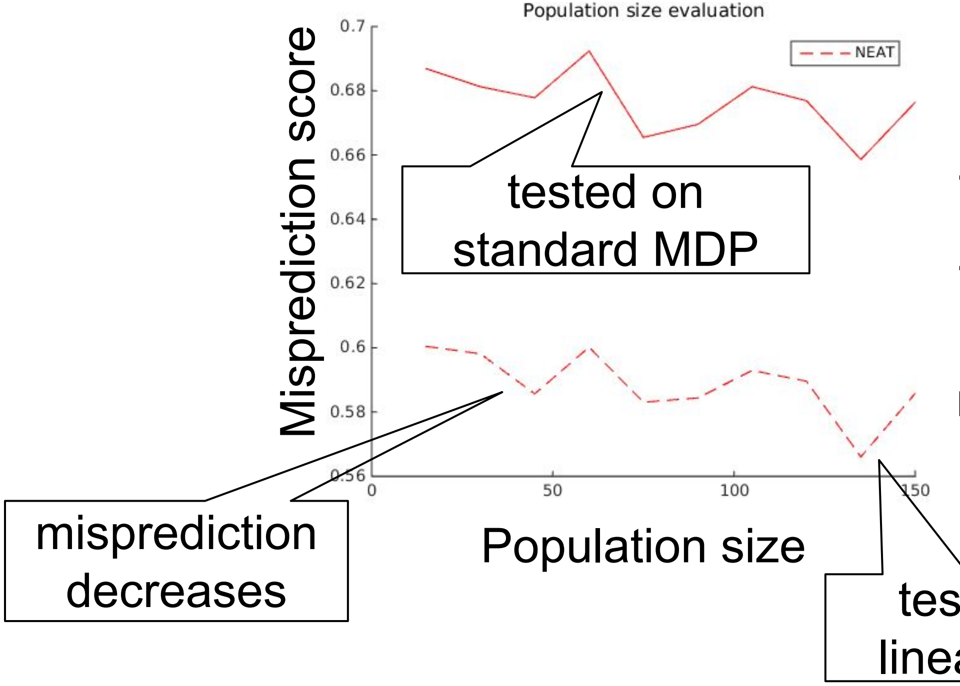
- NEAT parameters
  - $N_P = 150, N_G = 200$
    - Standard values from NEAT implementation
    - Used for isolated testing of NEAT-IRL
  - $N_P = 50, N_G = 50$
    - For comparison experiments, arbitrary
- Algorithms compared
  - GPIRL > FIRL > other popular IRL algorithms
  - Compare GPIRL, BNP-FIRL(mean), *NEAT-IRL* and *BNP-FIRL(NEAT)*

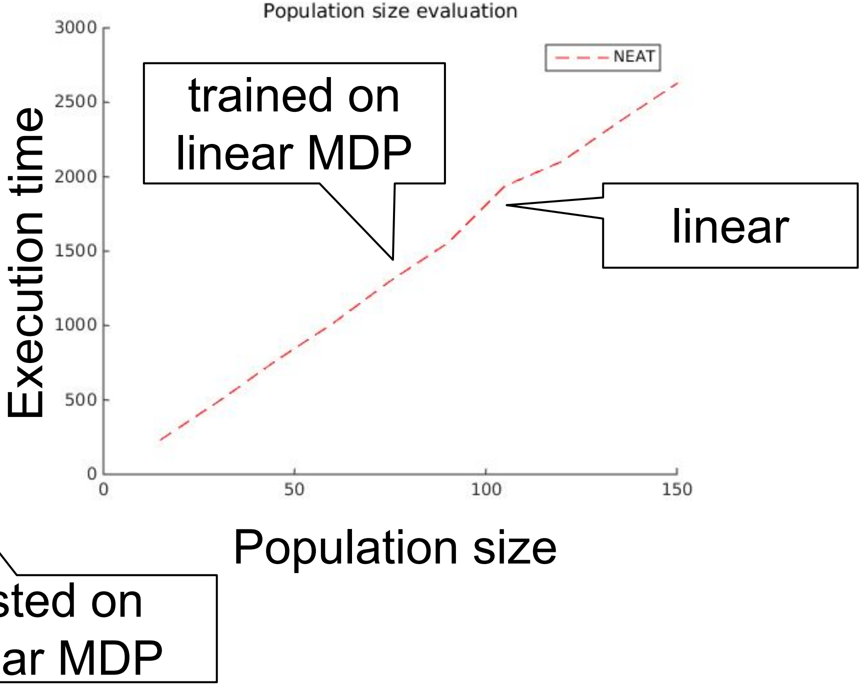
faster IRL completion

Evaluation

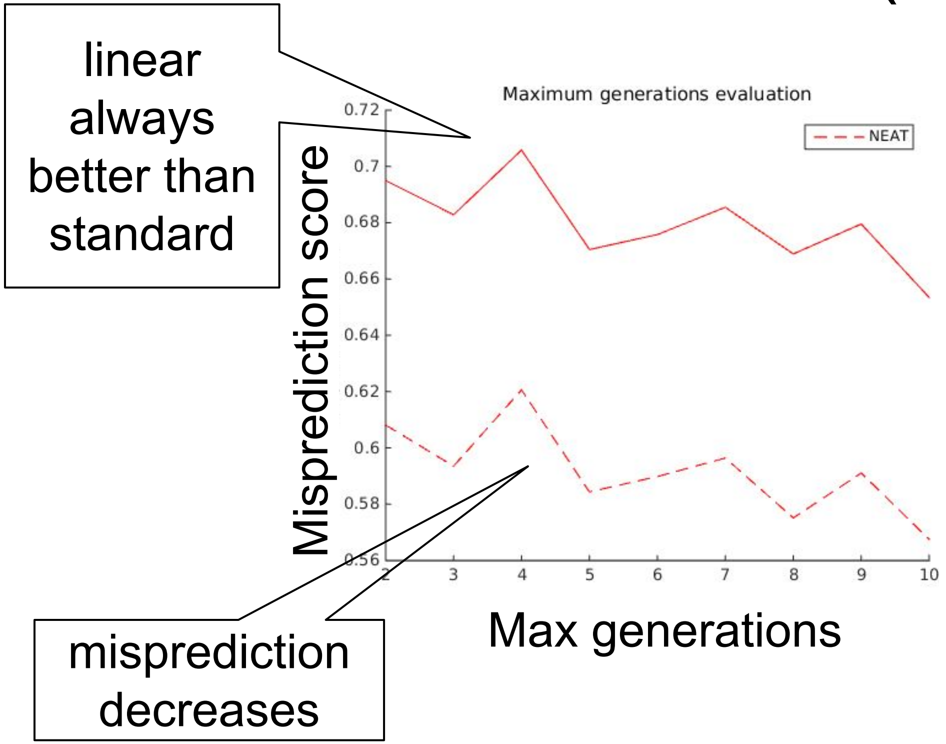
- ➔ Misprediction
  - ◆ Percentage of actions over all states predicted incorrectly
- ➔ Graph notation
  - ◆ Dotted line used to represent linear MDP data
    - *Linearly solvable* MDP
    - Cost function for *active* vs *passive* action is *convex*
  - ◆ Solid line used to represent standard MDP data

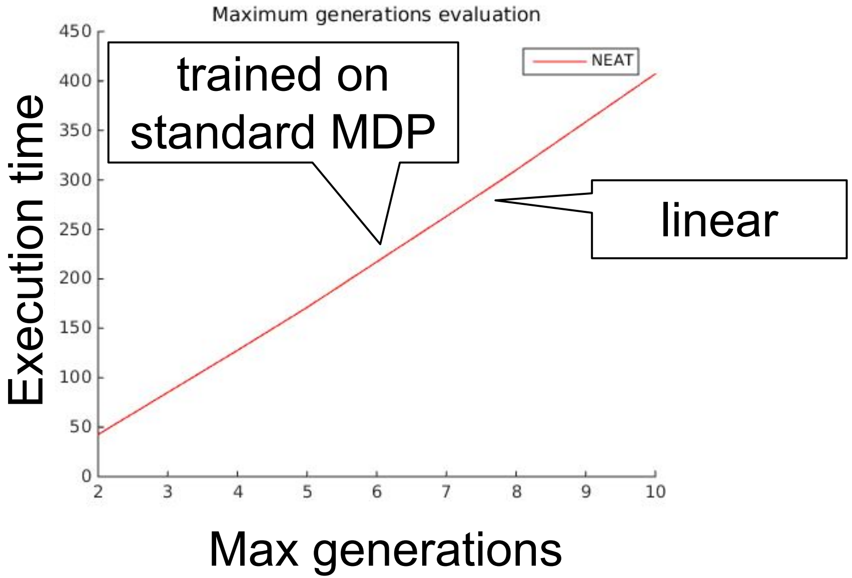
Evaluation: NEAT-IRL (N_P varied, N_G= 50)





Evaluation: NEAT-IRL (N_G varied, N_P = 150)



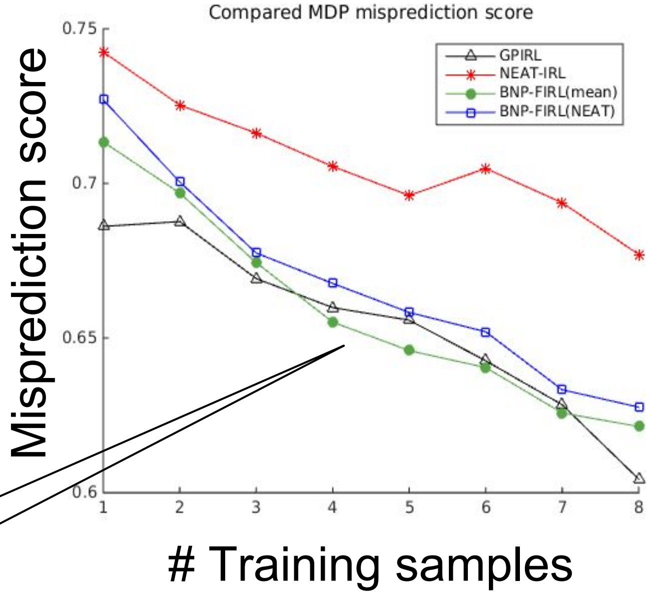


Evaluation

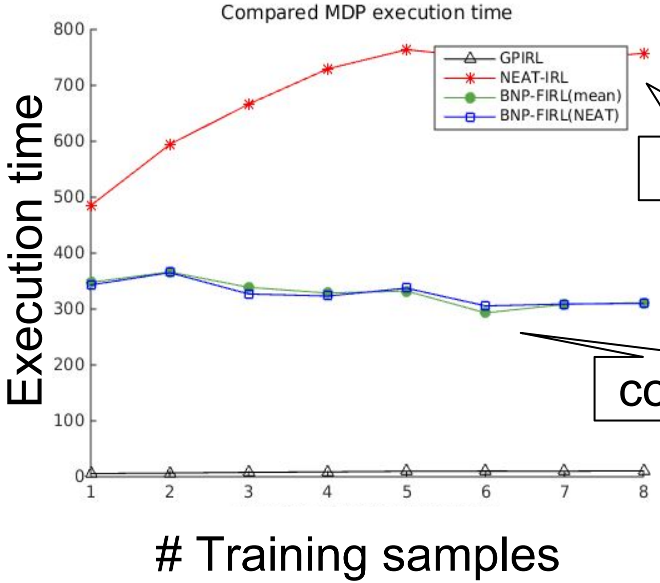
- ➔ GPIRL, BNP-FIRL(mean) vs NEAT-IRL, BNP-FIRL(NEAT)
  - ◆ Compared on standard vs linear MDP
  - ◆ Compared on deterministic vs non-deterministic MDP
    - Determinism (*d*) = *1.0, 0.7* respectively

Evaluation: Standard MDP, d = 0.7

➔ GPIRL, BNP-FIRL(mean) vs NEAT-IRL, BNP-FIRL(NEAT)



competitive

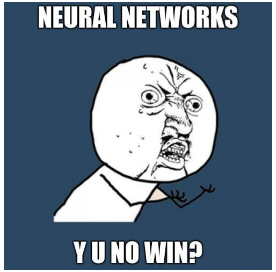


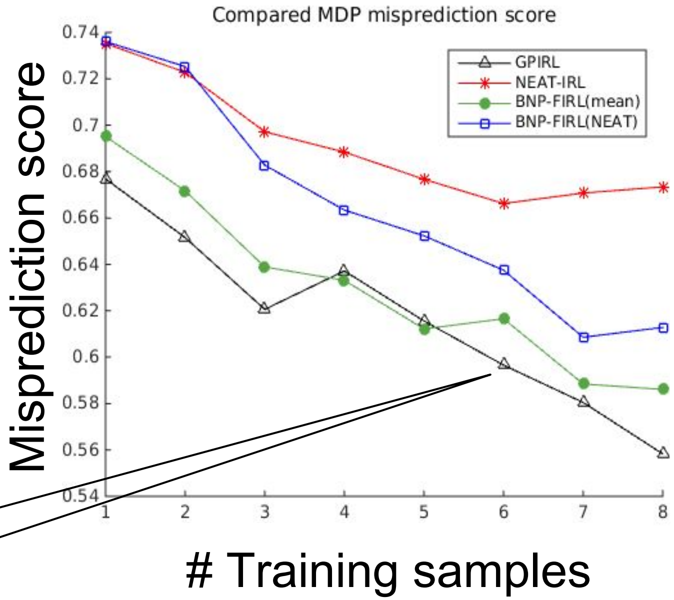
saturates

competitive

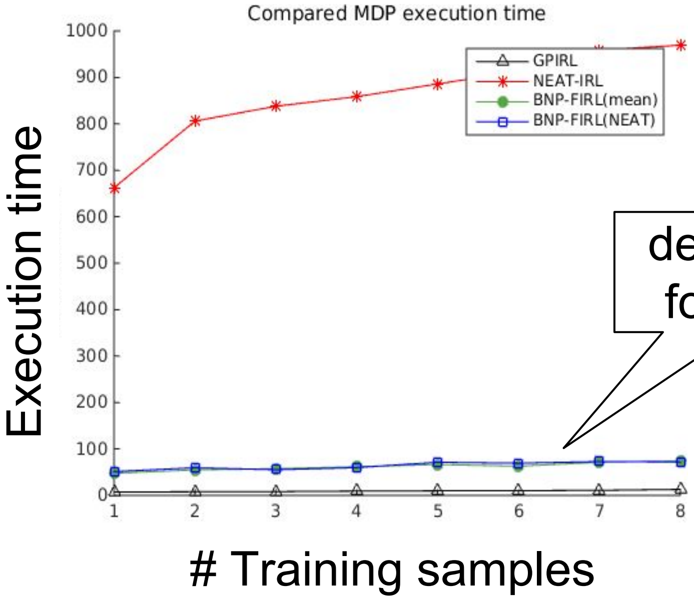
Evaluation: Standard MDP, d = 1.0

➔ GPIRL, BNP-FIRL(mean) vs NEAT-IRL, BNP-FIRL(NEAT)





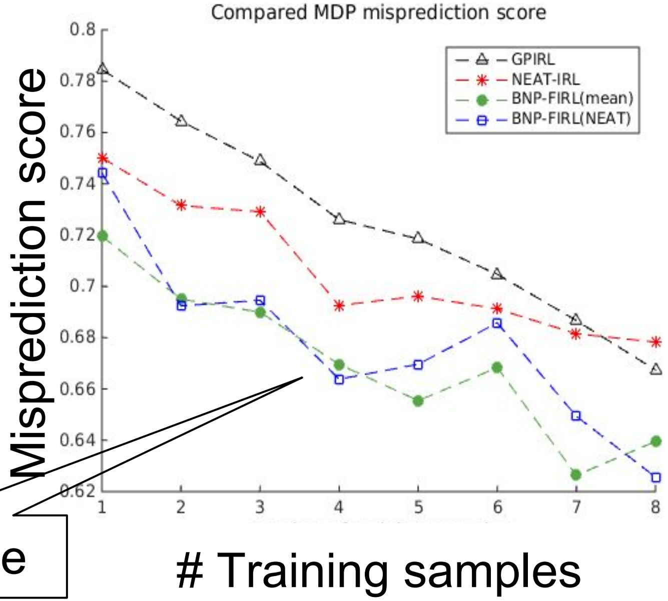
GPIRL is better



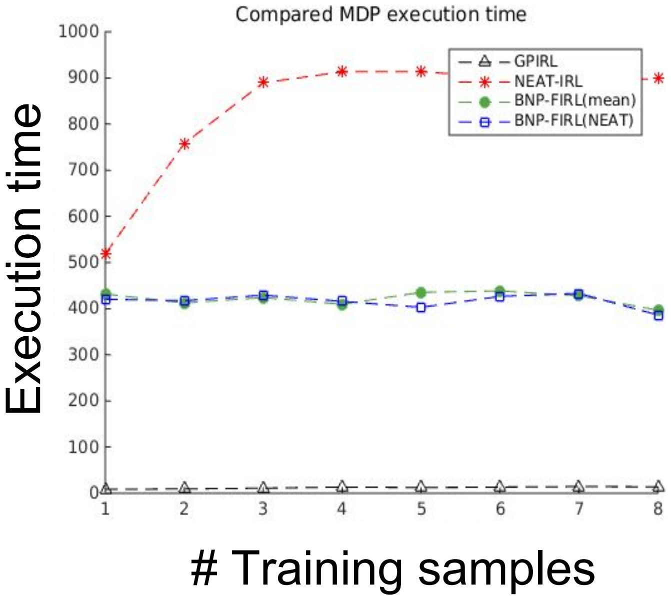
decreased for *d=1.0*

Evaluation: Linear MDP, d = 0.7

➔ GPIRL, BNP-FIRL(mean) vs NEAT-IRL, BNP-FIRL(NEAT)

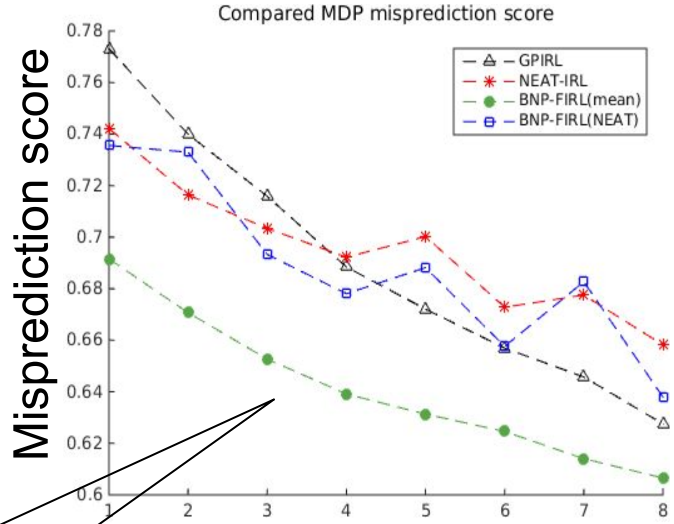


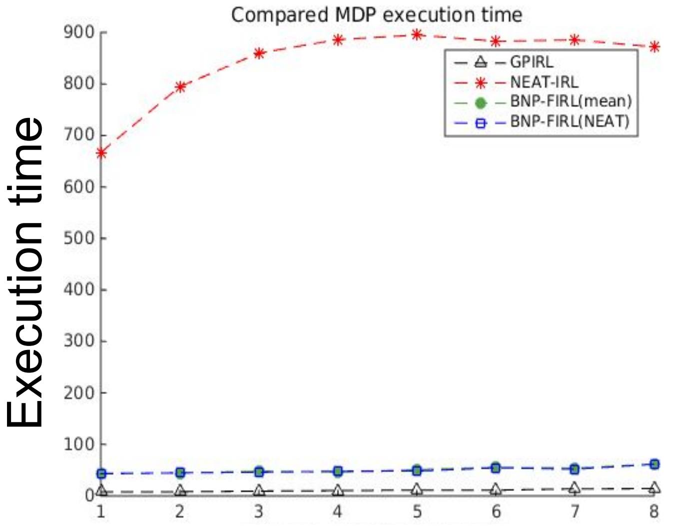
competitive



Evaluation: Linear MDP, d = 1.0

➔ GPIRL, BNP-FIRL(mean) vs NEAT-IRL, BNP-FIRL(NEAT)





BNP-FIRL(mean) is better

Training samples

Training samples

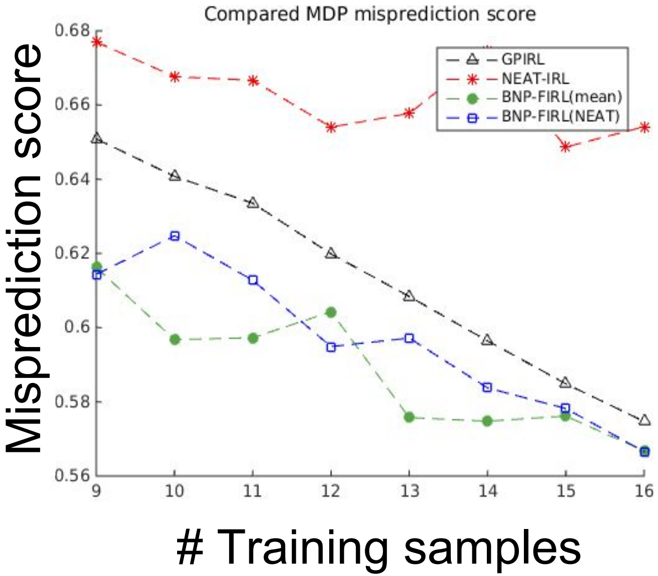
Evaluation

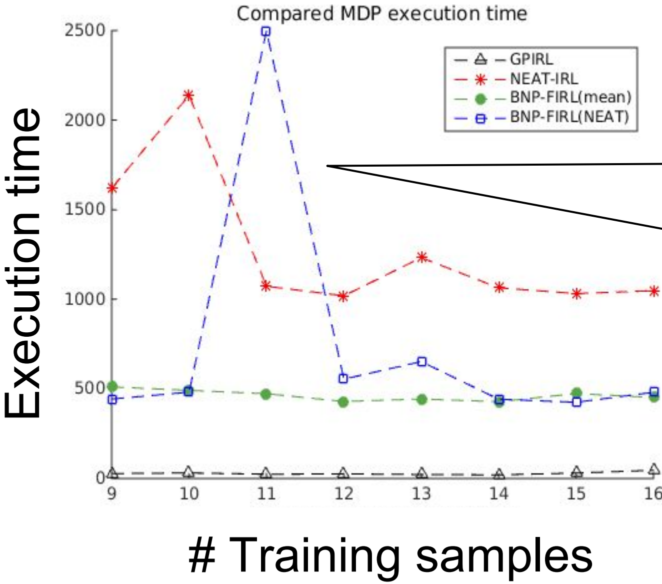
- ➔ BNP-FIRL(mean) vs BNP-FIRL(NEAT)
  - ◆ Linear MDP, $d = 0.7$ is favourable
  - ◆ Extend graph data
    - Examine for extended N_S values
    - Make smoother by averaging *100* executions

Evaluation: Extended N_S (linear MDP, d = 0.7)

➔ BNP-FIRL(mean) vs BNP-FIRL(NEAT)

same trends

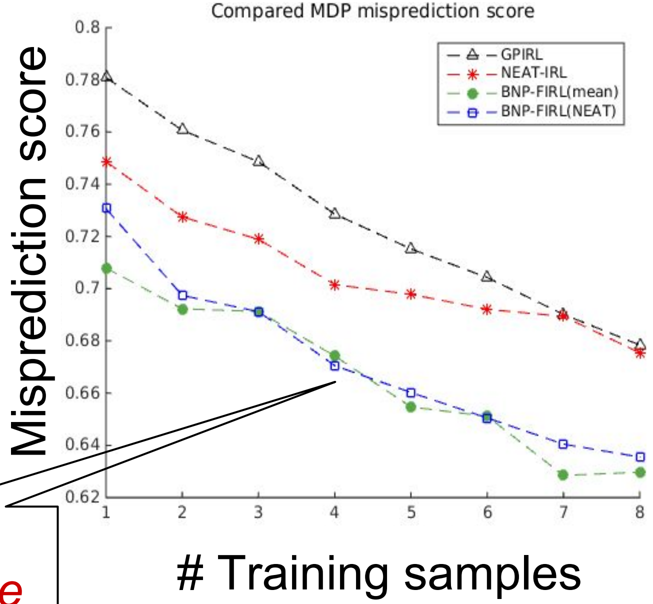




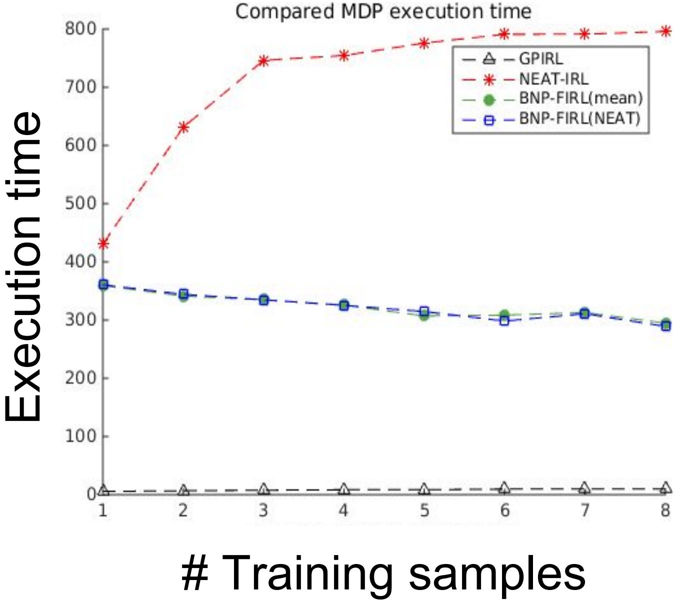
anomaly: average over more runs

Evaluation: 100 executions (linear MDP, d = 0.7)

➔ BNP-FIRL(mean) vs BNP-FIRL(NEAT)

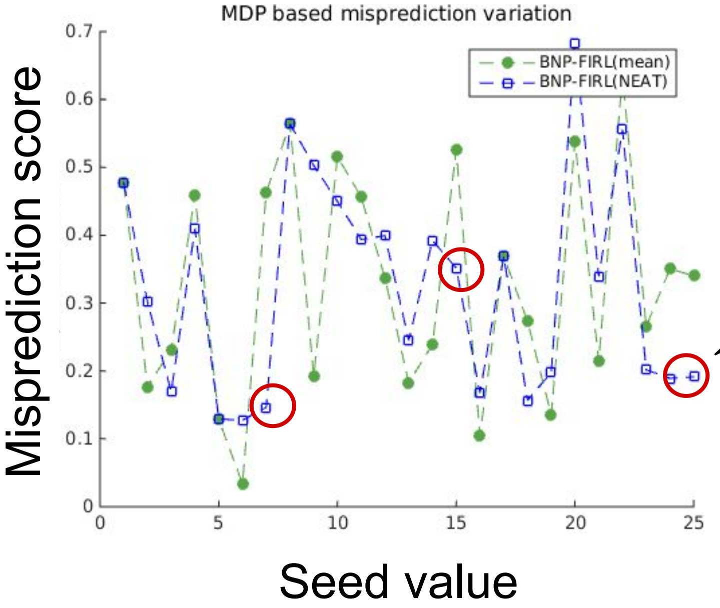


very competitive



Evaluation: Analyze MDPs (linear MDP, d = 0.7)

➔ BNP-FIRL(mean) vs BNP-FIRL(NEAT)

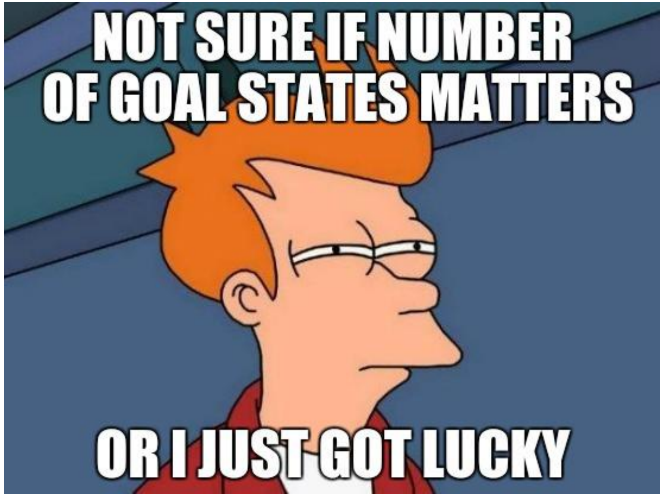


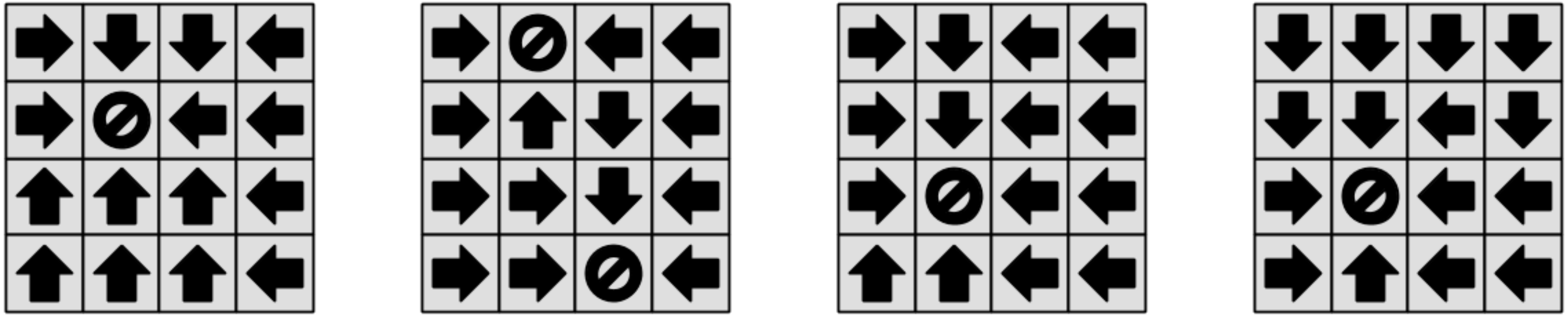
BNP-FIRL(NEAT) > BNP-FIRL(mean)

Evaluation

➔ BNP-FIRL(mean) vs BNP-FIRL(NEAT)

- Analyze optimal policies
- *Hypothesize that BNP-FIRL(NEAT) is better for multiple goal states*





Evaluation

- ➔ Hypothesis intuition
  - Multiple goals ⇒ multi-peaked state reward surface
  - NEAT should help BNP-FIRL with complex *r* functions
- ➔ Hypothesis evaluation
  - Two tailed t-test
  - Results vary for smaller averages
    - Average over 1000 executions
  - *M* denotes misprediction value

Evaluation

➔ Hypothesis evaluation

variance decreases as #goals increases

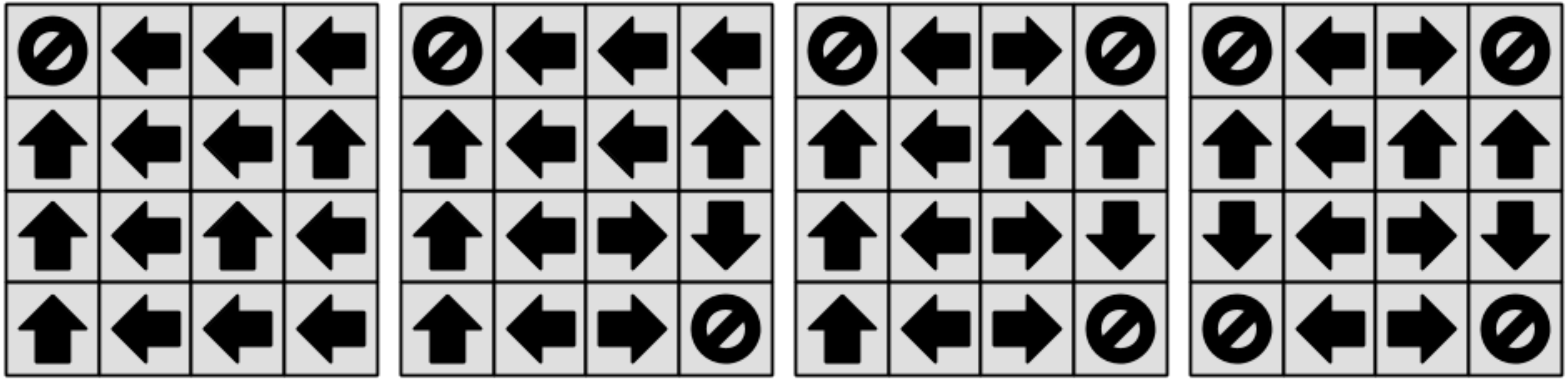
BNP-FIRL(mean) favourable

Number of Goals	Avg. $M_{BNP\text{-}FIRL(mean)}$	Avg. $M_{BNP\text{-}FIRL(NEAT)}$	p-value
1	0.2308	0.3392	3.8837 e-4
2	0.3292	0.3392	0.1870
3	0.4119	***0.33913***	***0.0063***
4	0.4954	***0.4674***	***3.9776 e-5***

BNP-FIRL(NEAT) favourable

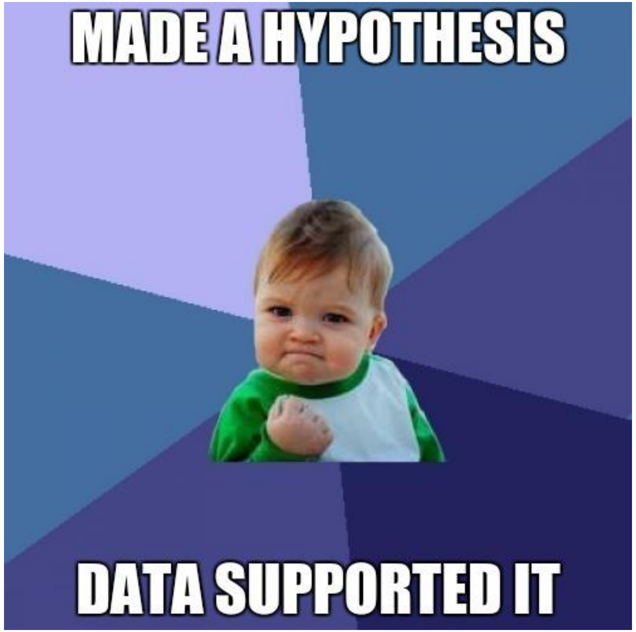
Evaluation

- ➔ Hypothesis evaluation: A closer look at MDPs
- ➔ Observe policy complexity in different cases



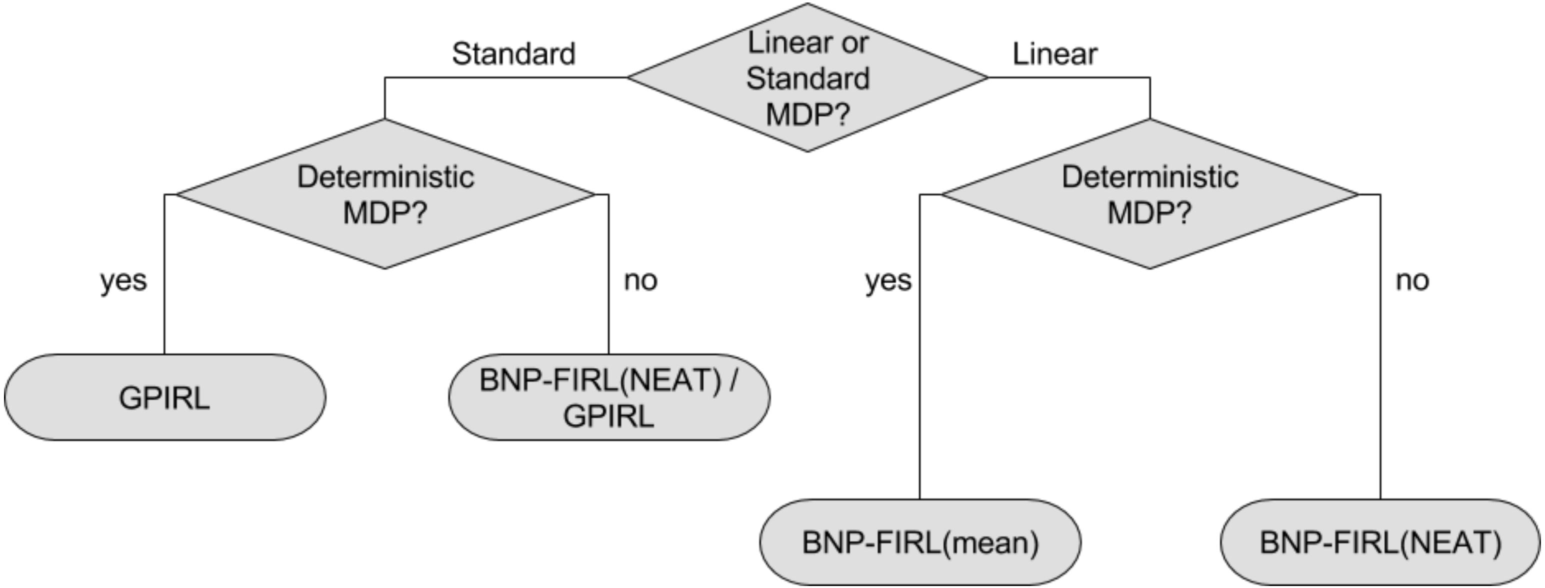
Evaluation

- ➔ Hypothesis evaluation
  - ◆ BNP-FIRL(NEAT) better for multiple goals!
  - ◆ Use of neural networks helps fit complex *r* functions
- ➔ Algorithm rating (linear MDP, $d = 0.7$)
  - ◆ BNP-FIRL(NEAT) > BNP-FIRL(mean) > NEAT-IRL > GPIRL



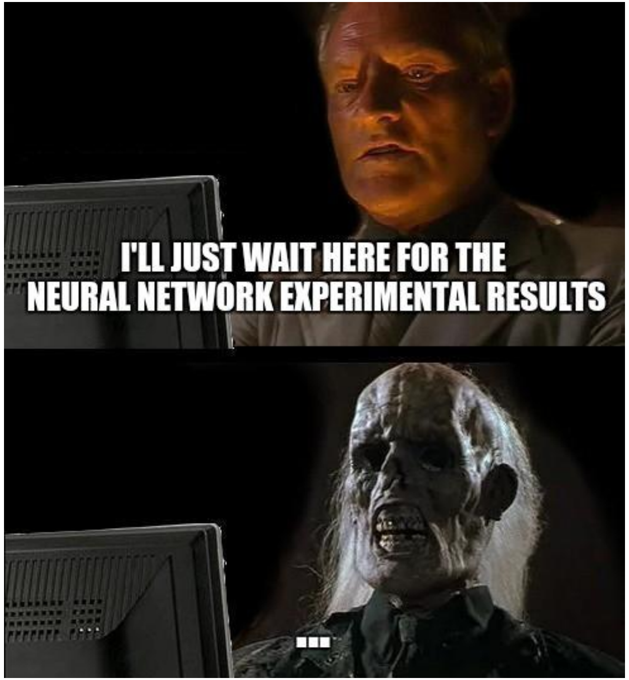
Evaluation

- ➔ Algorithm decision tree
  - ◆ Based on experimental results



Future Work

- ➔ NEAT parameters currently arbitrary
  - Can be tuned for a set of MDPs
- ➔ NEAT-IRL computationally inefficient
  - Learn from BNP-FIRL(NEAT)
  - Further computation parallelization
  - Use GPU computing
  - Limit policy prediction to example states
- ➔ Multiple agent setting
  - Incorporate information sharing at a cost



Selected References

- Choi, Jaedeug, and Kee-Eung Kim. "Bayesian nonparametric feature construction for inverse reinforcement learning." *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*. AAAI Press, 2013.
- Hahn, Jurgen, and Abdelhak M. Zoubir. "Inverse Reinforcement Learning using Expectation Maximization in mixture models." *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015.
- Levine, Sergey, Zoran Popovic, and Vladlen Koltun. "Feature construction for inverse reinforcement learning." *Advances in Neural Information Processing Systems*. 2010.
- Levine, Sergey, Zoran Popovic, and Vladlen Koltun. "Nonlinear inverse reinforcement learning with gaussian processes." *Advances in Neural Information Processing Systems*. 2011.
- Michini, Bernard, and Jonathan P. How. "Bayesian nonparametric inverse reinforcement learning." *Machine Learning and Knowledge Discovery in Databases*. Springer Berlin Heidelberg, 2012. 148-163.
- Yong, Chern Han, et al. "Incorporating Advice into Neuroevolution of Adaptive Agents." *AIIDE*. 2006.
- Karpov, Igor V., Vinod K. Valsalam, and Risto Miikkulainen. "Human-assisted neuroevolution through shaping, advice and examples." *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 2011.

Conclusion

- ➔ NN based IRL is good for non-deterministic linear MDP
- ➔ Better at understanding non-linear reward functions
- ➔ *BNP-FIRL(NEAT)* > BNP-FIRL(mean) > *NEAT-IRL* > GPIRL