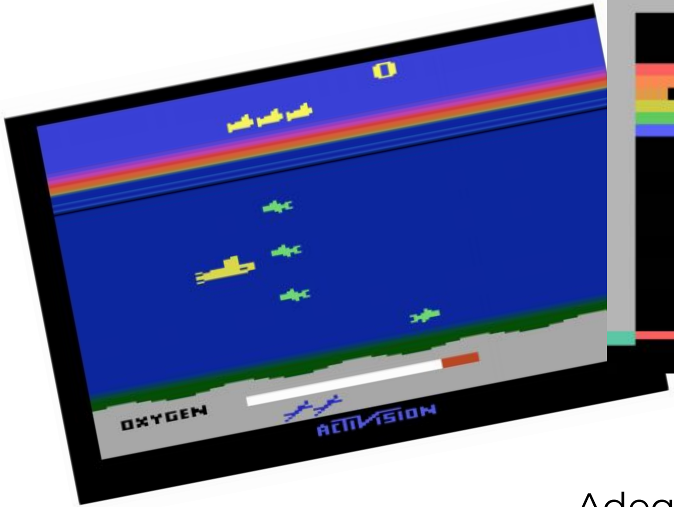
Neuro-algorithmic Policies Enable Fast Combinatorial Generalization

Marin Vlastelica, Michal Rolínek and Georg Martius

Max Planck Institute for Intelligent Systems

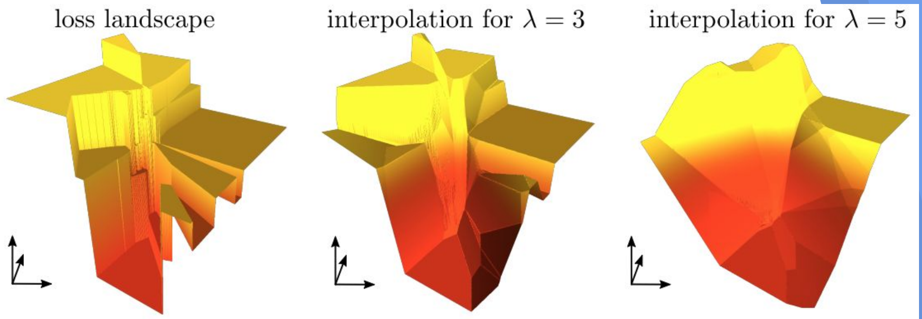
Generalizing in Control is Hard



Adequate priors are necessary

Our Contribution

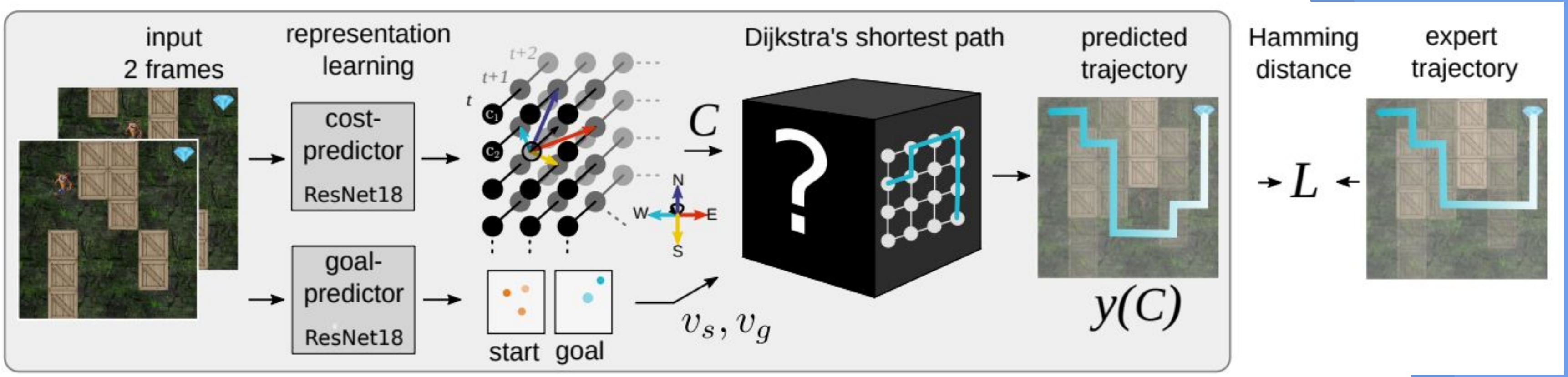
- We formulate the discrete control problem as a time-dependent shortest path problem (TDSP).
- We apply **blackbox differentiation theory** [1] to embed TDSP solvers into neural network architectures.
- We show that these **neuro-algorithmic policies** surpass standard methods for imitation learning as well as RL baselines in terms of generalization capability.



Assumptions on MDP

- State space factorizes
- We apply **blackbox differentiation theory** [1] to embed TDSP solvers into neural network architectures.
- We show that these **neuro-algorithmic policies** surpass standard methods for imitation learning as well as RL baselines in terms of generalization capability.

Neuro-algorithmic Policy Architecture



How do We Train Them?

Algorithm 1 Forward and backward Pass for the shortest-path algorithm

---

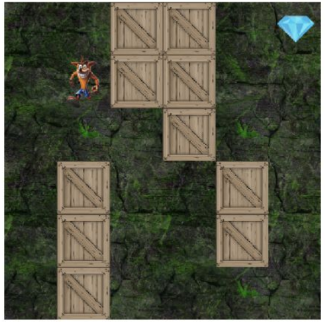
function FORWARDPASS(C, s, e)
  $Y := \textbf{TDSP}(C, s, e)$ *// Run Dijkstra's algorithm*
  save Y, C, s, e *// Needed for backward pass*
  return Y

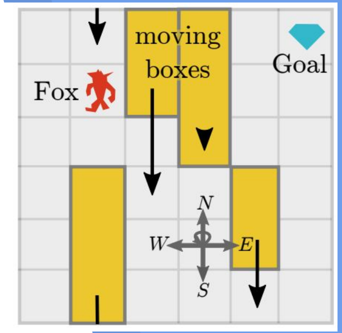
function BACKWARDPASS($\nabla L(Y)$, λ)
  load Y, C, s, e
  $C_\lambda := C + \lambda \nabla L(Y)$ *// Calculate modified costs*
  $Y_\lambda := \textbf{TDSP}(C_\lambda, s, e)$ *// Run Dijkstra's algo.*
  return $\frac{1}{\lambda}\left(Y_\lambda - Y\right)$

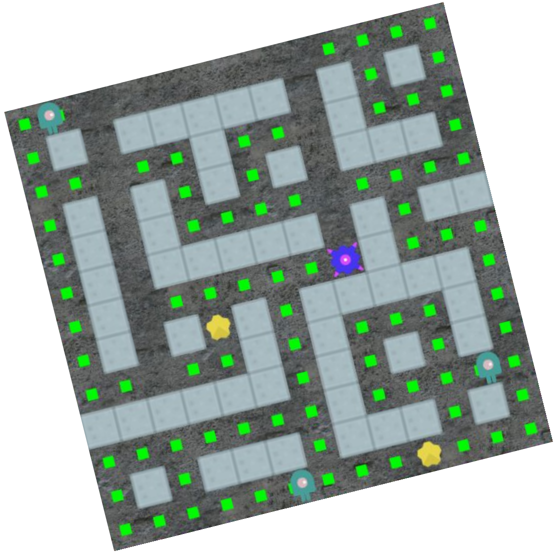
Combinatorially Challenging Environments

We evaluate on environments that are combinatorially challenging.

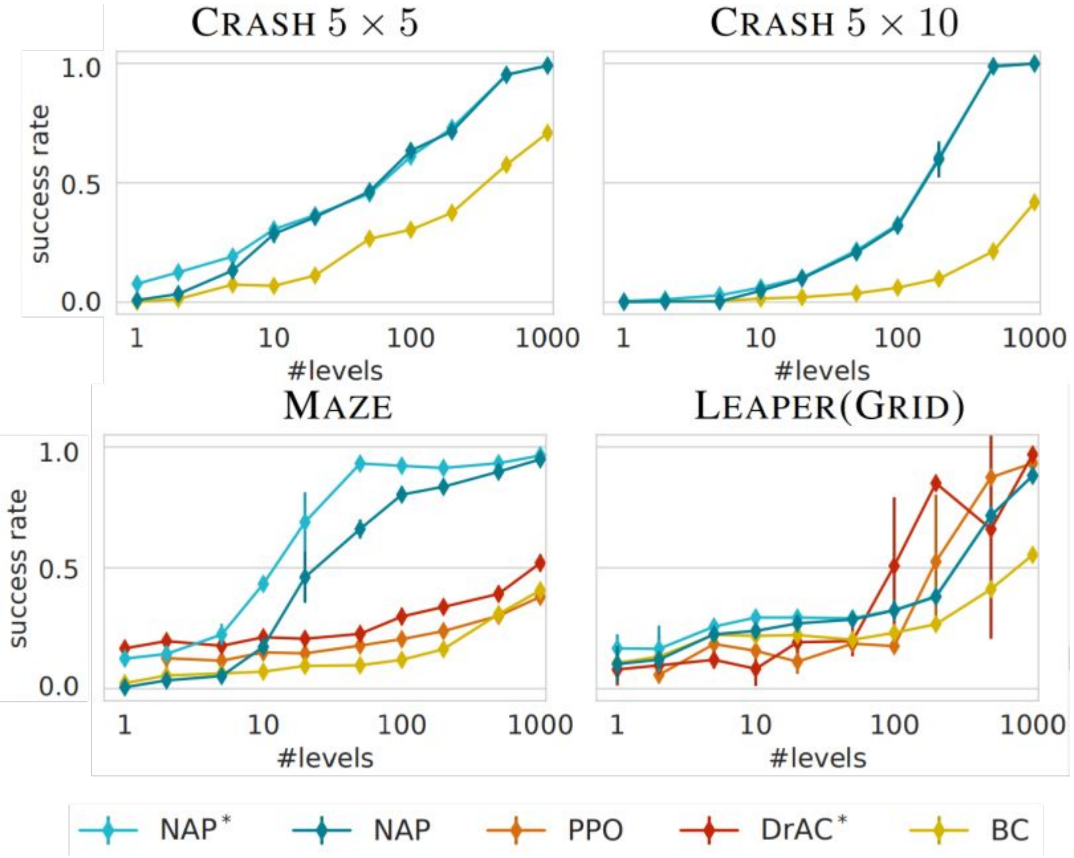
We want to generalize to unseen levels with few examples.





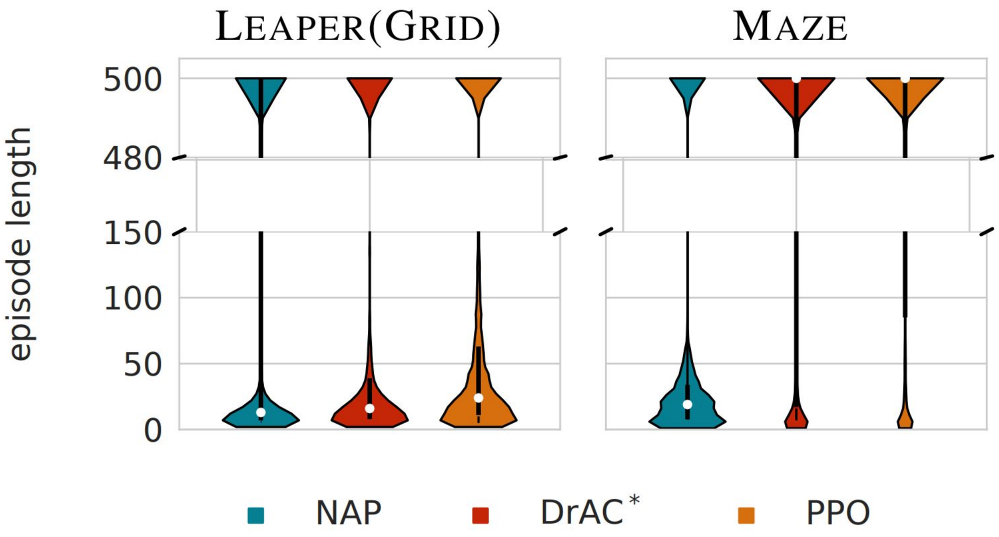


Results



CRASH 5 × 5
CRASH 5 × 10
MAZE
LEAPER(GRID)
success rate
1.0
0.5
0.0
1
10
100
1000
#levels
NAP*
NAP
PPO
DrAC*
BC

Path Length


LEAPER(GRID)
MAZE
episode length
500
480
150
100
50
0
NAP
DrAC*
PPO

Thank You