Scaling up and Stabilizing Differentiable Planning with Implicit Differentiation

Linfeng Zhao[1], Huazhe Xu[2,3], Lawson Wong[1]

[1] Khoury College of Computer Sciences, Northeastern University
[2] Institute for Interdisciplinary Information Sciences, Tsinghua University
[3] Shanghai Qi Zhi Institute

Northeastern University Khoury College of Computer Sciences

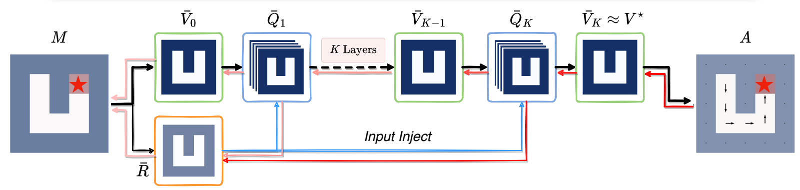
Implicit differentiation helps Differentiable Planning algorithms *scale up* in training and *stabilize* in convergence to the fix point of Bellman equation.

http://lfzhao.com/IDPlan

Scan to Visit the website

1. Motivation

Path Planning: Find shortest path to the goal location (red star)



Issue: Forward and Backward passes are coupled together

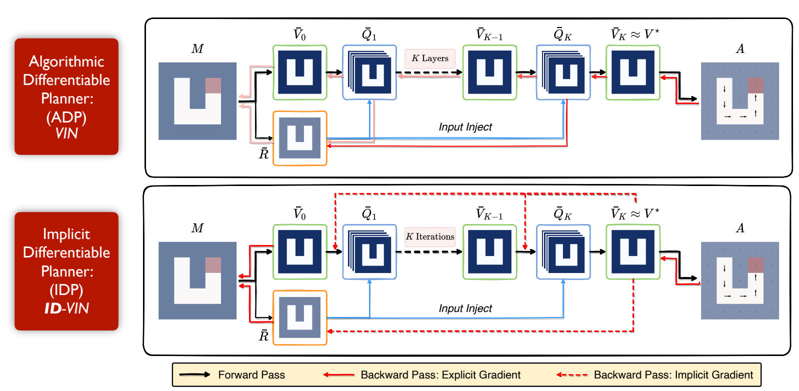
Differentiable planning algorithms, such as Value Iteration Network (VIN), typically need to *differentiate through the forward iteration process* — so the *backward gradient pass* is ***coupled*** with *forward iteration layers*. ***Can we decoupled this process?*** We use **implicit differentiation** through the fixed point directly.

2. Implicit Differentiation

Suppose $v^\star$ is the fixed point, x is arbitrary input, f is a Bellman operator.
The Bellman equation provides an equality constraint and has a fixed point.
Iteratively applying Bellman operators converges to a fixed point.
We can ***differentiate through the fixed point equation***, skipping forward layers.

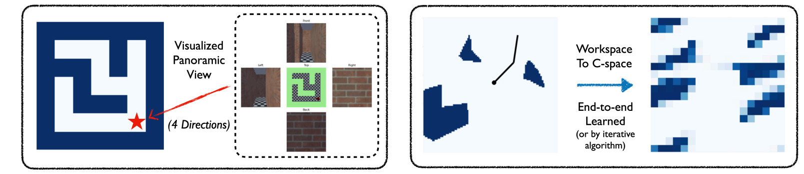
- Bellman equation: $$v^\star = f(v^\star, x)$$
- Differentiating both sides: $$\frac{\partial v^\star(\cdot)}{\partial(\cdot)} = \frac{\partial f(v^\star(\cdot), x)}{\partial(\cdot)} = \frac{\partial f(v^\star, x)}{\partial v^\star}\frac{\partial v^\star(\cdot)}{\partial(\cdot)} + \frac{\partial f(v^\star, x)}{\partial(\cdot)}$$ $$\frac{\partial \ell}{\partial(\cdot)} = \frac{\partial \ell}{\partial v^\star}\frac{\partial v^\star(\cdot)}{\partial(\cdot)} = \frac{\partial \ell}{\partial v^\star}\left(I - \frac{\partial f(v^\star, x)}{\partial v^\star}\right)^{-1}\frac{\partial f(v^\star, x)}{\partial(\cdot)}$$
- Solving backward fixed-point: $$w^\top \triangleq \frac{\partial \ell}{\partial v^\star}\left(I - \frac{\partial f(v^\star, x)}{\partial v^\star}\right)^{-1}; \quad w^\top = w^\top\frac{\partial f(v^\star, x)}{\partial v^\star} + \frac{\partial \ell}{\partial v^\star}.$$

3. Pipeline: Implicit Differentiable Planner



VIN with algorithmic differentiation needs to differentiate through the ***long computation graph***. We propose to use ***implicit differentiation*** to compute the gradient at an estimated fixed point $V_K \approx V^*$.

4. Environments

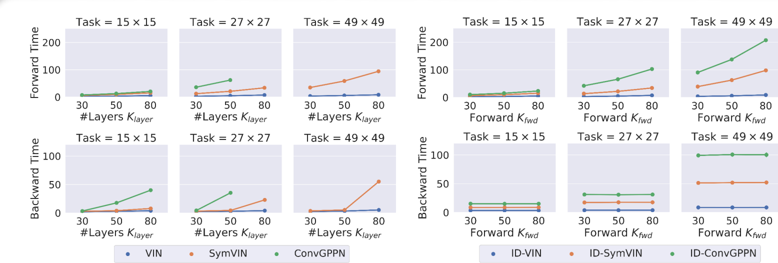


2D and Visual Maze Navigation

2-DOF Manipulation In Workspace and C-space

We experiment on 2D path planning in 4 different navigation and manipulation tasks. We use ***given*** 2D grid map or ***learned*** map (visual navigation and workspace manipulation, using a mapper module).

5. Performance: 2D Navigation Runtime

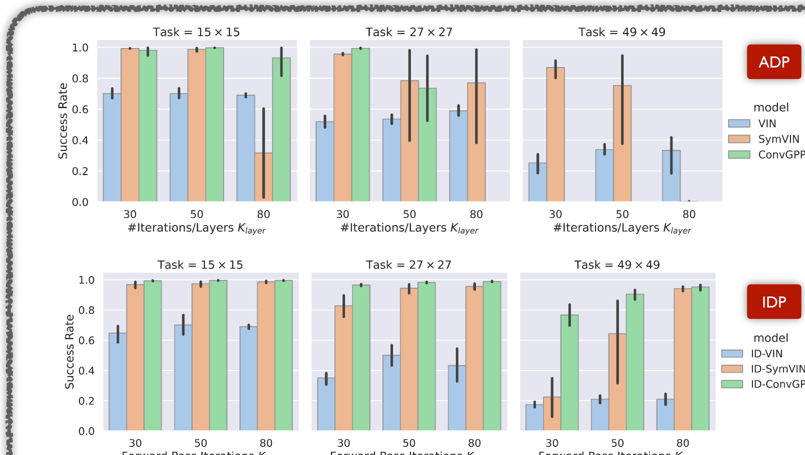


Algorithmic Differentiable Planners

Implicit Differentiable Planners

Algorithmic differentiable planners couple forward/backward passes. They have slow backward time and get out of memory for larger tasks/more iterations (10GB, thus missing dots). ***Implicit differentiable planners*** have same forward runtime but constant backward runtime and use less memory.

6. Performance: 2D Navigation Success Rate



Results on 3 sizes of 2D navigation. ***Algorithmic differentiable planners (ADPs)*** fail to converge for too many iterations. ***Implicit differentiable planners (IDPs)*** can successfully run and stably converge, and outperform counterparts.