MEMORY AUGMENTED CONTROL NETWORKS

Arbaaz Khan, Clark Zhang, Nikolay Atanasov, Konstantinos Karydis, Vijay Kumar, Daniel D. Lee
GRASP Laboratory, University of Pennsylvania

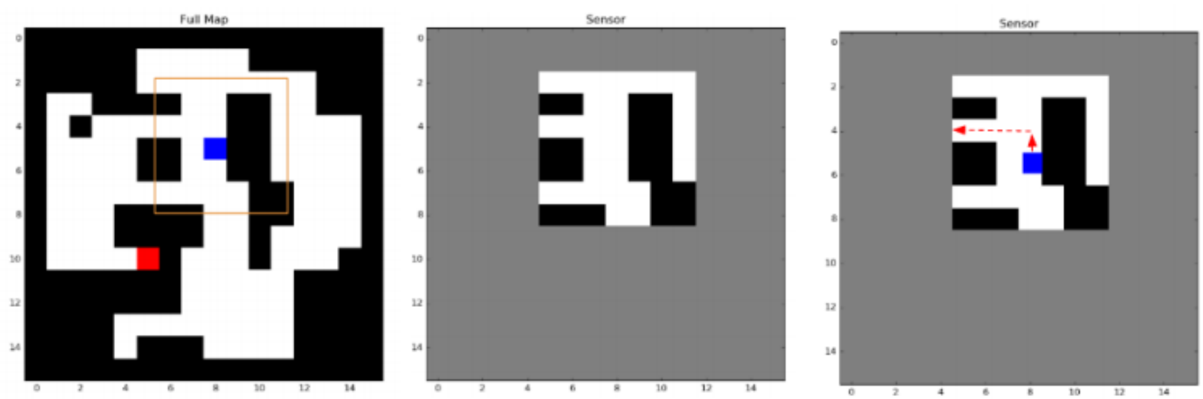
UNIVERSITY OF WATERLOO

Presented by Aravind Balakrishnan

Introduction

Partially observable environments with sparse rewards

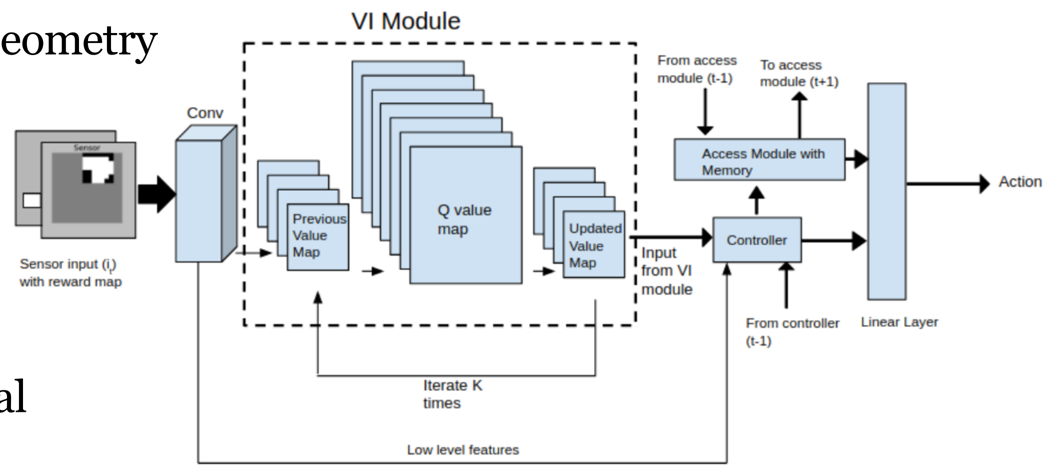
- Most real-world tasks
- Needs history of observations and actions



(a) 2D Grid World (b) Sensor observation. (c) Policy for sensor observation.

The solution - MACN

- **Differentiable Neural Computer (DNC)**
  - Neural network with differentiable external memory
  - maintains an estimate of the environment geometry



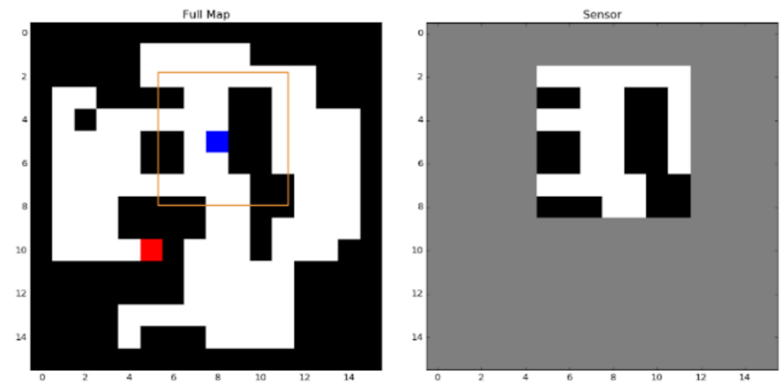
- **Hierarchical planning**
  - Lower level: Compute optimal policy on local observation
  - Higher level: Local policy + local environment features + map estimation to generate global policy

UNIVERSITY OF WATERLOO

Problem definition

- States: $s_t \in \mathcal{S}$, where $s \in \mathcal{S}^{\text{goal}}$ is the goal state
- Action: $a_t \in \mathcal{A}$
- **Map:** $m \in \{-1, 0\}^n$, -1 for tiles that are an obstacle
- **Local FOV:** $H(s) \in \mathbb{R}^{n \times n}$, 0 for non-observable tiles
- **Local observation:** $z_t = H(s_t)m$
- Information available to agent at time, t:

$$h_t := \left(s_{0:t}, z_{0:t}, a_{0:t-1}, \mathcal{S}^{\text{goal}}\right) \in \mathcal{H}$$



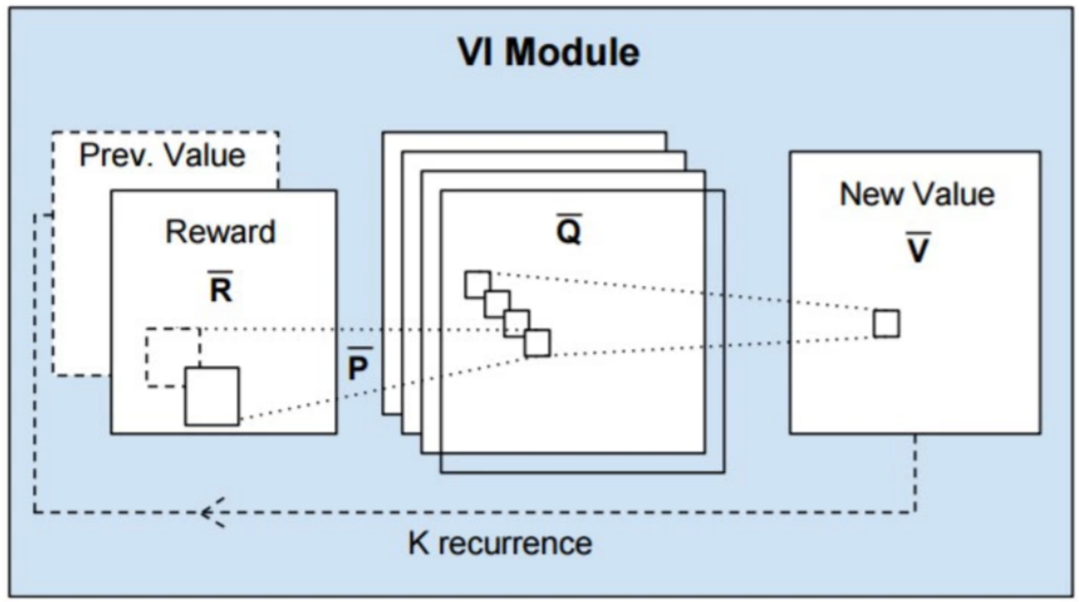
- **The problem:** Find mapping from z_t to action

Value Iteration Networks (VIN)

- Transition : $\mathcal{T}(h_t, a_t) = (h_t, s_{t+1} = f(s_t, a_t), z_{t+1} = H(s_{t+1})m, a_t)$
- Reward : $r(h_t, a_t) = z_t[s_t]$
- MDP : $\mathcal{M}(\mathcal{H}, \mathcal{A}, \mathcal{T}, r, \gamma)$

VIN: Value Iteration approximated by a Convolutional Neural Network:

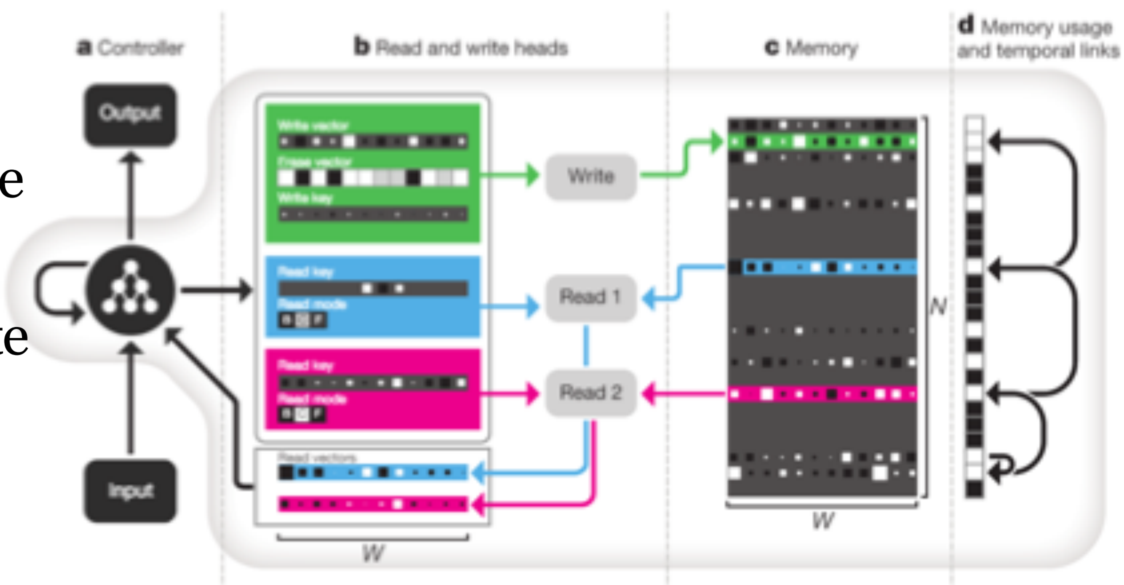
Previous value function stacked with reward, passed through a Conv layer, max-pooled along channel and repeated K times is an approximation of value iteration over K iterations



https://arxiv.org/abs/1602.02867

Differentiable Neural Computer (DNC)

- LSTM (controller) with an external memory
- Improved on Neural Turing Machine
- Uses differential memory attention mechanisms to selectively read/write to external memory, M

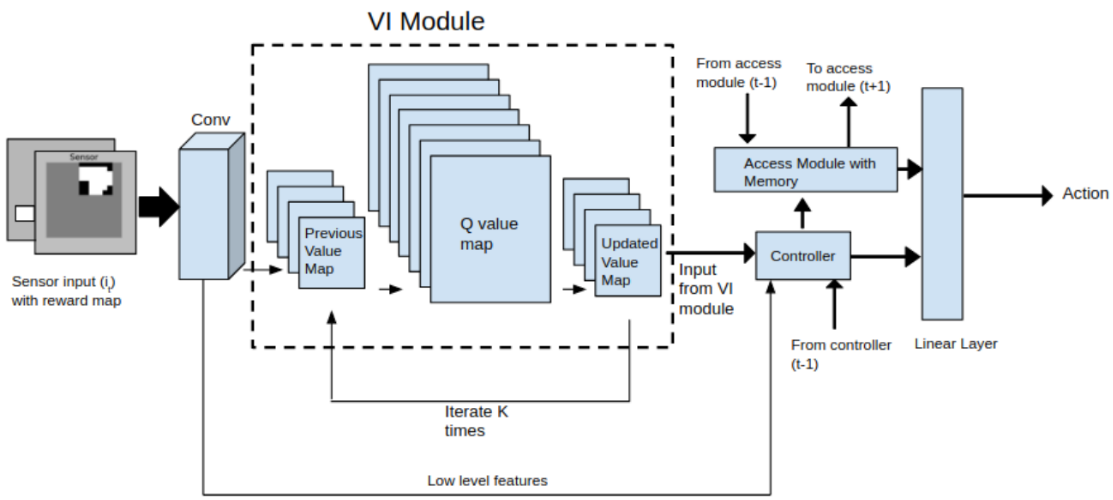


- **Read**: $re_t^i = M_t^\top w_t^{read,i}$
- **Write** $M_t = M_{t-1}(1 - w_t^W e_t^\top) + w_t^W v_t^\top$

https://www.nature.com/articles/nature20101.epdf

Architecture – Conv block

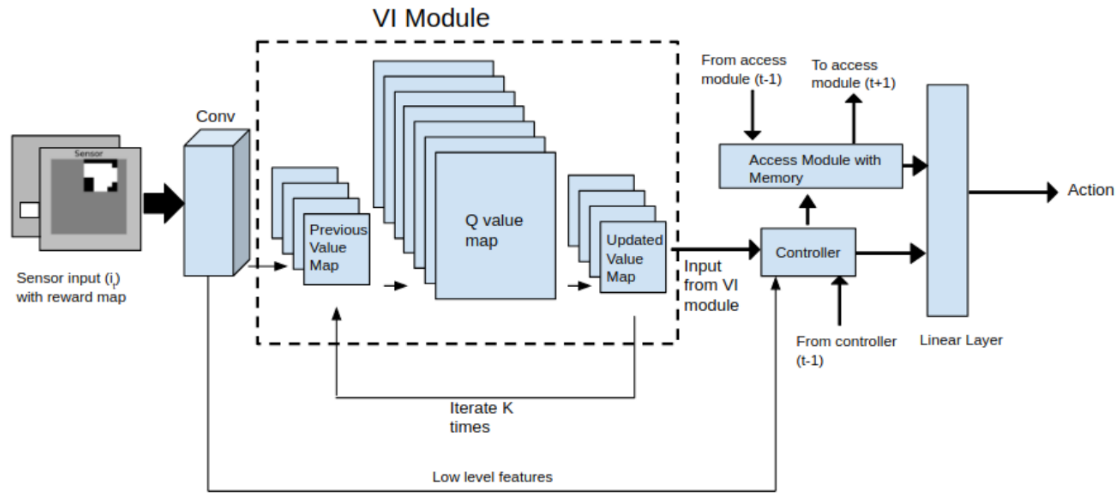
- **Conv block:** generate feature representation; R and initial V for VIN
  - **Input**: 2D map (m x n) stack with reward map (m x n) => (m x n x 2)
  - Convolve twice to get Reward layer (R)
  - Convolve once more to get initial V



UNIVERSITY OF WATERLOO

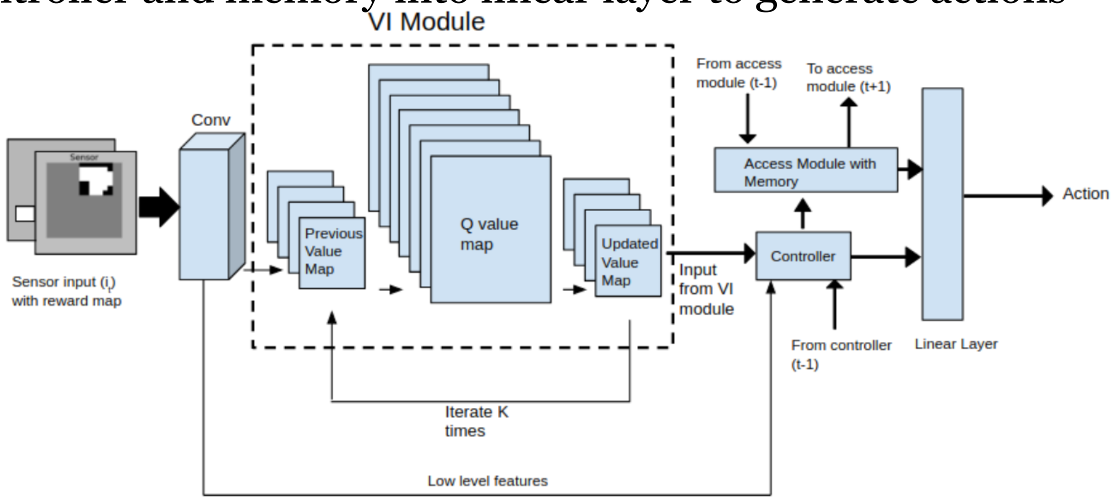
Architecture - VI Module

- First level of planning (Conv output into VIN)
- **VI module**: Plan in this space and calculate optimal value function in K iterations
  - **Input**: R and V concatenated
  - Convolved to get Q; Take max channel-wise to get updated V
  - Perform this K times to get Value map



Architecture - Controller

- Second level of planning (CNN output + VIN output into Controller):
- **Controller:**
  - **Input:** VIN output + low level feature representation (from Conv) into controller
  - Controller network (LSTM) interfaces with memory
  - Output from controller and memory into linear layer to generate actions



UNIVERSITY OF WATERLOO

Comparison with other work

- **Cognitive Mapping and Planning for Visual Navigation** (Gupta et al. 2017)
  - Value iteration Network + memory
  - Maps image scans to 2D map estimation by approximating all robot poses
- **Neural Network Memory Architectures for Autonomous Robot Navigation** (Chen et al. 2017)
  - CNN to extract features + DNC
- **Neural SLAM** (Zhang et al. 2017)
  - SLAM model using DNC
  - Efficient exploration

Experiment Setup

- **Baselines:**
  - **VIN**: just the VI module and no memory in place
  - **CNN + DNC**: CNN (4 Conv layers) extract features from observed map with the reward map and pass to the memory.
  - **MACN with a LSTM:** Planning module + LSTM instead of memory
  - **DQN**
  - **A3C**

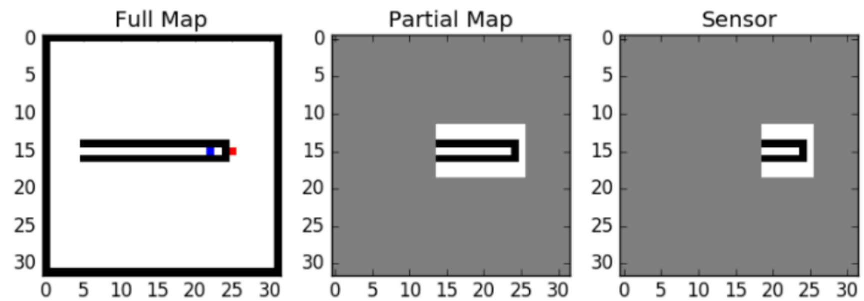
Experiments – 2D Maze

Model	Performance	16×16	32×32	64×64
VIN	*Success(%)*	0	0	0
	Test Error	0.63	0.78	0.81
CNN + Memory	*Success(%)*	0.12	0	0
	Test Error	0.43	0.618	0.73
MACN (LSTM)	*Success (%)*	88.12	73.4	64
	Test Error	0.077	0.12	0.21
MACN	*Success(%)*	**96.3**	**85.91**	**78.44**
	Test Error	**0.02**	**0.08**	**0.13**

- CNN+Memory performance is very poor
- MACN drop in accuracy on scaling is not as large as others

UNIVERSITY OF WATERLOO

Experiments – 2D Maze with Local Minima



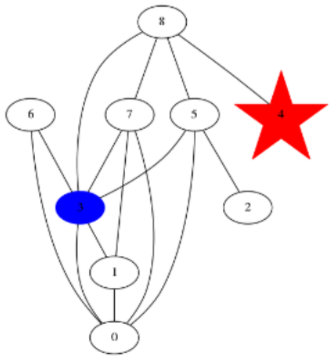
Model	*Success (%)*	*Maximum generalization length*
DQN	0	0
A3C	0	0
CNN + Memory	12	20
VIN	0	0
MACN	**100**	**330**

- Only MACN generalizes to longer tunnels
- Shift in memory states only when agent sees end of wall and on exit

UNIVERSITY OF WATERLOO

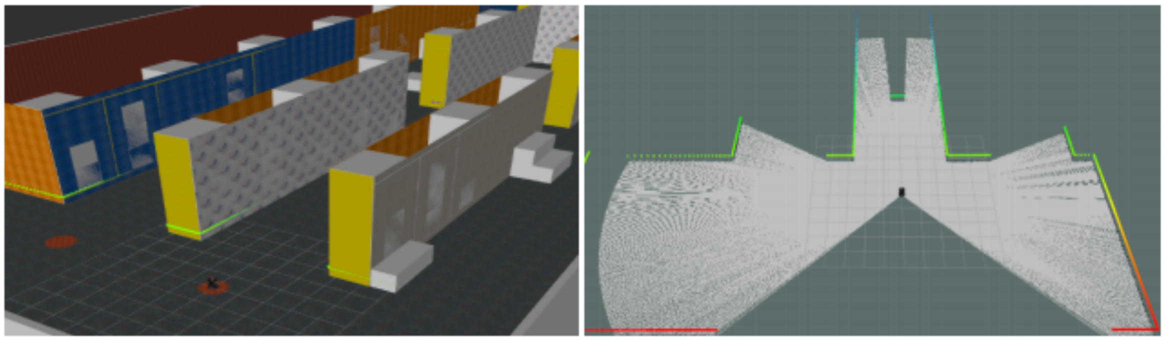
Experiments – Graph Search

- Blue node is the start state
- Red node is end state
- Agent can only observe edges connected to current node
- Problem where state space and action space are not limited

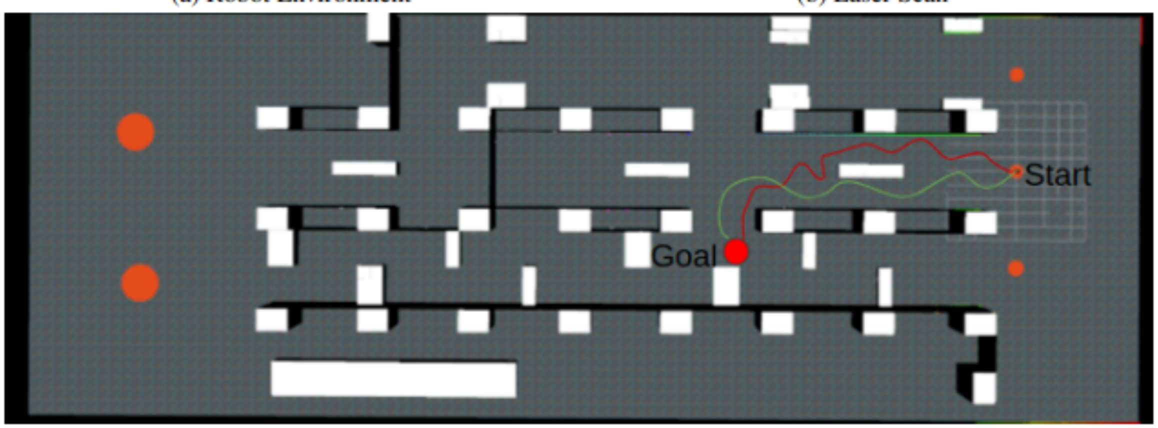


Model	Test Error, Success(%)			
	9 Nodes	16 Nodes	25 Nodes	36 Nodes
VIN	0.57, 23.39	0.61, 14	0.68, 0	0.71, 0
A3C	NA, 10	NA, 7	NA, 0	NA, 0
DQN	NA, 12	NA, 5.2	NA, 0	NA,0
CNN + Memory	0.25, 81.5	0.32, 63	0.56, 19	0.68, 9.7
MACN (LSTM)	0.14, 98	0.19, 96.27	0.26, 84.33	0.29, 78
MACN	**0.1, 100**	**0.18, 100**	**0.22, 95.5**	**0.28, 89.4**

Experiments – Continuous Control



(a) Robot Environment (b) Laser Scan



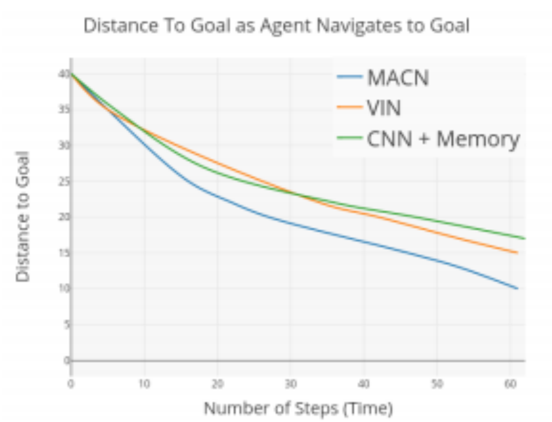
(c) Top Down View of Environment

- Converts this to required 2D
- Network output generates waypoints

Model	*Success (%)*
DQN,A3C	0
VIN	57.60
CNN + Memory	59.74
MACN	**71.3**

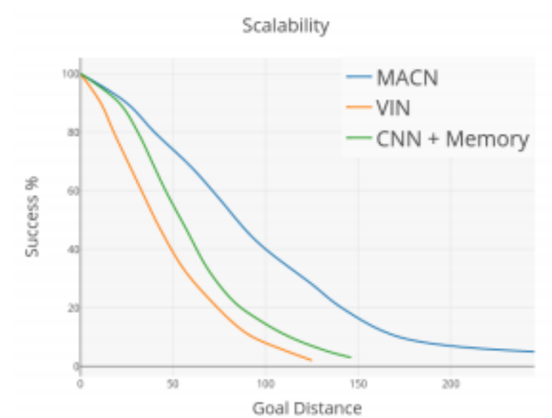
Experiments – Other comparisons

Convergence rate



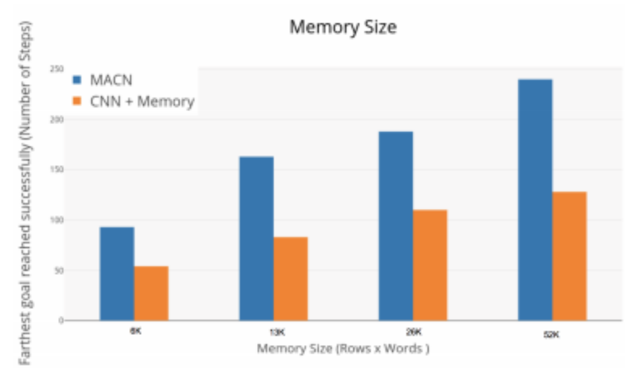
(a) Dist. to goal vs number of steps

Scaling with complexity



(b) Success % vs goal distance

Scaling with memory



UNIVERSITY OF WATERLOO

Conclusion and Discussion

- **Contributions:**
  - Novel end-to-end architecture that combines hierarchical planning and differentiable memory

- **Future work**
  - Efficient exploration
  - Take sensor errors into account