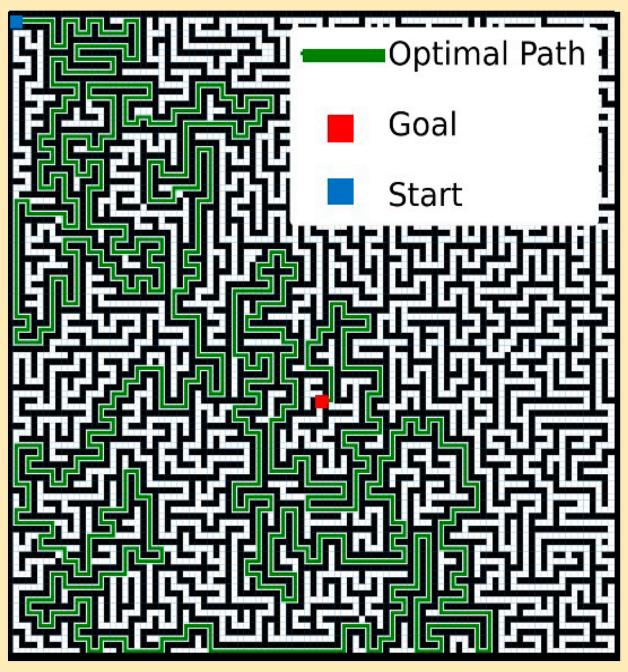
Scaling Value Iteration Networks to 5000 Layers for Extreme Long-Term Planning

Yuhui Wang*[1], Qingyuan Wu*[2], Weida Li[3], Dylan R. Ashley[4], Francesco Faccio[4], Chao Huang[2], Jürgen Schmidhuber[1,4,5]

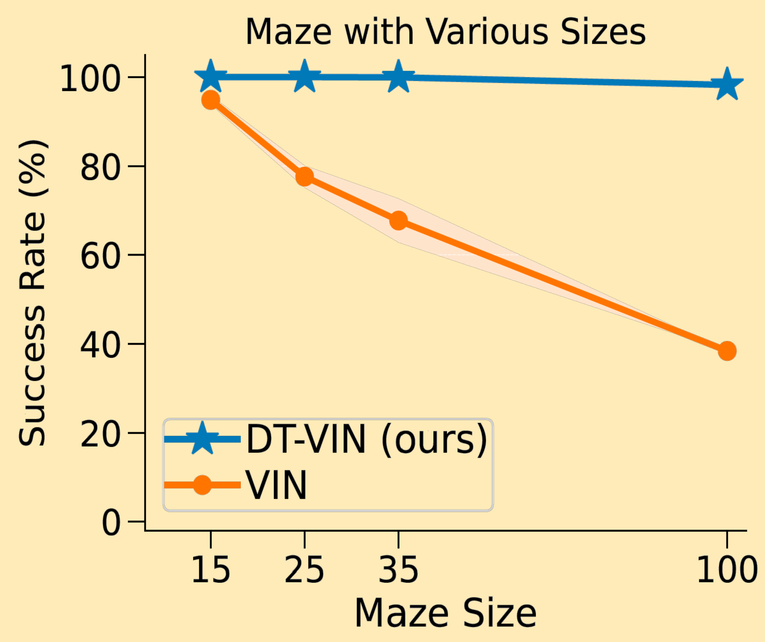
[1] Center of Excellence in GenAI, King Abdullah University of Science and Technology (KAUST), [2] The University of Southampton, [3] National University of Singapore, [4] The Swiss AI Lab IDSIA/USI/SUPSI, [5] NNAISENSE

Abstract

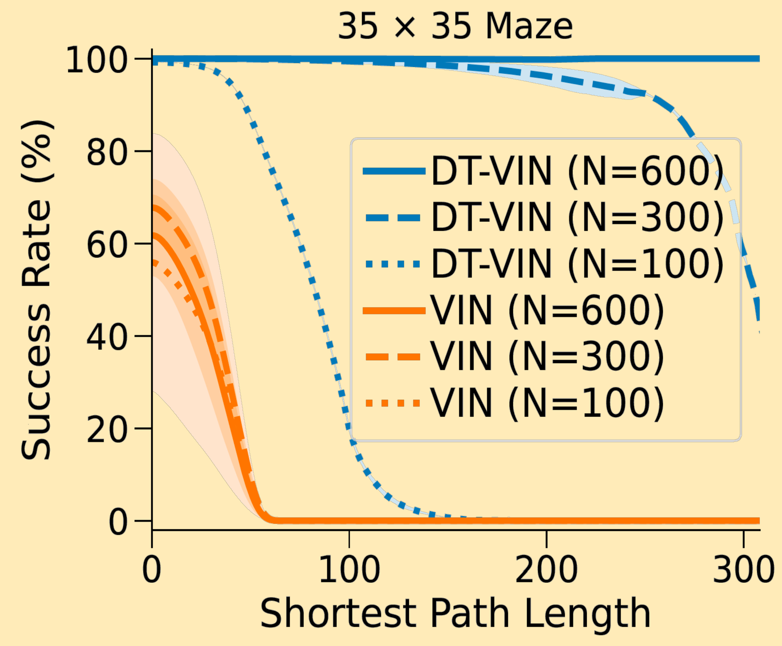
The Value Iteration Network (VIN) is an end-to-end differentiable architecture that performs value iteration on a latent MDP for planning in reinforcement learning (RL). However, VINs struggle to scale to long-term and large-scale planning tasks, such as navigating a 100x100 maze---a task which typically requires thousands of planning steps to solve. We observe that this deficiency is due to two issues: the representation capacity of the latent MDP and the planning module's depth. **We address these by augmenting the latent MDP with a dynamic transition kernel, dramatically improving its representational capacity, and, to mitigate the vanishing gradient problem, introduce an "adaptive highway loss" that constructs skip connections to improve gradient flow.** We evaluate our method on both 2D maze navigation environments and the ViZDoom 3D navigation benchmark. We find that our new method, named *Dynamic Transition VIN (DT-VIN)*, easily scales to 5000 layers and casually solves challenging versions of the above tasks. Altogether, we believe that DT-VIN represents a concrete step forward in performing long-term large-scale planning in RL environments.



100 × 100 Maze Navigation



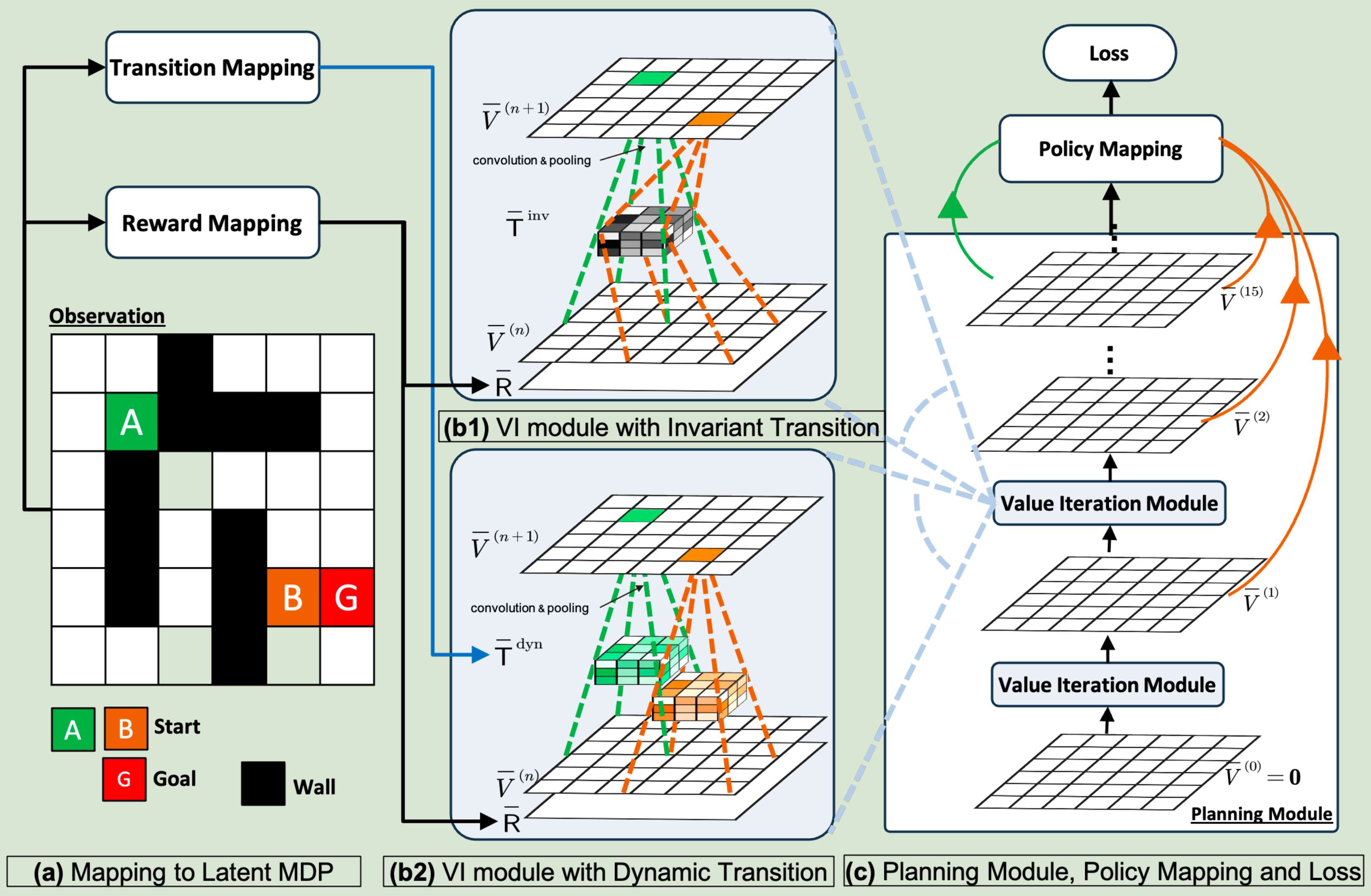
Performance for Different Domain Scales



Performance for Different Planning Steps

arXiv

Method



1) VI Module

VIN: invariant transition kernel

$$\overline{V}_{i,j}^{(n)} = \max_{\overline{a}} \sum_{i',j'} \overline{\mathsf{T}}^{\text{inv}}_{\overline{a},i',j'} \left(\overline{\mathsf{R}}_{i-i',j-j'} + \overline{V}^{(n-1)}_{i-i',j-j'} \right)$$

$$\overline{\mathsf{T}}^{\text{inv}} \in \mathbb{R}^{|\overline{\mathcal{A}}| \times F \times F}$$

Ours: dynamic transition kernel

$$\overline{V}_{i,j}^{(n)} = \max_{\overline{a}} \sum_{i',j'} \overline{\mathsf{T}}^{\text{dyn}}_{i,j,\overline{a},i',j'} \left(\overline{\mathsf{R}}_{i-i',j-j'} + \overline{V}^{(n-1)}_{i-i',j-j'} \right)$$

$$\overline{\mathsf{T}}^{\text{dyn}} = softmax\left(f^{\overline{\mathsf{T}}}(x) \right) \in \mathbb{R}^{m \times m \times |\overline{\mathcal{A}}| \times F \times F}$$

where $f^{\overline{\mathsf{T}}}$ is the transition mapping module

2) Loss function

VIN: Normal Loss

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} \ell \left(f^{\pi} \left(\overline{V}^{(N)}(x) \right) \right)$$

where

f^{π}: policy mapping module

l: loss function, e.g., cross entropy loss

x: observation

y: label

Ours: Adaptive Highway Loss

$$\mathcal{L}(\theta) = \frac{1}{K} \sum_{(x,y,l) \in \mathcal{D}} \sum_{1 \le n \le N} \mathbb{1}_{\{n \ge l\}} \mathbb{1}_{\{n \bmod l_j = 0\}} \ell \left(f^{\pi} \left(\overline{V}^{(n)}(x) \right), y \right)$$

where

n: the index of the layer

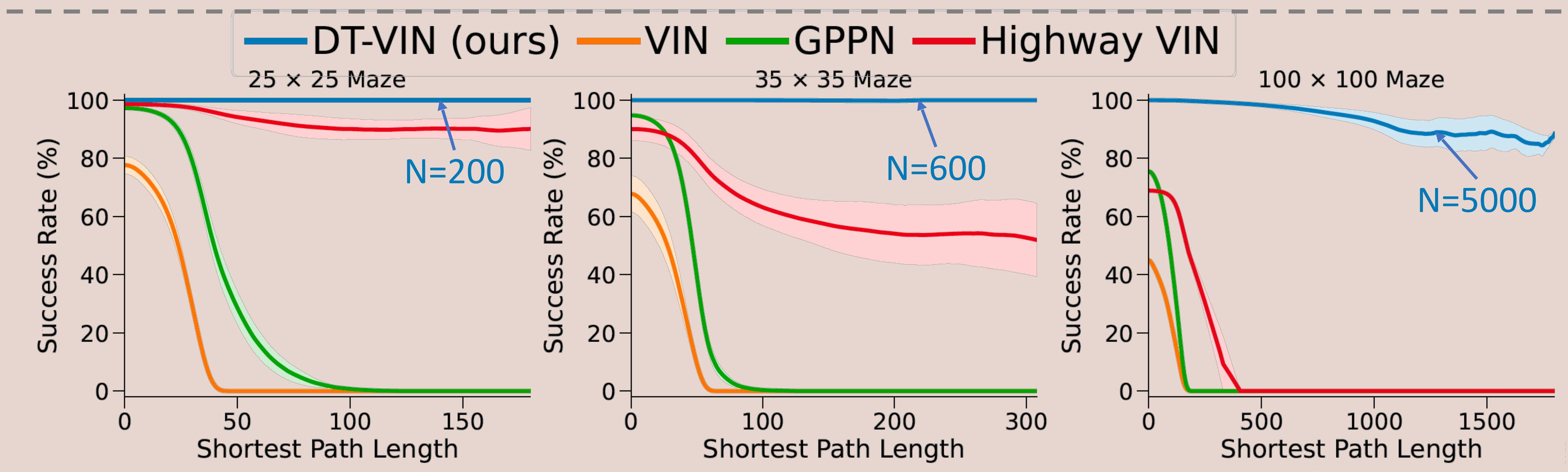
l: the length of the trajectory

l_j: jumping hyperparameter

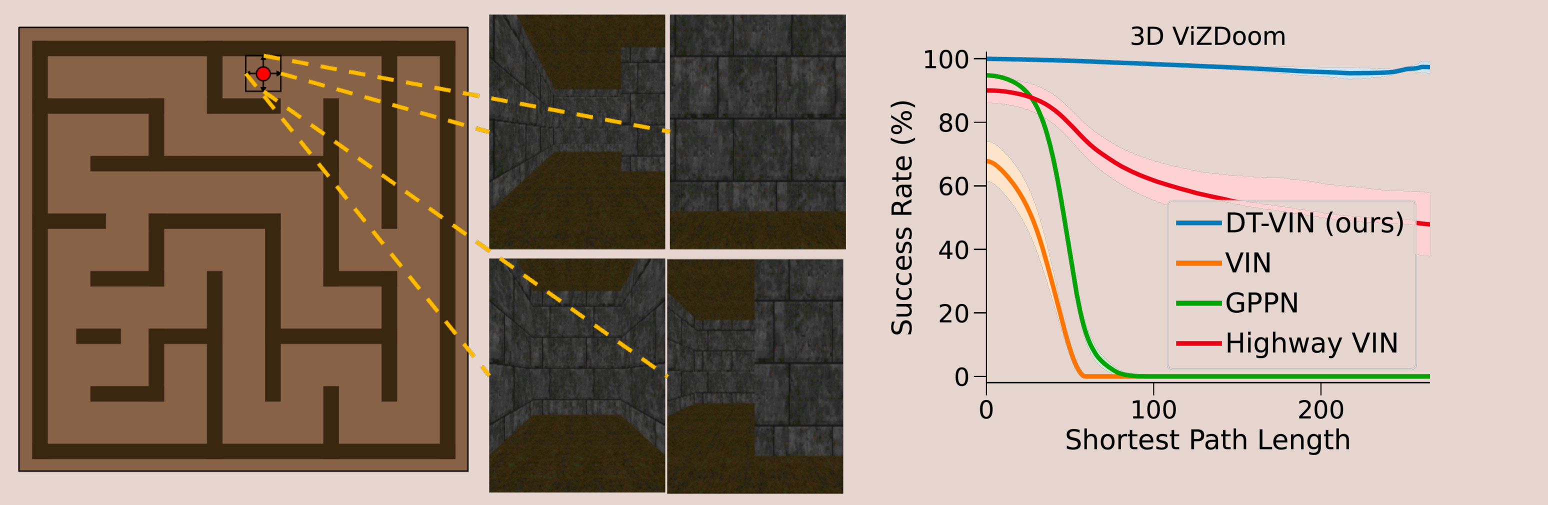
$\mathbb{1}$: indictor function

$$K = \sum_{(x,y,l) \in \mathcal{D}} \sum_{1 \le n \le N} \mathbb{1}_{\{n \ge l\}} \mathbb{1}_{\{n \bmod l_j = 0\}}$$

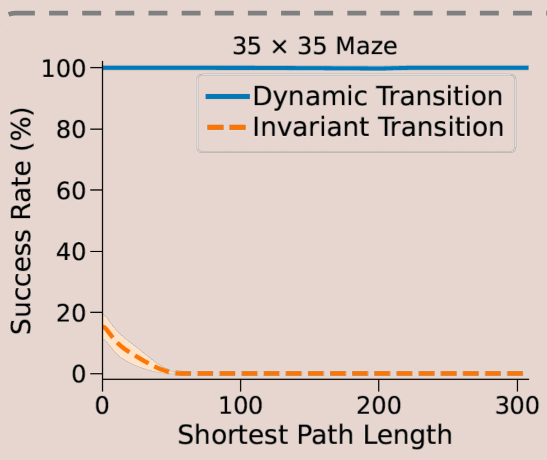
Results



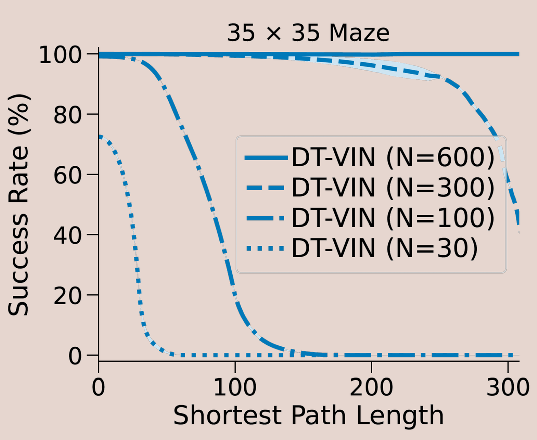
Success Rates on Grid World Navigation Task



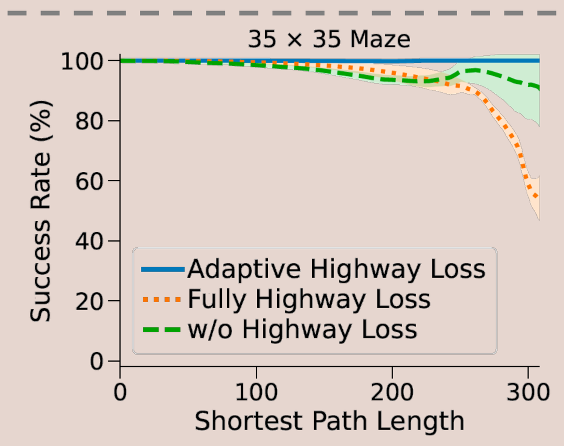
Success Rates on 3D ViZDoom Navigation Task



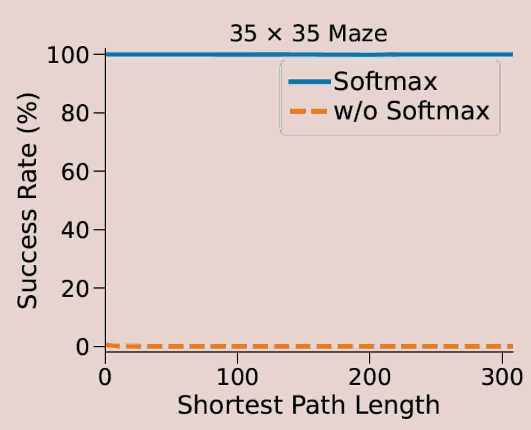
Dynamic Transition



Depth N



Adaptive Highway Loss



Softmax Operation

Correspondence to:
Yuhui Wang, yuhui.wang@kaust.edu.sa,
Dylan R. Ashley, mail@dylanashley.io