

Unrolling Dynamic Programming via Graph Filters



Universidad
Rey Juan Carlos



UNIVERSITY of
ROCHESTER

Sergio Rozada, Samuel Rey, Gonzalo Mateos, and Antonio G. Marques

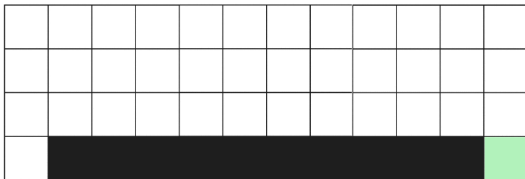
Dept. Signal Theory and Comms., King Juan Carlos University

Dept. Electrical and Computer Eng., University of Rochester

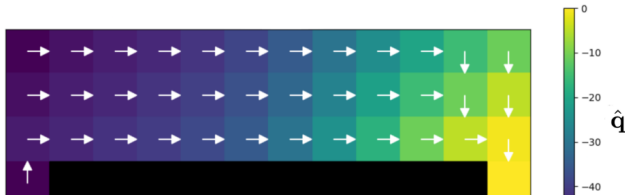


CAMSAP 2025 - Punta Cana - Dec. 14 - 17, 2025

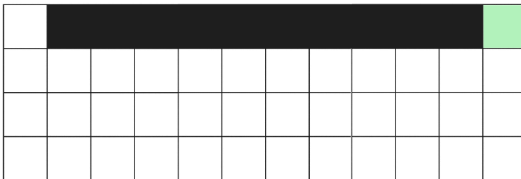
- ▶ We want to solve the cliff problem via **dynamic programming (DP)**



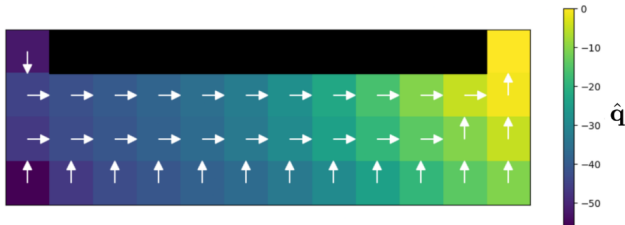
- ▶ Find **value function (VF)** via **Bellman's equation** and obtain **policy**



- ▶ If the problem changes in **structured** way



- ▶ We have to solve again \Rightarrow Solution does not **transfer**

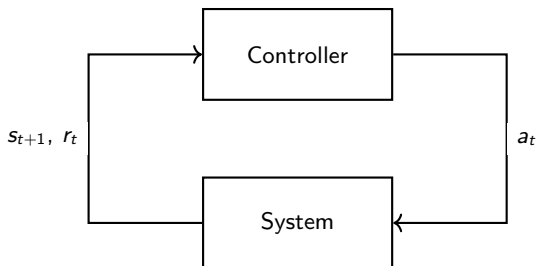


Can we transfer DP solutions?

- 1) Preliminaries: Policy iteration
- 2) DP meets GSP: Unrolling policy iteration
- 3) BellNet: A GNN to solve DP
- 4) Numerical results

- 1) Preliminaries: Policy iteration
- 2) DP meets GSP: Unrolling policy iteration
- 3) BellNet: A GNN to solve DP
- 4) Numerical results

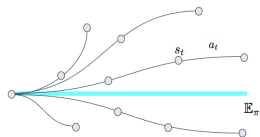
- ▶ DP formalized via the **Markov** decision process (MDP)
 - ⇒ Discrete **state space** \mathcal{S}
 - ⇒ Discrete **action space** \mathcal{A}
 - ⇒ **Transition** probability function $p(\cdot|s, a)$
 - ⇒ **Reward** function $r(s, a)$



[Bertsekas, Athena Scientific]

- ▶ Learn optimal **policy** $\pi : \mathcal{S} \mapsto \mathcal{A}$ solving

$$\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$$



⇒ Challenging **functional** optimization problem

- ▶ **VFs** key to **DP** formalization ⇒ Characterize **optimality conditions**

$$Q^{\pi}(s, a) := \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a \right]$$

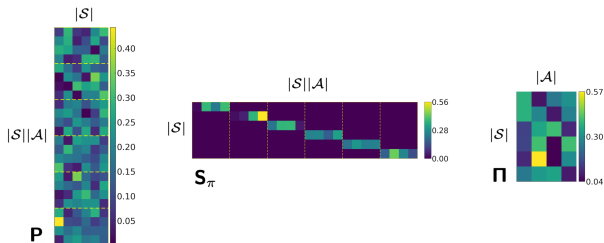
$$Q^{\pi}(s, a) = r(s, a) + \gamma \mathbb{E}_{\pi} [Q^{\pi}(s', a')]$$

- ▶ Bellman's equations \Rightarrow Linear system for fixed policy π

$$\mathbf{q}_\pi = \mathbf{r} + \gamma \mathbf{P}_\pi \mathbf{q}_\pi \quad \text{with} \quad \mathbf{P}_\pi = \mathbf{P} \mathbf{S}_\pi$$

$\Rightarrow \mathbf{P} \in \mathbb{R}^{|\mathcal{S}||\mathcal{A}| \times |\mathcal{S}|}$ is **transition** operator

$\Rightarrow \mathbf{S}_\pi = (\mathbf{I} \odot \mathbf{\Pi}^\top)^\top \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}||\mathcal{A}|}$ is **VF mapping** $\Rightarrow \mathbf{v}_\pi = \mathbf{S}_\pi \mathbf{q}_\pi$



- ▶ Leads to better policy $\Rightarrow \pi'(s) = \max_{a \in \mathcal{A}} Q^\pi(s, a)$

- ▶ **Policy iteration** \Rightarrow Classic algorithm to solve DP
 - \Rightarrow Guaranteed to **converge** to unique solution
 - \Rightarrow **Policy evaluation (PE)** \Rightarrow Solve Bellman's equation
 - \Rightarrow **Policy improvement (PI)** \Rightarrow Improve policy greedily
 - \Rightarrow Nested (double) loop:

$$\text{(PE)} \quad \mathbf{q}^{(n+1)} = \mathbf{r} + \gamma \mathbf{P}_{\pi^{(m)}} \mathbf{q}^{(n)} \quad \text{for } n = 1, \dots, N$$

$$\text{(PI)} \quad \pi^{(m+1)}(s) = \max_{a \in \mathcal{A}} Q^{(N)}(s, a) \quad \text{for } m = 1, \dots, M$$

- ▶ **New DP problem** requires running policy iteration **again**

- 1) Preliminaries: Policy iteration
- 2) DP meets GSP: Unrolling policy iteration
- 3) BellNet: A GNN to solve DP
- 4) Numerical results

- ▶ We **unroll** PE iteration

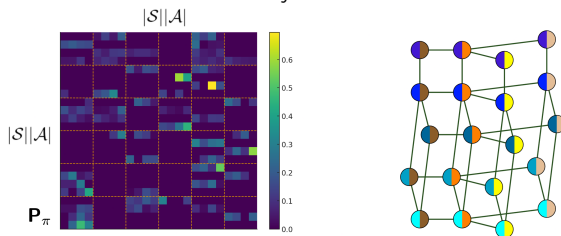
$$\begin{aligned}\mathbf{q}^{(k)} &= \mathbf{r} + \gamma \mathbf{P}_\pi \mathbf{q}^{(k-1)} \\ &= \mathbf{r} + \gamma \mathbf{P}_\pi \mathbf{r} + \gamma^2 (\mathbf{P}_\pi)^2 \mathbf{q}^{(k-2)} \\ &= \mathbf{r} + \gamma \mathbf{P}_\pi \mathbf{r} + \gamma^2 (\mathbf{P}_\pi)^2 \mathbf{r} + \gamma^3 (\mathbf{P}_\pi)^3 \mathbf{q}^{(k-3)} \\ &= \dots \\ &= \sum_{j=0}^{k-1} \gamma^j (\mathbf{P}_\pi)^j \mathbf{r} + \gamma^k (\mathbf{P}_\pi)^k \mathbf{q}^{(0)}\end{aligned}$$

- ▶ In the limit \Rightarrow PE converges to **true** \mathbf{q}_π and $\gamma^k \rightarrow 0$

$$\mathbf{q}_\pi = \sum_{j=0}^{\infty} \gamma^j (\mathbf{P}_\pi)^j \mathbf{r}$$

- ▶ VFs are **graph-filtered rewards** with known coefficients

$$\mathbf{q}_\pi = \sum_{j=0}^{\infty} \gamma^j (\mathbf{P}_\pi)^j \mathbf{r}$$



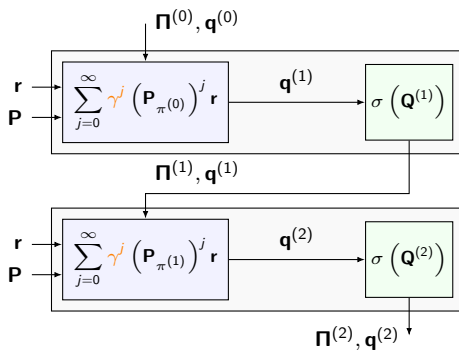
- ▶ PI **improves** over actions $\Rightarrow \pi'(s) = \max_{a \in \mathcal{A}} Q^\pi(s, a)$

$$\mathbf{\Pi}' = \sigma(\mathbf{Q}^\pi) \Leftrightarrow \Pi'_{ij} = \begin{cases} 1 & \text{if } j = \operatorname{argmax}_k Q_{ik}^\pi \\ 0 & \text{otherwise} \end{cases}$$

- PE + PI yields **linear–nonlinear** operator

$$\text{(PE)} \quad \mathbf{q}_\pi = \sum_{j=0}^{\infty} \gamma^j (\mathbf{P}_\pi)^j \mathbf{r}$$

$$\text{(PI)} \quad \Pi' = \sigma(\mathbf{Q}^\pi)$$



- 1) Preliminaries: Policy iteration
- 2) DP meets GSP: Unrolling policy iteration
- 3) BellNet: A GNN to solve DP
- 4) Numerical results

- **Obs.1:** By **Cayley-Hamilton** there exists finite-order filter

$$\mathbf{q}_\pi = \sum_{j=0}^{\infty} \gamma^j (\mathbf{P}_\pi)^j \mathbf{r} = \sum_{j=0}^K h_j (\mathbf{P}_\pi)^j \mathbf{r}$$

- **Obs.2:** In practice, policy iteration can be **truncated**

$$\hat{\mathbf{q}}_\pi = \sum_{j=0}^K \gamma^j (\mathbf{P}_\pi)^j \mathbf{r} + \gamma^{K+1} (\mathbf{P}_\pi)^{K+1} \mathbf{q}^{(0)}$$

- **Obs.3:** Max. non-linearity can be **relaxed** for tractability

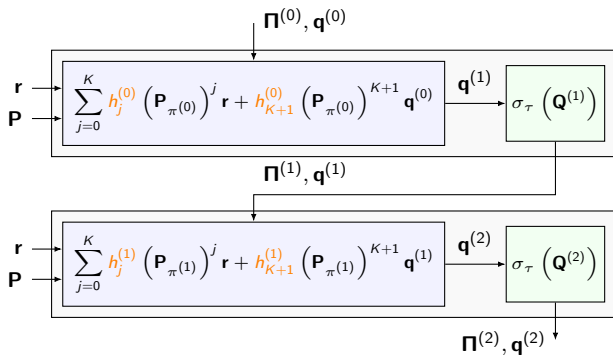
$$\mathbf{\Pi}' = \sigma_\tau(\mathbf{Q}^\pi) \quad \Leftrightarrow \quad \Pi'_{ij} = \frac{e^{Q_{ij}^\pi / \tau}}{\sum_{k=1}^{|\mathcal{A}|} e^{Q_{ik}^\pi / \tau}}$$

► BellNet $\Rightarrow \Phi(\cdot; \mathcal{H})$ with parameters $\mathcal{H} = \{(h_1^{(l)}, \dots, h_{K+1}^{(l)})\}_{l=0}^L$

\Rightarrow Implements PE and PI L times $\Rightarrow \{\hat{\mathbf{q}}, \hat{\Pi}\} := \Phi(\bar{\mathbf{q}}; \mathcal{H})$

$$\text{(PE)} \quad \mathbf{q}^{(l+1)} = \sum_{j=0}^K h_j^{(l)} (\mathbf{P}_{\pi^{(l)}})^j \mathbf{r} + h_{K+1}^{(l)} (\mathbf{P}_{\pi^{(l)}})^{K+1} \mathbf{q}^{(l)}$$

$$\text{(PI)} \quad \Pi^{(l+1)} = \sigma_{\tau}(\mathbf{Q}^{(l+1)})$$



- ▶ How to **train** our BellNet architecture?
- ▶ **Optimality condition** of VFs in DP

$$\mathbf{q}^* = \mathbf{r} + \gamma \mathbf{P}\mathbf{v}^* \quad \text{with} \quad \mathbf{v}^*(s) = \max_{a \in \mathcal{A}} \mathbf{q}^*(s, a)$$

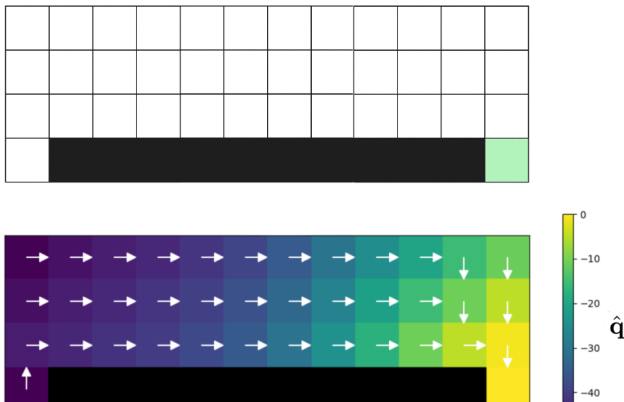
- ▶ Consider the **quadratic optimization** problem

$$\begin{aligned} \min_{\mathcal{H}} \quad & \ell(\mathcal{H}) = \|\mathbf{r} + \gamma \mathbf{P}\hat{\mathbf{v}} - \hat{\mathbf{q}}\|_2^2 \\ \text{s.t.} \quad & \{\hat{\mathbf{q}}, \hat{\mathbf{\Pi}}\} = \Phi(\bar{\mathbf{q}}; \mathcal{H}) \quad \hat{\mathbf{v}} = \mathbf{S}_{\hat{\pi}} \hat{\mathbf{q}} \end{aligned}$$

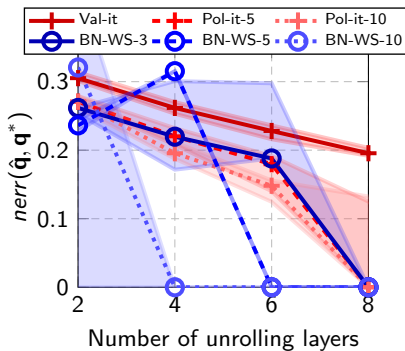
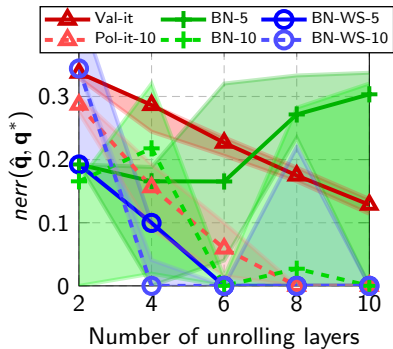
- ▶ Solve via **gradient descent** $\Rightarrow \nabla_{\mathcal{H}} \ell(\mathcal{H})$

- 1) Preliminaries: Policy iteration
- 2) DP meets GSP: Unrolling policy iteration
- 3) BellNet: A GNN to solve DP
- 4) Numerical results

- ▶ We train **BellNet** to solve the cliff-walking problem



- ▶ We test **BellNet** by solving the **mirrored** cliff-walking problem





Many thanks!