Latent Spherical Flow Policy for Reinforcement Learning with Combinatorial Actions

Lingkai Kong*, Anagha Satish*, Hezi Jiang, Akseli Kangaslahti, Andrew Ma, Wenbo Chen, Mingxiao Song, Lily Xu, Milind Tambe

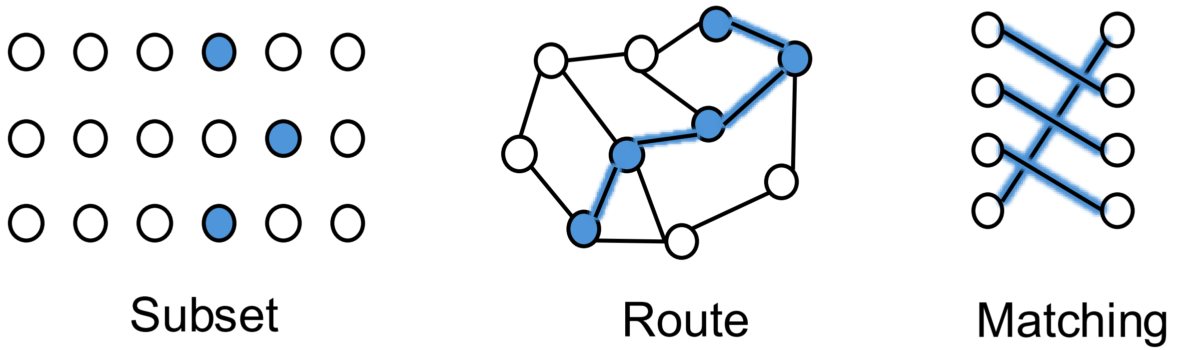
Spotlight

RL with Combinatorial Actions

Problem Definition and Bottleneck

Action: Satisfy state-dependent combinatorial **constraints** $a \in \mathcal{A}(s)$

Goal: Learn a policy distribution $\pi(a|s)$ over feasible actions $\mathcal{A}(s)$



Bottleneck: Combinatorial explosion

$|\mathcal{A}(s)|$ grows exponentially $\Longrightarrow$ cannot explicitly represent $\pi(a|s)$

e.g., *m=50, k=5* $\Longrightarrow |\mathcal{A}(s)| = \binom{50}{5} = \mathbf{2,118,760}$

Our Contribution:

The 1st flow matching policy for RL with combinatorial actions

Contribution 1

Solver-Parameterized Flow Policy

Compact Representation + Guaranteed feasibility

Contribution 2

Efficient & Stable Training

Reparameterized critic + Smoothed Bellman

Contribution I

Continuous Representation via CO Solver

Combinatorial Optimization (CO) Solver

- Off-the-shelf: Gurobi, CPLEX, etc.
- Linear objectives are most mature & scalable

Input → **Solver** → **Output**

Input
Cost vector: $c \in \mathbb{R}^m$
Constraints: $\mathcal{A}(s)$

Solver
$\arg\min_{a \in \mathcal{A}(s)} c^\top a$

Output
$a^\star \in \mathcal{A}$
✅ Always feasible

Key Observation:

Different cost vectors c → Different feasible actions $a^\star$

Cost vector $c \in \mathbb{R}^m$ is a continuous representation!

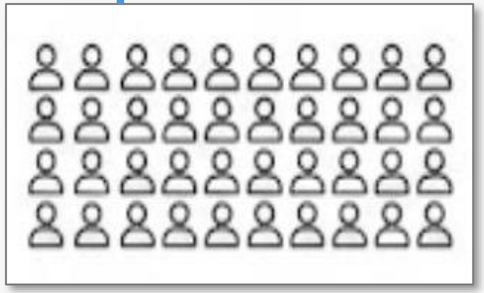
Our Idea Learn a ***continuous*** distribution over c instead of learning over exponential ***discrete*** actions

Flow Policy with Solver Parameterization

Architecture Overview

State s → Spherical Flow $\pi_\theta(c|s)$ $c \in \mathcal{S}^{m-1}$ Scale invariant Only direction matters → Sample cost c → CO Solver $\arg\min_{a \in \mathcal{A}(s)} c^\top a$ → Action $a^*(s,c)$

Illustrating Example:



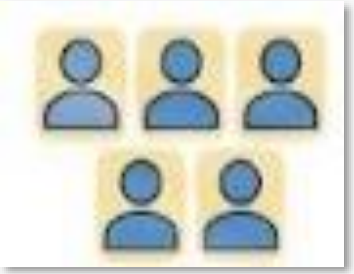
Pool of 50 people

$$\begin{bmatrix} \text{P1: \$0.1} \\ \text{P2: \$0.15} \\ \text{P3: \$0.2} \\ \text{......} \\ \text{P50: \$0.13} \end{bmatrix}$$

Cost Vector c

$$\boxed{\arg\min_{a \in \mathcal{A}(s)} c^\top a}$$

Solver select 5 people



5 people selected

✅ **Solver parameterization**: Compact, continuous representation ($c \in \mathbb{S}^{49}$, not $|\mathcal{A}(s)| \approx 2\text{M}$) with guaranteed feasibility

✅ **Flow Matching**: Expressive tool to learn distributions in this space

How to Train?

Two Challenges in Actor-critic

Training Framework: Actor-critic

Actor (Policy): Improve flow $\pi_\theta(c|s)$ using the critic

Critic: Q-function scores the generated samples $Q(s, a^\star(s, c))$

$\boldsymbol{Q(s,a)}$: Expected future reward of taking action a in state s

Two challenges arise:

1 **Efficiency**

2 **Discontinuity**

Policy Training:

Challenge I: Efficiency

How do we compute $Q(s, a^{\star}(s, c))$

Sample Cost

$c \sim \pi_\theta(\cdot \mid s)$

→

Solver

$$\boldsymbol{a}^*(\boldsymbol{s}, \boldsymbol{c}) = \arg\min_{a \in \mathcal{A}(s)} c^\top a$$

Solver call required for every sample

→

Critic Score

$Q_\phi(s, a^*(s, c))$

❌ Batch combinatorial optimization at every flow training step

Policy Training:

Solution: Reparameterized Critic

Observation Solver defines a mapping $c \to a \implies Q(s, a^\star(s, c))$ is just a function of (s, c)

Key Idea: Learn this composed function directly --- skip the solver

$$\tilde{Q}_\phi(s, c) \coloneqq Q\big(s, a^\star(s, c)\big)$$

Reparameterized critic: input (s, c) output Q value

Before

$(s, c) \longrightarrow$ Solver $\xrightarrow{a^\star}$ Q $\longrightarrow$ 1.5

✗ Solver call every time

After

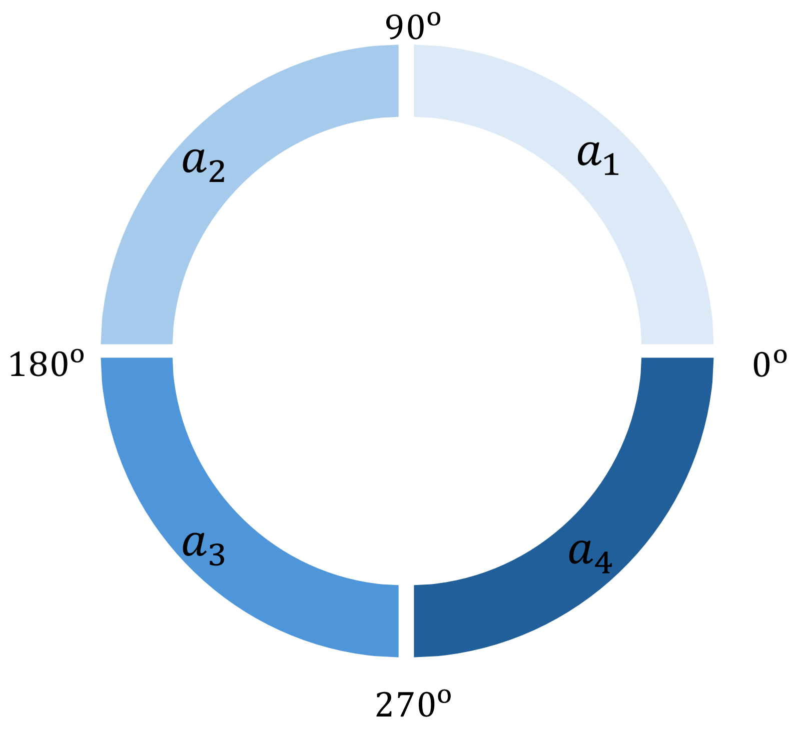
$(s, c) \longrightarrow$ $\tilde{Q}$ $\longrightarrow$ 1.5

✓ No solver at training time

Policy Training:

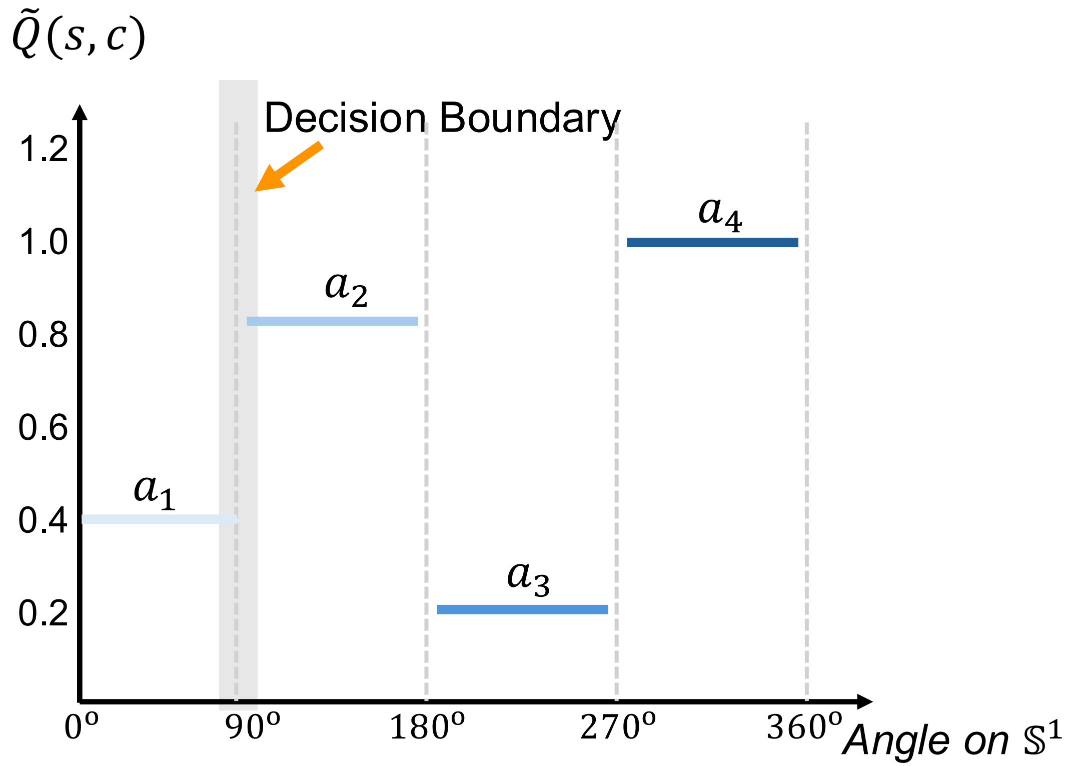
Challenge II: Discontinuity

Figure 1: 2D cost space



Solver partitions cost space

Figure 2: $\tilde{Q}(s,c)$ vs Angle



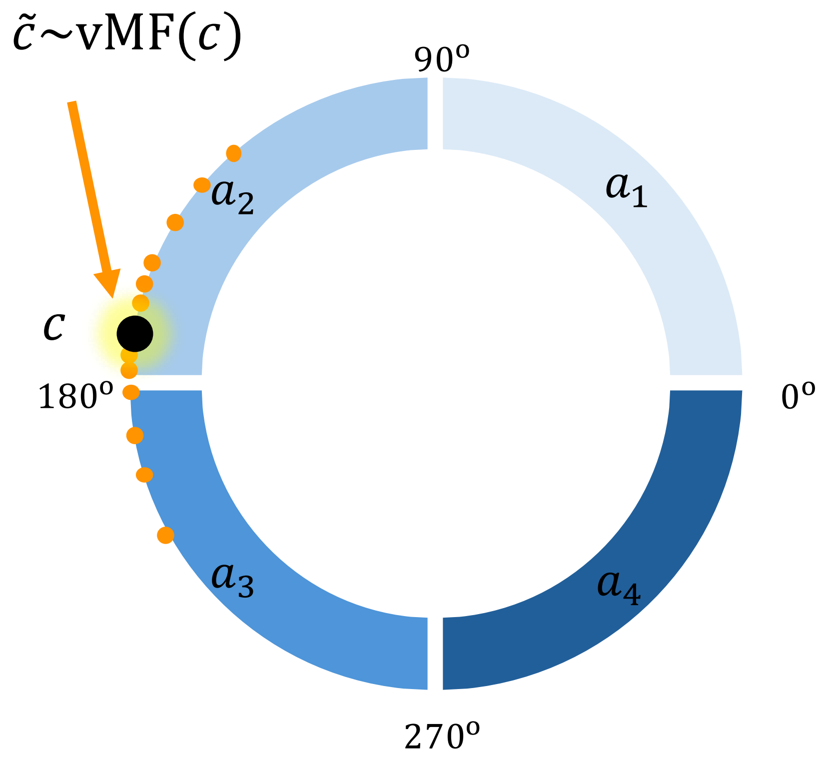
Boundary cross ⇒ different action ⇒ Q jumps

Discontinuities break critic learning – standard TD methods fail

Policy Training:

Solution: Smoothed Bellman Update

Solution: Local average on sphere



Smoothed Bellman Equation:

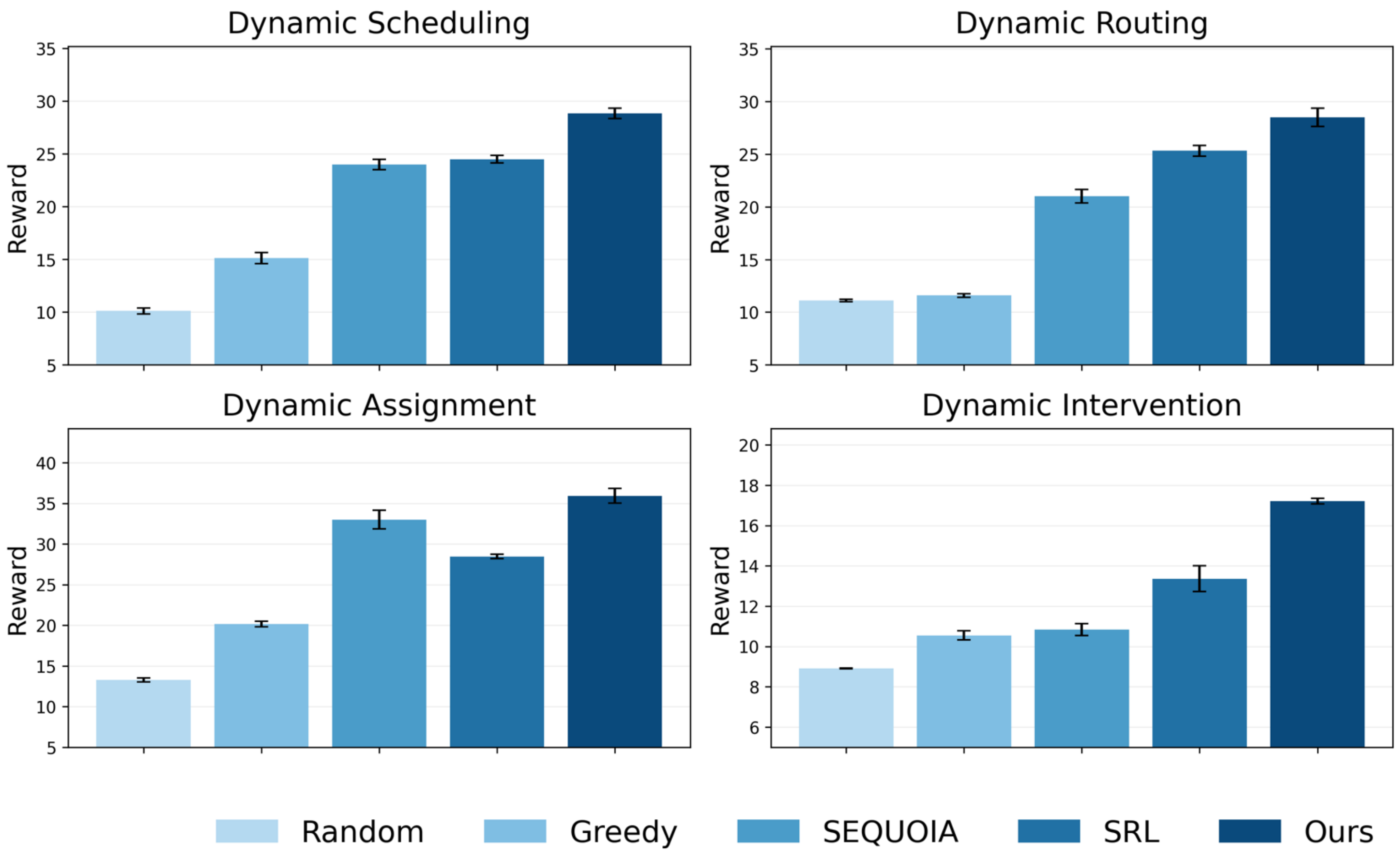
$$\tilde{Q}(s,c) = \mathbb{E}_{\tilde{c}\sim \mathrm{vMF}(c)}[\, r(s,a^*(\tilde{c})) + \gamma \mathbb{E}_{\tilde{c}' \sim \mathrm{vMF}(c')} \mathbb{E}_{s' \sim P} \mathbb{E}_{c'} [\tilde{Q}(s', \tilde{c}')]]$$

Proposition 1: Uniqueness	**Proposition 2: Smoothness**
Exists a unique fixed point $\tilde{Q}^*(s,c)$	$\tilde{Q}^*(s,\cdot)$ is smooth over $c \in \mathbb{S}^{d-1}$

Stable critic learning via standard TD methods

Simulation Results

Combinatorial RL Benchmarks



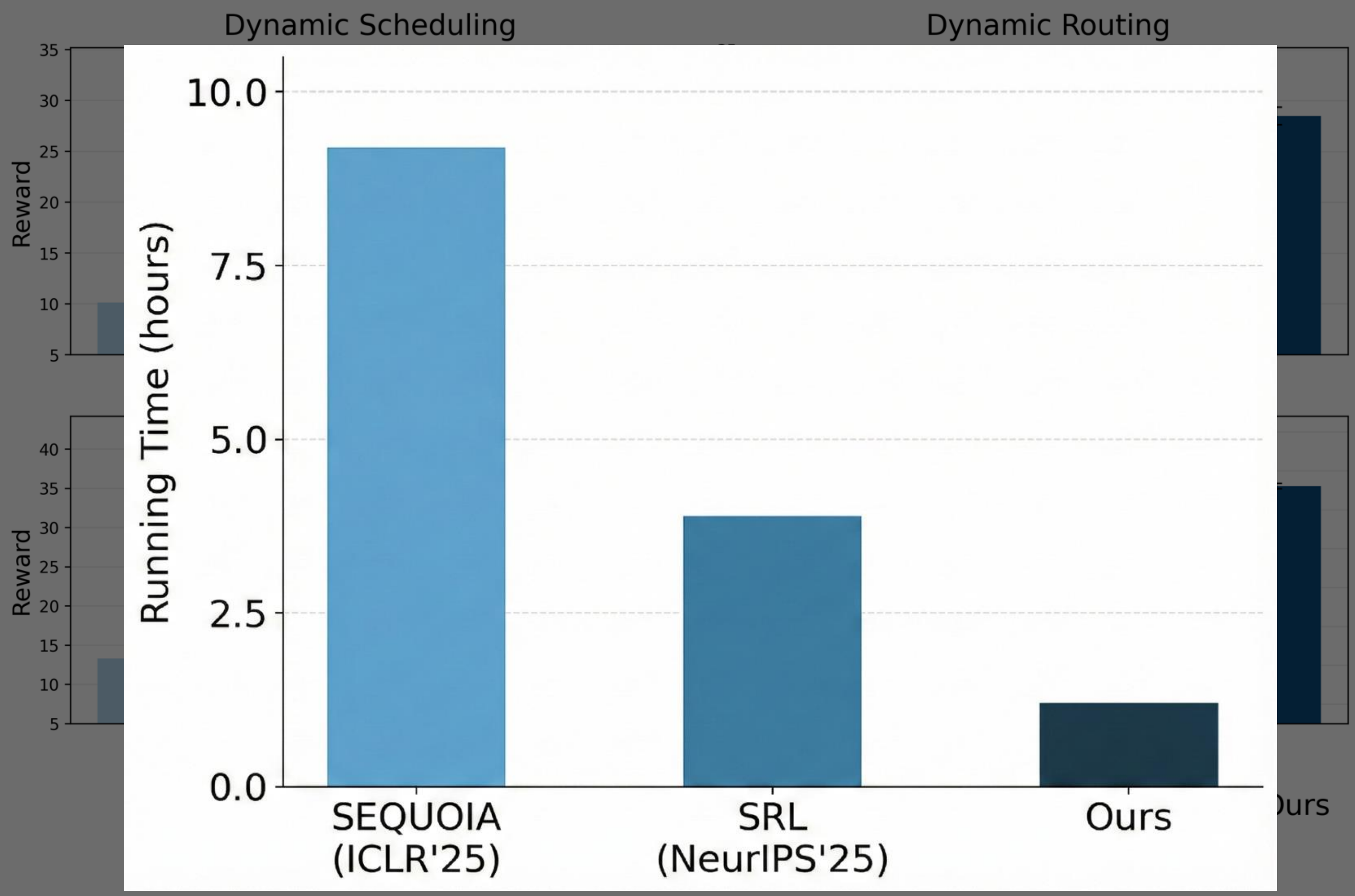
+21% over the best baseline

SEQUOIA (Xu et al, ICLR'25)

SRL (Hoppe et al, NeurIPS'25)

Simulation Results

Combinatorial RL Benchmarks



SEQUOIA (Xu et al, ICLR'25)

SRL (Hoppe et al, NeurIPS'25)

⚡ Faster training time!