Factored Policy Gradients

Leveraging Structure for Efficient Learning in MOMDPs

T. Spooner, N. Vadori and S. Ganesh

J.P. Morgan AI Research

December 2021

Overview

Introduction

1. Motivation and High-Level Overview

Influence Networks

1. Policy Factorisation

Factored Policy Gradients

1. Variance Properties

Experiments

Conclusions

Context

Many real-world problems are naturally **modular/hierarchical** [1]:

1. *market making*;
2. *multi-venue optimal execution*;
3. control of water reservoirs;
4. energy consumption optimisation;
5. elevator scheduling. . .

State-of-the-art methods for MDPs fail when:

1. the *dimensionality of the action-space* is too large; or
2. the *multiplicity of the objective* is too high.

Context

Where does our research sit?

Figure: Spectrum from model-free to model-based reinforcement learning.

Context

Where does our research sit?

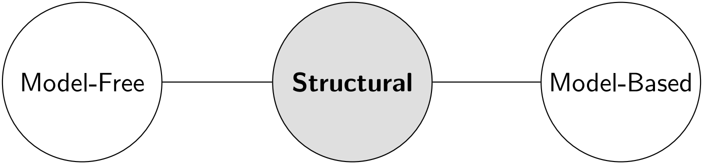


Figure: Spectrum from model-free to model-based reinforcement learning.

Key Research Question

How do we leverage structural knowledge?

Influence Networks

Consider a **scalarised multi-objective MDP**, with

$$J(\boldsymbol{\theta}) \doteq \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\psi(s, \boldsymbol{a}) \doteq \sum_{j=1}^{M} \lambda_j \psi_j(s, \boldsymbol{a})\right], \tag{1}$$

and **parameterised policy** $\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|s)$.

The *target,* ψ*, breaks down into m distinct components.*

- **Each sub-target,** ψ_i**, depends on a subset of the full action.**

Influence Networks

Example (Search Bandit)

The search bandit has an target of the form:

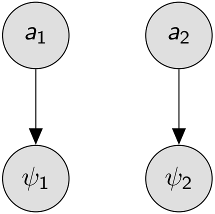
$$\psi(\boldsymbol{a}) \doteq \sum_{i=1}^{2} \psi_i(a_i) \doteq -|a_1 - c_1| - |a_2 - c_2|.$$

Influence Networks

Example (Search Bandit)

The search bandit has an target of the form:

$$\psi(\boldsymbol{a}) \doteq \sum_{i=1}^{2} \psi_i(a_i) \doteq -\left|a_1 - c_1\right| - \left|a_2 - c_2\right| .$$



(a) Influence network.

$$\boldsymbol{K} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

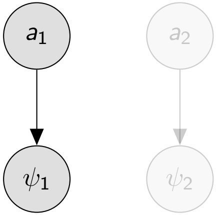
(b) Influence matrix.

Influence Networks

Example (Search Bandit)

The search bandit had an target of the form:

$$\psi(\boldsymbol{a}) \doteq \sum_{i=1}^{2} \psi_i(a_i) \doteq -\left|a_1 - c_1\right| - \left|a_2 - c_2\right|.$$



(a) Influence network.

$$\boldsymbol{K} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

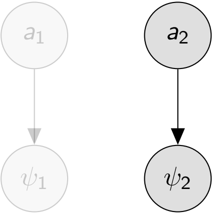
(b) Influence matrix.

Influence Networks

Example (Search Bandit)

The search bandit had an target of the form:

$$\psi(\boldsymbol{a}) \doteq \sum_{i=1}^{2} \psi_i(a_i) \doteq -|a_1 - c_1| - |a_2 - c_2|.$$



(a) Influence network.

$$\boldsymbol{K} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(b) Influence matrix.

Influence Networks

Example (Coupled)

Consider another target with the form:

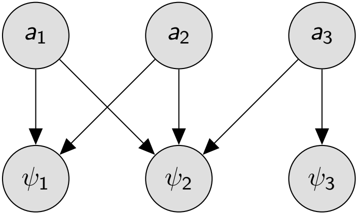
$$\psi(s, \boldsymbol{a}) \doteq \lambda_1 \psi_1(s, (a_1, a_2)) + \lambda_2 \psi_2(s, (a_1, a_2, a_3)) + \lambda_3 \psi_3(s, (a_3))\,.$$

Influence Networks

Example (Coupled)

Consider another target with the form:

$$\psi(s, \boldsymbol{a}) \doteq \lambda_1 \psi_1(s, (a_1, a_2)) + \lambda_2 \psi_2(s, (a_1, a_2, a_3)) + \lambda_3 \psi_3(s, (a_3)) \,.$$



(a) Influence network.

$$\boldsymbol{K} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

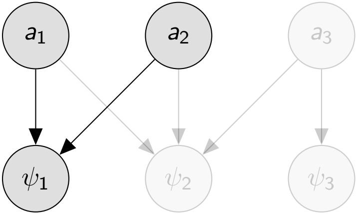
(b) Influence matrix.

Influence Networks

Example (Coupled)

Let's analyse the following objective:

$$\psi(s, \boldsymbol{a}) \doteq \lambda_1 \psi_1(s, (a_1, a_2)) + \lambda_2 \psi_2(s, (a_1, a_2, a_3)) + \lambda_3 \psi_3(s, (a_3)) \,.$$



(a) Influence network.

$$\boldsymbol{K} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

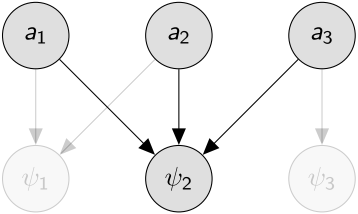
(b) Influence matrix.

Influence Networks

Example (Coupled)

Let's analyse the following objective:

$$\psi(s, \boldsymbol{a}) \doteq \lambda_1 \psi_1(s, (a_1, a_2)) + \lambda_2 \psi_2(s, (a_1, a_2, a_3)) + \lambda_3 \psi_3(s, (a_3)) \,.$$



(a) Influence network.

$$\boldsymbol{K} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

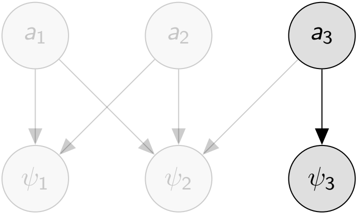
(b) Influence matrix.

Influence Networks

Example (Coupled)

Let's analyse the following objective:

$$\psi(s, \boldsymbol{a}) \doteq \lambda_1 \psi_1(s, (a_1, a_2)) + \lambda_2 \psi_2(s, (a_1, a_2, a_3)) + \lambda_3 \psi_3(s, (a_3)).$$



(a) Influence network.

$$\boldsymbol{K} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

(b) Influence matrix.

Key Research Question

How do we encode policy factorisation in an influence network?

Factored Influence Networks

We consider the class of **parameterised, stochastic policies**:

$$\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|s) \doteq \mathbb{P}_{\boldsymbol{\theta}}\left(\boldsymbol{a}|s\right).$$

The *policy is typically broken down into a* **product distribution**:

$$\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|s) \doteq \prod_{i=1}^{N} \pi_{i,\boldsymbol{\theta}}(\sigma_i^\pi(\boldsymbol{a})|s),$$

where $\{\sigma_i^\pi(\boldsymbol{a}) : i \in [N]\}$ denotes a set of disjoint partitions over $\boldsymbol{a}$.

Factored Influence Networks

For example, a common choice is the **Normal distribution**:

$$\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|s) \doteq \mathcal{N}(\boldsymbol{a} \mid \boldsymbol{\mu}_{\boldsymbol{\theta}}(s), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(s)),$$

where $\boldsymbol{\mu}_{\boldsymbol{\theta}} : \mathcal{S} \to \mathbb{R}^{|\mathcal{A}|}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\theta}} : \mathcal{S} \to \mathbb{R}^{|\mathcal{A}| \times |\mathcal{A}|}$.

Factored Influence Networks

For example, a common choice is the **Normal distribution**:

$$\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|s) \doteq \mathcal{N}(\boldsymbol{a} \mid \boldsymbol{\mu}_{\boldsymbol{\theta}}(s), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(s)),$$

where $\boldsymbol{\mu}_{\boldsymbol{\theta}} : \mathcal{S} \to \mathbb{R}^{|\mathcal{A}|}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\theta}} : \mathcal{S} \to \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$.

For tractability, we *typically assume isotropicity*:

1. The covariance $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ is constrained to diagonal matrices.
2. The policy then reduces to a product:

$$\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|s) = \prod_{i=1}^{N} \mathcal{N}(a_i \mid \mu_{i,\boldsymbol{\theta}}(s), \Sigma_{i,\boldsymbol{\theta}}(s)).$$

We can exploit this kind of factorisation!

Factored Influence Networks

Search Example

Consider the factorisation:

$$\sigma_1^\pi(\boldsymbol{a}) \doteq (a_1),$$
$$\sigma_2^\pi(\boldsymbol{a}) \doteq (a_2).$$

The new graph represents the **probabilistic influence of each policy factor** over the targets.

1. *Note that* $\sigma_1^\pi(\boldsymbol{a}) \perp\!\!\!\perp \sigma_2^\pi(\boldsymbol{a})$.

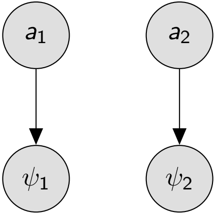


Figure: Original Influence Network, $\mathcal{G}$.

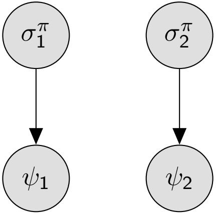


Figure: Factored Influence Network, $\mathcal{G}_\Sigma$.

Factored Influence Networks

Coupled Example

Consider the factorisation:

$$\sigma_1^\pi(\boldsymbol{a}) \doteq (a_1, a_2),$$
$$\sigma_2^\pi(\boldsymbol{a}) \doteq (a_3).$$

The new graph represents the **probabilistic influence of each policy factor** over the targets.

1. *Note that* $\sigma_1^\pi(\boldsymbol{a}) \perp\!\!\!\perp \sigma_2^\pi(\boldsymbol{a})$.

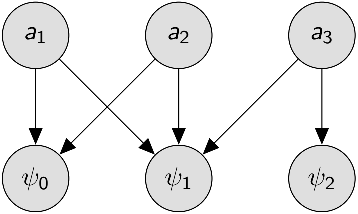


Figure: Original Influence Network, $\mathcal{G}$.

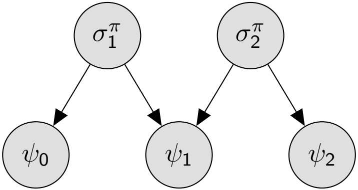


Figure: Factored Influence Network, $\mathcal{G}_\Sigma$.

Key Research Question

How do we use (factored) influence networks?

Policy Gradients

We consider the **policy optimisation** setting.

- Task is the *solve for the optimal policy*, $\pi_{\boldsymbol{\theta}^\star}$, where

$$\boldsymbol{\theta}^\star \doteq \arg\max_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \, .$$

Policy gradient methods leverage Sutton's key result [2]:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}_{\pi_{\boldsymbol{\theta}}}[\boldsymbol{S}(s, \boldsymbol{a}) \, \psi(s, \boldsymbol{a})] \, .$$

ψ Target function.

$\boldsymbol{S}$ Score matrix of size $|\boldsymbol{\theta}| \times 1$; i.e. $\nabla_{\boldsymbol{\theta}} \ln \pi_{\boldsymbol{\theta}}(\boldsymbol{a}|s)$.

Policy Gradients

Vanilla policy gradients ignore known independencies.

If we assume that:

1. The *problem has dependence structure* encoded by $\mathcal{G}_\Sigma$.

...then we know two things:

1. That ψ *is linearly separable*.
2. That $\pi_{\boldsymbol{\theta}}$ *has a factored representation*.

We can remove extraneous targets that contribute only noise.

Factored Baselines

For each policy factor, $i \in [K]$, we define a **factor-level baseline**:

$$b_i^{\mathrm{F}}(s, \boldsymbol{a}) \doteq \left[(\boldsymbol{J} - \boldsymbol{K})\ \psi(s, \boldsymbol{a})\right]_i ,$$

where $\boldsymbol{K}$ *is the biadjacency matrix of the factored influence network*.

Example (Search Bandit)

In the search bandit the influence matrix is unit-diagonal $\boldsymbol{K} = \boldsymbol{I}_2$, s.t.

$$b_1^{\mathrm{F}}(s, \boldsymbol{a}) = \psi_2(s, \boldsymbol{a}), \qquad \text{and} \qquad b_2^{\mathrm{F}}(s, \boldsymbol{a}) = \psi_1(s, \boldsymbol{a}).$$

Factored Policy Gradients

Factored policy gradient (FPG) methods use an expanded variant:

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \doteq \mathbb{E}_{\pi_{\boldsymbol{\theta}}}\left[\boldsymbol{S}(s, \boldsymbol{a})\, \boldsymbol{J}\left(\psi(s, \boldsymbol{a}) - \boldsymbol{b}^{\mathrm{F}}\right)\right],$$

$$= \mathbb{E}_{\pi_{\boldsymbol{\theta}}}[\boldsymbol{S}(s, \boldsymbol{a})\, \boldsymbol{K}\, \psi(s, \boldsymbol{a})].$$

- ψ Vector of target functions.
- $\boldsymbol{J}$ All-ones matrix.
- $\boldsymbol{K}$ *Biadjacency matrix of the factored influence network*.
- $\boldsymbol{S}$ Score matrix of size $|\boldsymbol{\theta}| \times k$; i.e.

$$\boldsymbol{S}(s, \boldsymbol{a}) \doteq \left[\nabla_{\boldsymbol{\theta}} \ln \pi_{1,\boldsymbol{\theta}}(\boldsymbol{a}|s)^{\top}, \ldots, \nabla_{\boldsymbol{\theta}} \ln \pi_{k,\boldsymbol{\theta}}(\boldsymbol{a}|s)^{\top}\right]^{\top}.$$

Factored Policy Gradients

If the factored influence network is unbiased (i.e. correct), then

$$\mathbb{E}_{\pi_{\boldsymbol{\theta}}}[\boldsymbol{S}(s, \boldsymbol{a})\ \psi(s, \boldsymbol{a})] \equiv \mathbb{E}_{\pi_{\boldsymbol{\theta}}}[\boldsymbol{S}(s, \boldsymbol{a})\, \boldsymbol{K}\, \boldsymbol{\psi}(s, \boldsymbol{a})]\,.$$

- Hinges on *key property of score functions*: $\mathbb{E}_{\pi_{\boldsymbol{\theta}}}[\boldsymbol{S}(s, \boldsymbol{a})] = 0$.

Factored policy gradients *remove redundant terms*.

- The matrix $\boldsymbol{K}$ captures independencies between $\pi_{\boldsymbol{\theta}}$ and $\boldsymbol{\psi}$.

Key Research Question

When are FPGs $\succ$ VPGs?

Variance Decomposition

We show that for each policy factor, $i \in [N]$, there is a **linear decomposition**:

$$\mathbb{V}[\mathrm{VPGs}_i] - \mathbb{V}[\mathrm{FPGs}_i] = \underbrace{\alpha_i \, \mathbb{E}_{\bar{\sigma}_i^\pi(\boldsymbol{a})}\left[\left(b_i^{\mathrm{F}}\right)^2\right]}_{\text{Symmetric}} + \underbrace{2\beta_i \mathbb{E}_{\bar{\sigma}_i^\pi(\boldsymbol{a})}\left[b_i^{\mathrm{F}}\right]}_{\text{Asymmetric}},$$

where

$$\alpha_i = \mathbb{E}_{\sigma_i^\pi(\boldsymbol{a})}[\langle \boldsymbol{S}_{\cdot,i}, \boldsymbol{S}_{\cdot,i} \rangle] \,,$$
$$\beta_i = \mathbb{E}_{\sigma_i^\pi(\boldsymbol{a})}\left[\langle \boldsymbol{S}_{\cdot,i}, \boldsymbol{S}_{\cdot,i} \rangle \left(\psi + b_i^{\mathrm{F}}\right)\right] .$$

Variance Decomposition

The **first term is a free-lunch** that scales with $\left(b_i^{\mathrm{F}}\right)^2$.

- Non-negative reduction deriving from the removal of terms in the gradient that are not related to the policy factors.

The **second is a coupling term** that scales with b_i^{F}.

- Coupling/covariance term between the new and old estimators.

Variance Reduction

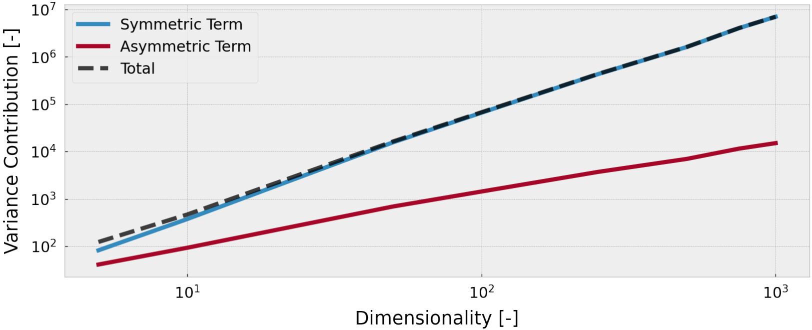


Figure: Variance reduction due to FPGs as a function of the action-space dimensionality on the search bandit.

Key Research Question

Do these theoretical results translate into practice?

Search Bandit

Take a 1000-dimensional **search bandit** with $R(\boldsymbol{a}) \doteq -\sum_{i=1}^{1000} |a_i - c_i|$.

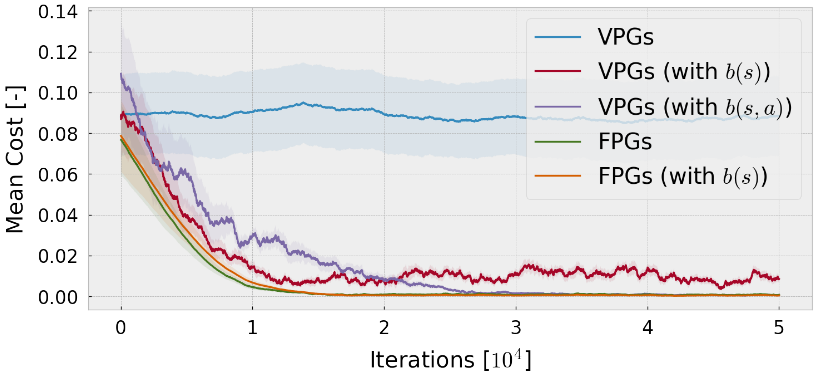


Figure: Benchmarks comparison for $|\mathcal{A}| = 1000$.

Traffic Networks

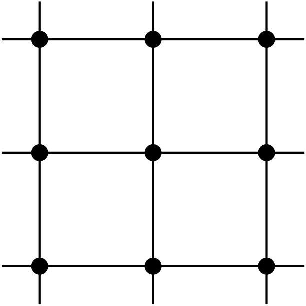
The **3×3 traffic network** [3] problem can be formulated as a graph:

- Vertices: The intersections are the vertices $\mathcal{V}$.
- Edges: The roads of the network are the edges $\mathcal{E}$.
- Cars: A set of $\mathcal{C}$ cars populate the network.

Objective is to *minimise delay:*

$$R(s, \boldsymbol{a}) \doteq \sum_{e \in \mathcal{E}} R_e(s, \boldsymbol{a}),$$

$$R_e(s, \boldsymbol{a}) \doteq \frac{1}{|\mathcal{C}_e|} \sum_{c \in \mathcal{C}_e} \frac{[v_{\text{Target}} - v_c]_+ \Delta t}{v_c}.$$



Traffic Networks

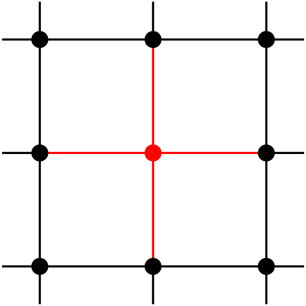
The **3×3 traffic network** [3] problem can be formulated as a graph:

Vertices	The intersections are the vertices $\mathcal{V}$.
Edges	The roads of the network are the edges $\mathcal{E}$.
Cars	A set of $\mathcal{C}$ cars populate the network.

Objective is to *minimise delay:*

$$R(s, \boldsymbol{a}) \doteq \sum_{e \in \mathcal{E}} R_e(s, \boldsymbol{a}),$$

$$R_e(s, \boldsymbol{a}) \doteq \frac{1}{|\mathcal{C}_e|} \sum_{c \in \mathcal{C}_e} \frac{[v_{\text{Target}} - v_c]_+ \Delta t}{v_c}.$$



Traffic Networks

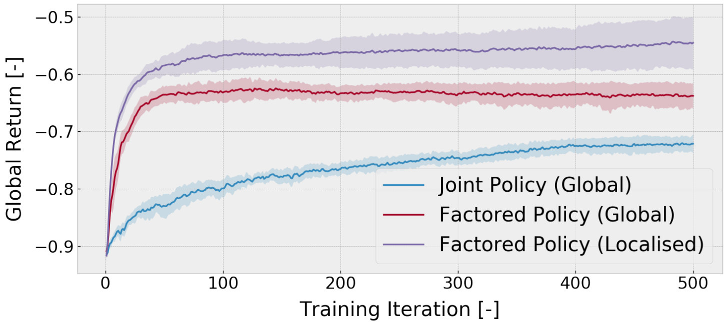


Figure: Learning performance across joint/factored policy distributions, and baselines for the global objective in the 3×3 traffic grid.

Conclusions

Influence networks provide a *unified approach* for encoding structural information into policy optimisation algorithms.

Factored policy gradients *provide tangible benefits over SOTA*.

1. Scalability to concurrent and high-dimensional control problems.
2. No practical increase in complexity – time, sample or cognitive.

FPGs allow us to scale RL to large real-world problems:

1. *Traffic light control* in large networks.
2. *Optimal execution* in multi-venue/multi-asset problems.
3. Learnable policies in *highly parallelised client interaction settings*.

Thank You

[1] Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley.
A Survey of Multi-Objective Sequential Decision-Making.
JAIR, 48:67–113, 2013.

[2] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour.
Policy Gradient Methods for Reinforcement Learning with Function Approximation.
In *Proc. NeurIPS*, pages 1057–1063, 2000.

[3] Eugene Vinitsky, Aboudy Kreidieh, Luc Le Flem, Nishant Kheterpal, Kathy Jang, Cathy Wu, Fangyu Wu, Richard Liaw, Eric Liang, and Alexandre M Bayen.
Benchmarks for reinforcement learning in mixed-autonomy traffic.
In *Proc. of CoRL*, pages 399–409. PMLR, 2018.