The Laplacian Keyboard: Beyond the Linear Span

Siddarth Chandrasekar
Marlos C. Machado

CIFAR
UNIVERSITY OF ALBERTA

Behavioral Foundation Models (BFMs)

Pre-trained without rewards to learn a whole family of behaviors, one for every reward it can represent

$\phi : \mathcal{S} \to \mathbb{R}^k$ — A learned representation

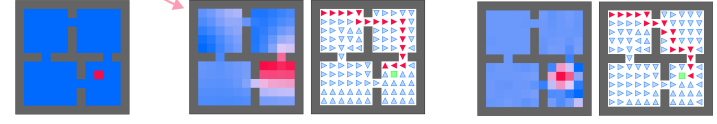
$r(s) = \phi(s)^\top \mathbf{w}$ — Every $\mathbf{w}$ defines a reward function

$\pi_{\mathbf{w}}(a \mid s)$ Learn the optimal policy for every $\mathbf{w}$

Zero-shot Solution

Projection of the reward function onto the learned representations

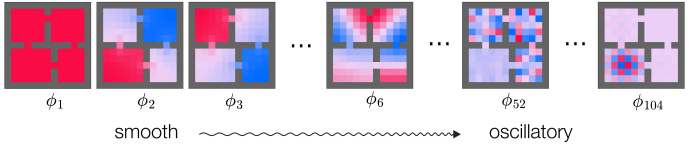
$$\mathbf{w}_r = \arg\min_{\mathbf{w}} \mathbb{E}_{s\sim\mathcal{D}} \left[\left(r(s) - \mathbf{w}^\top \phi(s) \right)^2 \right]$$



Laplacian Representation

A representation that captures how the states are connected

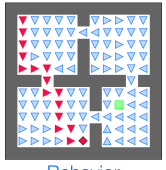
$$\min_{\phi} \ \mathbb{E}_{s,s'\sim\mathcal{D}} \left[\|\phi(s) - \phi(s')\|_2^2 \right] \quad \text{s.t.} \quad \mathbb{E}\left[\phi(s)\phi(s)^\top\right] = I$$

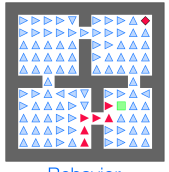


A BFM is a very large library of behaviors. Why restrict to executing one single behavior?

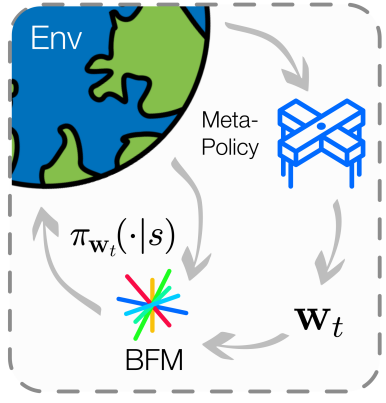
Laplacian Keyboard

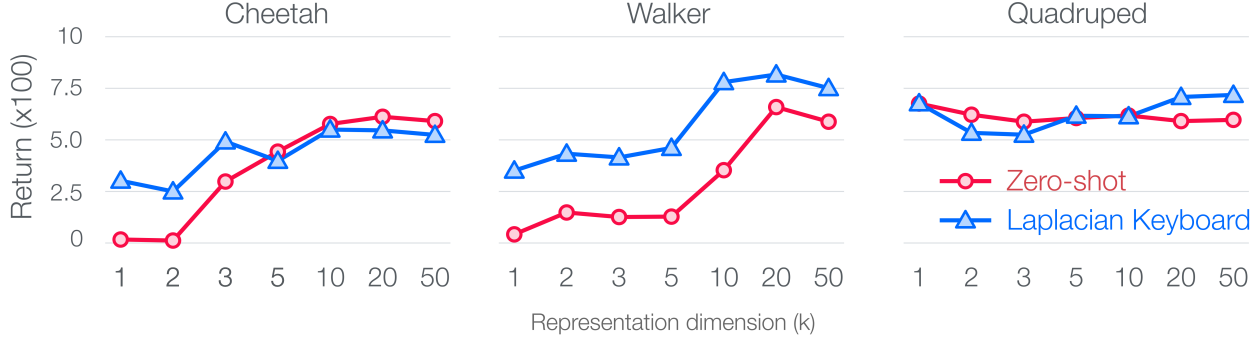
A meta-policy $\pi_{\text{LK}} : \mathcal{S} \to \mathbb{R}^k$ learns to pick and stitch the underlying behaviors


Behavior induced by $\mathbf{w}_1$


Behavior induced by $\mathbf{w}_2$

π_{LK} is trained with RL on the task reward: its action is a $\mathbf{w}$, which the BFM executes via $\pi_{\mathbf{w}}$





TAKEAWAYS

1. *Larger k doesn't reliably improve zero-shot performance*
2. *With k < |S|, no fixed representation captures every task*
3. Stitching beats a single w, especially at small dimensions
4. Stitching behaviors lets the agent act beyond the linear span

Paper

siddart2@ualberta.ca