Trajectory-Level Data Augmentation for Offline Reinforcement Learning

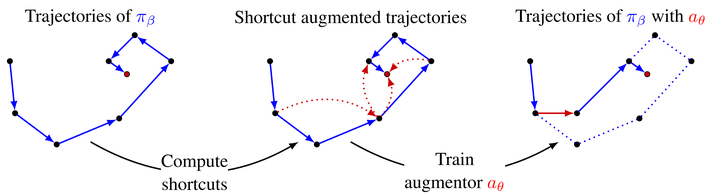
Tobias Schmähling, Matthias Burkhardt, Tobias Windisch
University of Applied Sciences Kempten, Germany

Problem and Core Idea

Motivation: Offline RL performance is strongly determined by dataset quality. In active positioning, logging policies are often highly structured and suboptimal, which makes learning robust offline RL agents difficult.

Our idea: A trajectory-level data augmentation that exploits the geometric interplay in active positioning problems by accumulating actions along a trajectory into a single *shortcut* (SC) action.

Contributions: Better trajectory returns, stronger offline RL training, and a theoretical explanation of why shortcuts work and lead to better trajectories.

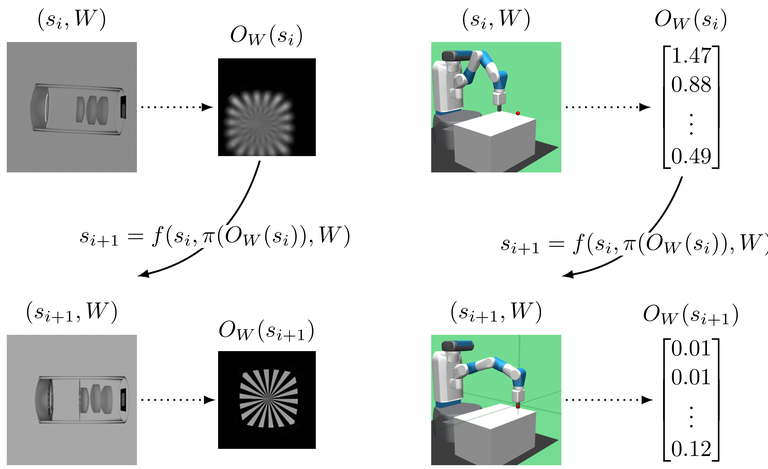


Active Positioning Setup

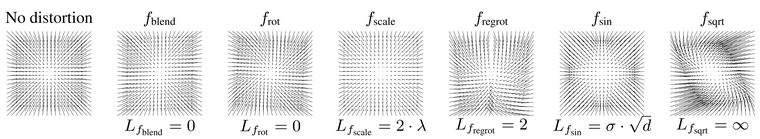
State (s, W) with position $s \in \mathcal{P}$ and static context $W \in \mathcal{W}$. Actions $a \in \mathcal{A}$ are transformed by movement distortion f:

$$s' = f(s, a, W), \qquad r = -\|s' - s_W\|.$$

Goal: reach s_W in as few steps as possible under partial observability $o = \mathcal{O}_W(s)$.



Distortions used in experiments
(key quantity: linear placement error constant L_f)



LIFT: Theory and Algorithm

Shortcut condition (Theorem, informal) If π is distance-improving, V^π is L_V-Lipschitz, and f has placement error constant L_f, then the accumulated action $a = \sum_{k=i}^{j-1} a_k$ is beneficial when

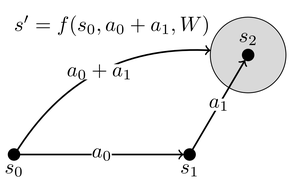
$$\gamma V^\pi(s_j, W) - V^\pi(s_i, W) - \|s_j - s_W\| \geq (\gamma L_V + 1) L_f \sum_{k=i}^{j-1} \|a_k\|.$$

Policy improvement Applying an augmentor shortcut a_θ *only* where it is a π_β-shortcut never hurts: $J(\pi_{\text{aug}}) \geq J(\pi_\beta)$.

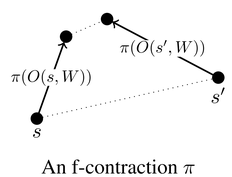
Algorithm 1: Shortcut sampling

Require: $C \geq 0$, $i \in [n]$, $\{(o_0, a_0, r_0), \ldots, (o_n, a_n, r_n)\}$
Ensure: Tuple $(o_i, \hat{a}, r_{j-1}, o_j)$

Compute returns $G_0 \ldots, G_n$ for trajectory
$S = ()$
for $j = i + 1 \cdots n$ **do**
  $\hat{a}_i \leftarrow \sum_{k=i}^{j-1} a_k$
  if $\gamma G_j - G_i + r_{j-1} \geq C \cdot \sum_{k=i}^{j-1} \|a_k\|$ and $\hat{a}_i \in \mathcal{A}$ **then**
    Add $(o_i, \hat{a}_i, r_{j-1}, o_j)$ to S
  end if
end for
Let $m = |S|$ and let $\hat{r} = (\hat{r}_1, \ldots, \hat{r}_m)$ be the rewards of the tuples in S
Let $\rho \sim \hat{r} - \min_i \hat{r}_i$ a mass function
Sample $(o_i, \hat{a}_i, r_{j-1}, o_j)$ from S w.r.t. ρ
return $(o_i, \hat{a}_i, r_{j-1}, o_j)$



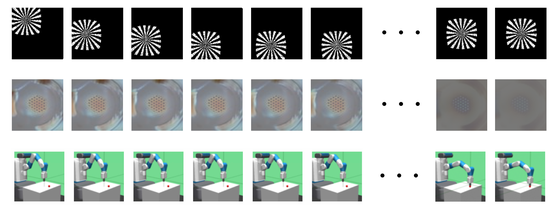
Movement uncertainty



An f-contraction π

Observation Modalities

We evaluate across direct state observations and image-based sensing. LIFT is designed to stay compatible with deterministic logging routines and robust hand-offs.



Each row shows representative trajectories from start to near-goal behavior.

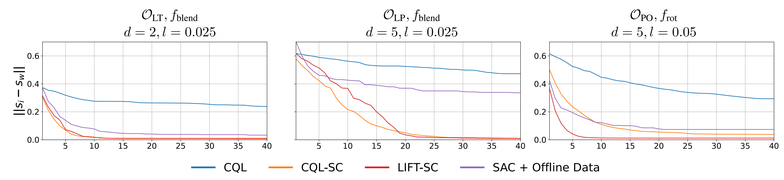
Results and Takeaways

Empirical findings:

- Shortcut training (CQL-SC) improves CQL across scenarios.
- Data collected with Shortcut consistently improves downstream offline RL.
- LIFT-SC is the strongest method in most settings.

Why it helps: Shortcuts improve data quality and diversity by breaking up the highly structured logging data.

Why LIFT helps even more: Applying shortcuts already *during data collection* lets a shortcut-trained policy shape the trajectories as they are gathered, so the data is better from the start.



Acknowledgements and Code

Code: `github.com/HS-Kempten/lift`

Federal Ministry of Research, Technology and Space

Funded by the German Federal Ministry of Research, Technology and Space (BMFTR), project 13FH605KX2.