Shortcut MixUp Policy: Toward Improving Robustness and Speed in Goal-Conditioned RL

Matt Hyatt [1,2] Yassir Atlas [1] Hal Brynteson [1,4] Diego A. Roa Perdomo [1,5] Athena Angara [1] Mengjiao Han [1] Joseph Insley [1] Janet Knowles [1] Yongho Kim [1]
Victor Mateevitsi [1,4] Michael E. Papka [1,4] Silvio Rizzi [1] George K. Thiruvathukal [1,2] Nicola Ferrier [1,3]

[1]Argonne National Laboratory [2]Loyola University Chicago [3]University of Chicago [4]University of Illinois Chicago [5]University of Delaware

OGBench Environment

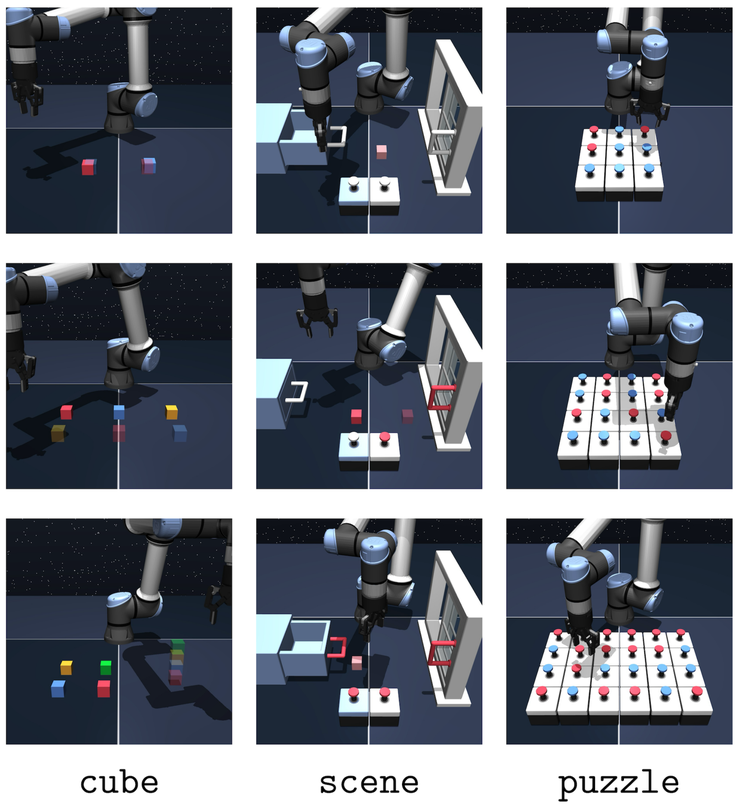

Figure 1. Manipulation Environments from the Offline Goal-Conditioned Benchmark - **OGBench** [2]. The task is specified by a target goal state that the policy must reach by interacting with the environment.

Policy Shortcut Guidance

The model uses classifier-free guidance (CFG) to produce actions that are more closely aligned to the target goal state than goal-conditioned models.

$$v = v_{\text{unc}} + \lambda\left(v_g - v_{\text{unc}}\right) + \mu\left(v_{\text{gdt}} - v_g\right) \tag{1}$$

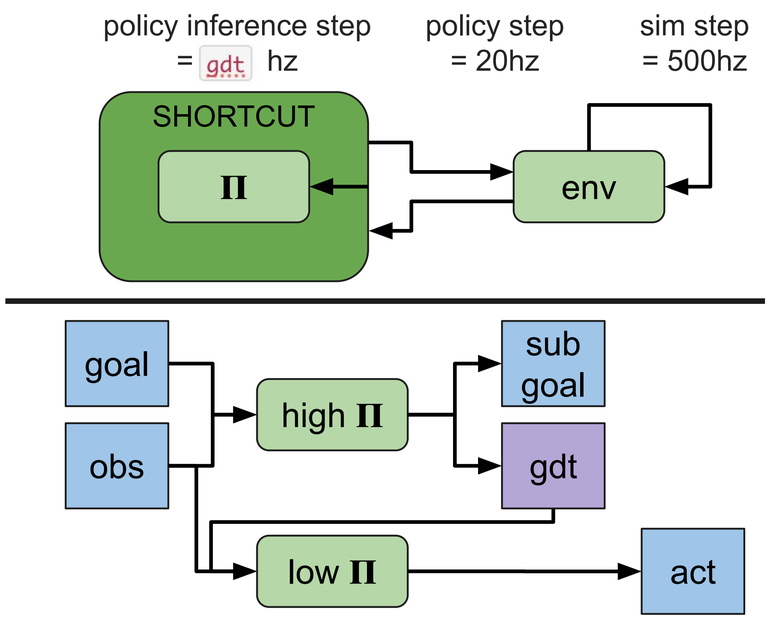

Figure 2. In the hierarchical shortcut model, actions size and subgoals are selected by the high-level model, while the low-level model produces an action guided by the step size.

Additionally guidance is used to align the action distribution with the subset of possible actions that take gdt steps toward the goal. We add an loss term during training to incentivize the model to produce actions conditioned on a $gdt \in [0, 1]$ where 1 represents the maximum shortcut size. This bootstrap loss, L_{boot} uses self-generated targets such that

$$s_{\text{target}} = \frac{1}{2}s_\theta(x_t, s_t, g_t, t, d) + \frac{1}{2}s_\theta(x_{t+d}, t, d) \tag{2}$$

at 2 points in time t and $t + d$ during a given episode. In practice, our model processes state s_t, goal g_t, and the noisy action x_t unlike in image generation tasks that only see x_t.

Goal Stitching

Pro. Goal stitching allows models to learn viable paths from arbitrary offline data.

Con. Goal stitching does not teach the optimal path to the goal. We would prefer to distill a policy that does not need to reach irrelevant states.

Why have speed?

In **sparse reward settings**, long-horizon is a critical barrier to the offline learning process.

In [3] the authors show that while the TD error of the value function is low, there is increasingly large errors in the ground truth q-value of any given state.

Why can a model that accurately predicts the TD value be such a poor predictor of the true state-value? We hypothesize that value function bootstrapping introduces problems that are only a hindrance during long-horizon sparse reward tasks.

Some works aim to learn a dense reward function from sparse signals. Conversely, shortening the horizon might also reduce q-value error.

Preliminary Results

Policy shortcuts speed up execution

Figure 3. We train the low-level policy using an offline dataset of 650M state transitions. The low-level model completes the task up to 2x as fast as the naive goal-stitching model- CFGRL [1].

Indiscriminate policy shortcuts reduce task success rate (SR)

Strategy	SR %
No shortcut	44.00
1-2	28.00
1-4	14.67
1-8	06.67

Table 1. Low-level shortcut policy trained on up to 128 shortcut steps. At inference time, shortcuts are randomly selected with geometric distribution (highest probability given to small/no shortcut)

Hierarchical shortcut selection is not enough

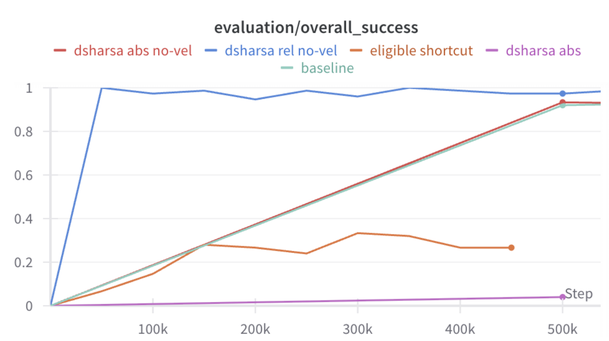

Figure 4. We train the hierarchical model, and observe suboptimal success rate (SR) compared to vanilla model.

We train models with relative and absolute action spaces. Interestingly, relative actions significantly outperform unless the model is both hierarchical and the observation space does not contain joint velocity.

State Mixup

We hypothesize that policy shortcuts reduce success rate through the following **Failure Modes**:

1. **Shortcut Forecasting.** Without **online** data, the model cannot determine if a shortcut would lead to failure.
2. **New State Visits.** Excessive shortcuts cause the model to enter states not seen during training, which leads to undesirable actions on the subsequent timestep.

To address failure mode 1, we implement C-MixUp [5], a form of data regularization designed to create smooth decision boundaries in the latent space [5, 6, 4].

To illustrate C-MixUp, we begin with a debug dataset of 1000 points $(x, y) \in [0, 1]$ where color (r, g, b) is determined by the location of the point. We use the Kernel Density (KDE) sampler from C-MixUp to produce a matrix of sampling probabilities (Fig.1).

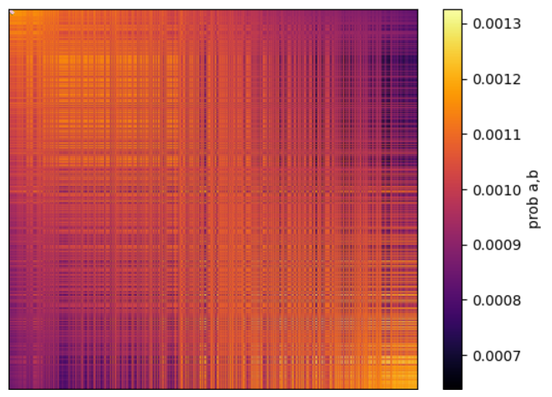

Figure 5. Matrix of input-output pairs, where color represents the relative similarity between the output values. Output similarity is the sampling probability for candidate MixUp pairs.

In Fig.1, we show the plotted points, and draw a graph, where edges show the sampling frequency as edge thickness. The new MixUp points will be pseudolabeled with linearly interpolated output values (r, g, b) from their contributors.

$$\tilde{x} = \mathring{u}x_i + (1)\mathring{u}x_j, \qquad \tilde{y} = \mathring{u}y_i + (1)\mathring{u}y_j \tag{3}$$

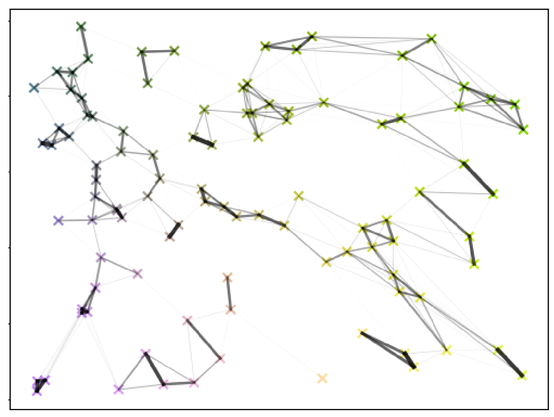

Figure 6. Subset of 100 x,y points and their respective colors. Edges between points represent MixUp likelihood **and also illustrate the new x,y values of mixed samples**.

References

[1] Kevin Frans, Seohong Park, Pieter Abbeel, and Sergey Levine. Diffusion guidance is a controllable policy improvement operator, 2025. URL `https://arxiv.org/abs/2505.23458`.

[2] Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations (ICLR)*, 2025.

[3] Seohong Park, Kevin Frans, Deepinder Mann, Benjamin Eysenbach, Aviral Kumar, and Sergey Levine. Horizon reduction makes rl scalable. *arXiv preprint arXiv:2506.04168*, 2025.

[4] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states, 2019. URL `https://arxiv.org/abs/1806.05236`.

[5] Huaxiu Yao, Yiping Wang, Linjun Zhang, James Zou, and Chelsea Finn. C-mixup: Improving generalization in regression, 2022. URL `https://arxiv.org/abs/2210.05775`.

[6] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018. URL `https://arxiv.org/abs/1710.09412`.

Acknowledgments

This work was supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (ASCR) under the EXPRESS initiative "Harnessing Technology Innovations to Accelerate Science through Visualization" (DOE-145-SE-DAIMSL, NF-24).

This research used resources of the Argonne Leadership Computing Facility, a U.S. Department of Energy (DOE) Office of Science user facility at Argonne National Laboratory and is based on research supported by the U.S. DOE Office of Science-Advanced Scientific Computing Research Program, under Contract No. DE-AC02-06CH11357.

TLDR: we [1] use bootstrapped policy actions to shortcut irrelevant states (only some states are critical to task completion) and [2] use state-goal-action MixUp to improve robustness in unseen scenarios.