On the Role of Proposal Support in Diffusion-Based Offline RL for ARC-AGI Sequential Decision-Making

Yunho Kim[1] Jaehyun Park[1] Heejun Kim[1] Sejin Kim[1] Byung-Jun Lee[2] Sundong Kim[1]

[1]Gwangju Institute of Science and Technology [2]Korea University

ICML International Conference On Machine Learning

Motivation

ARC-AGI tasks require inferring a transformation rule from input–output demonstrations and applying it step by step to a test input. Unlike tasks with direct input–output mappings, success depends on constructing a coherent sequence of intermediate actions where errors accumulate and each step constrains the next. The explicit intermediate state transitions allow us to measure the contribution of each decision step independently, enabling fine-grained failure analysis beyond final-outcome evaluation.

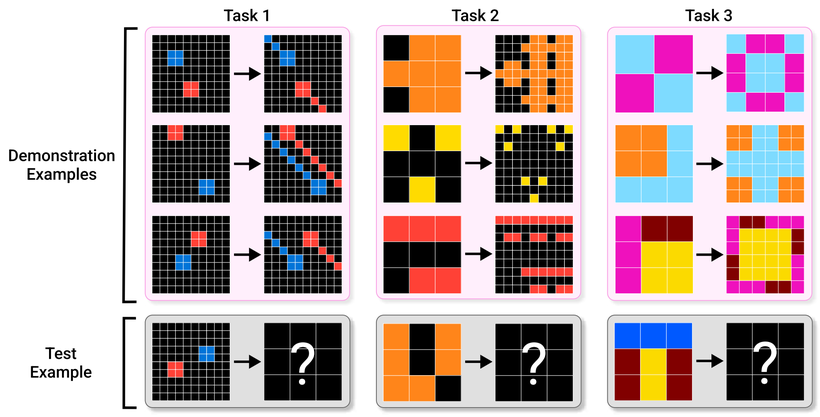


Figure 1. An ARC-AGI task. Input–output demonstrations define the transformation rule. The agent applies it step by step to the test input.

SOLAR: Trajectory Dataset for Offline RL

SOLAR (Synthesized Offline Learning data for Abstraction and Reasoning) provides explicit state–action trajectories for ARC-AGI tasks. Dataset composition is controlled by mixing ratio α where $\mathcal{D}(\alpha) = \alpha\,\mathcal{D}_{\text{exp}} + (1-\alpha)\,\mathcal{D}_{\text{sub}}$. Suboptimal trajectories deviate at a random step via two task-plausible perturbations, either modifying the selection region while keeping the operation fixed or replacing a color-changing operation while preserving the selection.

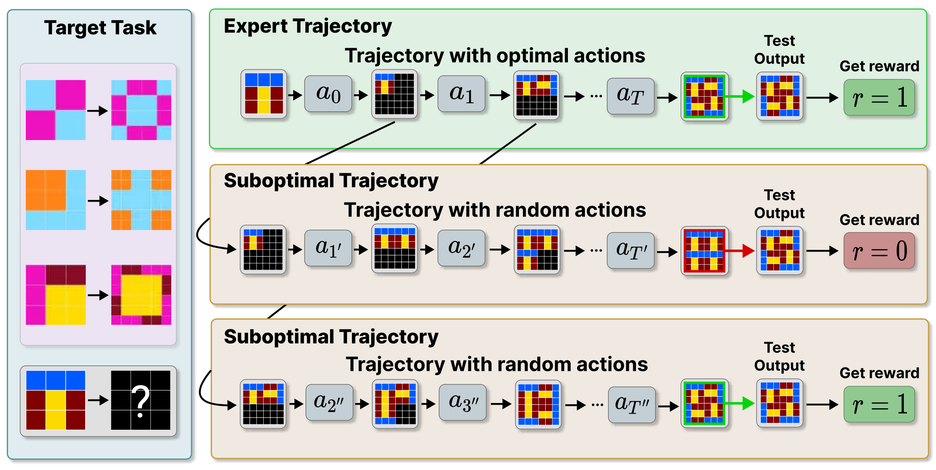


Figure 2. SOLAR episodes. Suboptimal trajectories branch at a random step via task-plausible perturbations (region or color-operation swap).

LDCQ (Latent Diffusion Constrained Q-Learning)

Diffusion-based offline RL can be viewed as a **proposal–selection** pipeline, but the tightly coupled components make it difficult to attribute failure to candidate generation versus value-based selection. LDCQ directly instantiates this pipeline via a β-VAE encoder, a diffusion prior for proposals conditioned on s_t, and a Q-network for selection, making each stage independently diagnosable.

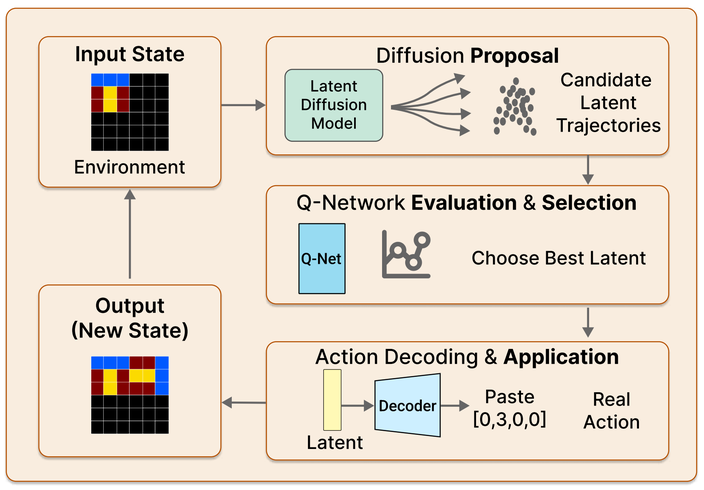


Figure 3. The LDCQ pipeline. A β-VAE encodes trajectory segments, a diffusion prior proposes latent candidates conditioned on s_t, a Q-network selects the highest-value latent, and the VAE decoder maps it to an executable action.

Experimental Results

LDCQ achieves the highest accuracy under expert data (α=1), but its drop under mixed quality (α=1/2) is notably larger than CQL and IQL, suggesting the LDCQ is more sensitive to distributional shift.

Task	Acc.	Task	Acc.
5c0a986e-fix	100%	a65b410d	100%
4c4377d9	98%	6f8cd79b	91%
4258a5f9	79%	007bbfb7	77%
46442a0e	76%	5c0a986e-diff	68%
74dd1130	24%	0d3d703e	0%

Table 1. Single-task accuracy ($\alpha = 1$). `fix`: color mapping fixed across all examples; `diff`: mapping inferred per demonstration pair. `0d3d703e` (0%): requires global color permutation, not recoverable from step-level transitions.

Method	α=1	α=1/2
BC	38.8%	9.7%
CQL	61.9%	37.5%
IQL	58.3%	32.4%
LDCQ	**66.9%**	23.7%

Table 2. Multi-task accuracy. LDCQ leads under expert data (66.9%) but degrades sharply under mixed quality (23.7%), below CQL and IQL.

Selection Enables Efficient Path Stitching

The SOLAR dataset for task `a65b410d` contains two complementary trajectories. T_1 takes an optimal path in the first half but an inefficient path in the second, and T_2 does the reverse. No single training trajectory is fully optimal end-to-end. A fully efficient solution can only be constructed by combining efficient segments from different trajectories.

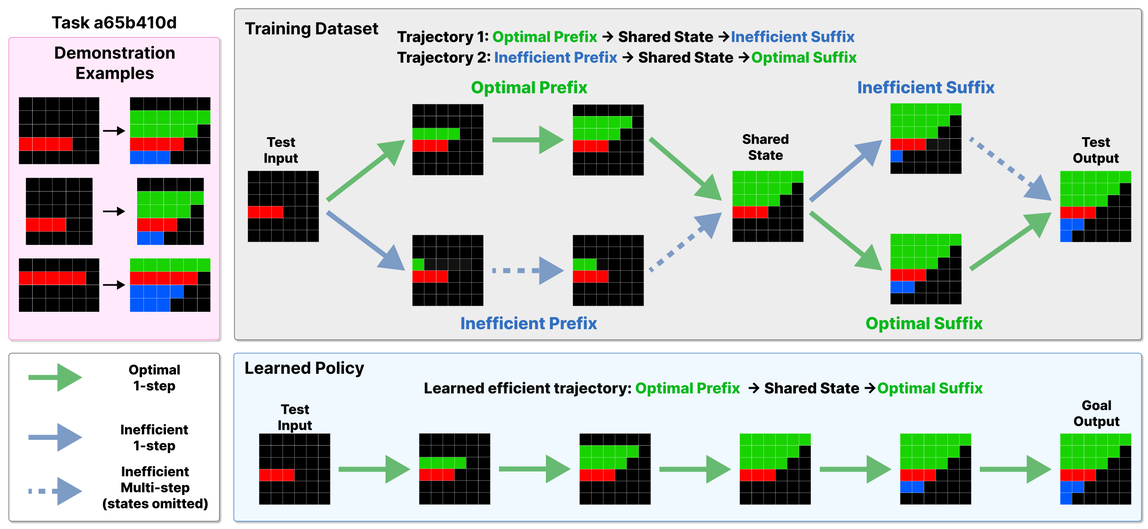


Figure 4. Task `a65b410d`. T_1 is efficient in the first half, T_2 in the second. LDCQ's Q-network stitches the optimal segments end-to-end.

Policy	Acc.	Avg. Steps
VAE Prior	90%	19.4
Diff. Prior	99%	15.4
LDCQ	100%	**7.0**

Table 3. Generative priors match accuracy but follow training trajectories; LDCQ's Q-network stitches efficient segments across T_1 and T_2, requiring $2-3\times$ fewer steps.

Diagnosing Failure: Coverage vs. Selection

We include the ground-truth latent (**oracle**) in the proposal set and separately measure **coverage** (whether the oracle appears among proposals) and **conditional selection accuracy** (whether the Q-network selects it when present).

	Expert	Mixed
P(oracle $\in$ proposals)	91.0%	62.2%
P(correct | oracle present)	95.6%	88.4%

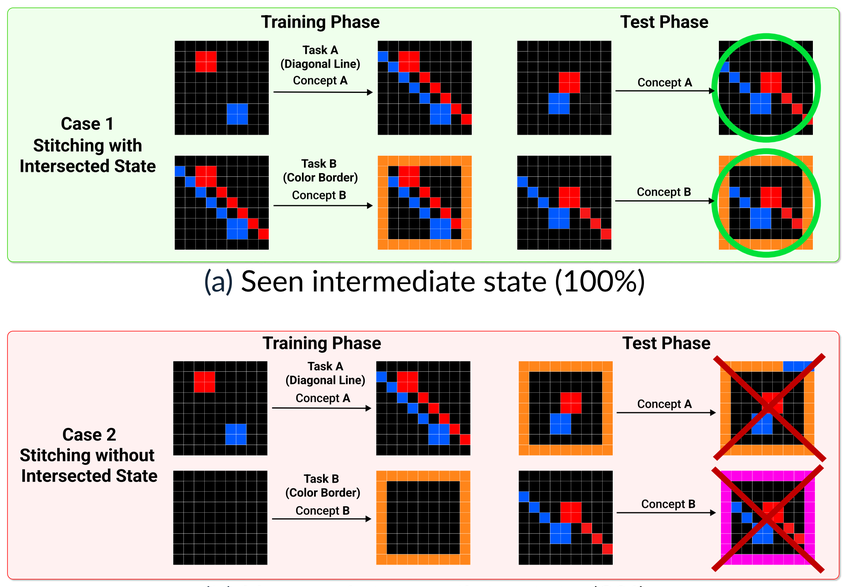
Table 4. **Coverage** (Row 1) is the fraction of states where the correct latent appears among the 100 proposals. **Cond. sel.** (Row 2) is the Q-network's selection accuracy when the correct candidate is present.

# Oracle	Expert	Mixed
0	9.0%	**37.8%**
1–20	1.4%	37.2%
20–50	2.4%	8.4%
50–99	10.8%	6.8%
100	76.4%	9.8%

Table 5. Oracle latents per state (100 proposals). Under mixed data, **37.8%** of states have none.

Compositional Generalization

LDCQ is trained on Task A (draw diagonal line) and Task B (apply color border) independently, then evaluated on their sequential composition requiring both in order. Composition succeeds when the intermediate state at the composition boundary is covered by training data, but fails when it is not, even though each transformation has been learned individually. Performance degrades to 0% (Task A) and 33% (Task B) when each transformation is applied to an unseen intermediate state.



(a) Seen intermediate state (100%)

(b) Unseen intermediate state (0%)

Figure 5. Generalization is constrained by trajectory-level support, as composition fails without the bridging state in the proposal distribution.

Conclusion & Future Directions

- The proposal stage is the primary bottleneck. Under mixed data, 37.8% of states have no correct candidate among proposals, while selection remains reliable (88.4%) when candidates are present.
- Compositional generalization fails when intermediate states fall outside the training distribution. Trajectory-segment latents encode fixed behaviors and lack the modularity needed for recombination.
- Latent reasoning from offline trajectories faces fundamental limits in coverage and extensibility. Modular latent structures and hybrid offline-to-online approaches are key directions.

GitHub

ARC Prize

rladbsgh797210@gmail.com