MUST DECISION MAKING BE TRAJECTORY-CENTERED? BEHAVIORAL STRUCTURE AS A RELATION-CENTERED ALTERNATIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning is commonly learned from trajectories: finite sequences generated by interactions between an agent and its environment. Yet trajectories are samples of realized behavior, not necessarily the underlying structure of what behaviors are possible. We ask whether decision making must be organized around observed trajectories, or can instead be organized around the behavioral relations revealed by them. We introduce **Behavioral Structure**, a relation-centered framework that represents action-conditioned relations among states and composes them directly for decision making. On sparse long-horizon goal-directed control, Behavioral Structure reaches **32% success versus 17% for goal-conditioned Decision Transformer and 13% for Decision Transformer** on Ant. When training trajectories are deliberately fragmented across withheld behavioral boundaries, Behavioral Structure achieves **32% versus 2% for endpoint-conditioned Decision Diffuser**, while retaining **36% success at composition depths 4–5 versus 0%**. These results support a view in which trajectories describe realized experience while behavioral relations provide an alternative substrate for organizing decisions.

1 INTRODUCTION

Reinforcement learning (RL) learns from experience that is typically observed as **trajectories**: sequences of states, actions, and rewards. This organization underlies value-based offline RL such as CQL and IQL (Kumar et al., 2020; Kostrikov et al., 2021), sequence models such as Decision Transformer (DT) (Chen et al., 2021), and generative trajectory models (Janner et al., 2022; Ajay et al., 2023). Across these formulations, trajectories provide the empirical record from which behavior is learned.

But a trajectory describes **what behavior was realized**, not necessarily the structure of **what behaviors are possible**. Each trajectory is one realization under a particular policy and initial condition. This distinction suggests that the form in which experience is observed need not be identical to the representation that ultimately organizes decision making.

This leads to our central question:

Must decision making be trajectory-centered?

We investigate an alternative in which trajectories provide observations of behavior, while **behavioral relations**—whether, and through what action, one state can lead toward another—provide reusable decision primitives. We call the resulting organization **Behavioral Structure**. In our implementation, transition-supervised representation learning proposes a sparse local topology, an action-conditioned model evaluates candidate relations, and standard shortest-path planning composes the weighted relations toward a specified goal.

The distinction is between an **observation unit** and a **representation/composition unit**. If useful behavioral evidence remains reusable outside the complete trajectory in which it was observed, decision making should remain possible even when training trajectories are deliberately fragmented. Compositional reuse is therefore not our motivation but a **stress test of the representation hypothesis**.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

On standard sparse goal-directed Ant control, Behavioral Structure achieves **32% success**, compared with **17% for goal-conditioned DT and 13% for DT**. More importantly, we construct a controlled withheld-boundary evaluation in which training trajectories are split and the transitions joining adjacent fragments are removed. Under exactly matched queries with no withheld-boundary leakage, Behavioral Structure achieves **32% success versus 2% for endpoint-conditioned Decision Diffuser**; at composition depths four and five it retains **36%**, while Decision Diffuser obtains **0%**. Geometric graph controls further show that geometric organization alone is insufficient for successful execution.

Our contributions are threefold:

1. We formulate a **relation-centered representation hypothesis**: trajectories can be observation units without necessarily being the primary units through which decisions are represented and composed.
2. We introduce **Behavioral Structure**, which represents local action-conditioned behavioral relations and composes them directly for goal-directed decision making.
3. We introduce a **withheld-boundary stress test** that deliberately removes complete trajectory continuity while preserving local behavioral evidence, and provide empirical evidence that relation-centered composition remains effective under this intervention.

2 Problem Formulation

2.1 From Trajectory Observations to Behavioral Relations

Consider an offline dataset

$$\mathcal{D} = \{\tau_i\}_{i=1}^{N}, \qquad \tau_i = (s_0, a_0, r_0, s_1, \ldots, s_T).$$

A trajectory is a finite realization of behavior generated under a particular policy, initial condition, and environment interaction. It therefore records *what happened*. It does not, by itself, specify the complete structure of behavioral possibilities supported by the collected experience.

This distinction motivates a separation between the form in which experience is observed and the form in which it is represented for decision making:

$$\boxed{\text{observation unit } \neq \text{ representation/composition unit.}}$$

Trajectories are the observation units available in the dataset. Our central hypothesis is that they need not also be the fundamental units from which a decision is represented and composed.

For states s_i, s_j and action a, let

$$\mathcal{R}(s_i, a, s_j)$$

denote *local behavioral support*: evidence that taking action a at s_i is behaviorally consistent with reaching toward s_j. In this work, $\mathcal{R}$ is neither a calibrated estimate of the transition probability $P(s_j \mid s_i, a)$ nor a direct estimate of multi-step reachability. It is a local, action-conditioned relation inferred from behavioral observations.

This gives three distinct objects:

$$\boxed{\text{trajectory observation } \longrightarrow \text{ local behavioral relations } \longrightarrow \text{ behavioral structure.}}$$

The first is how experience is observed; the second is the local unit of behavioral evidence; the third organizes such units so that they can be composed for decision making. Conflating these levels would reduce Behavioral Structure to a particular graph construction. Our claim is instead about the choice of representation unit.

2.2 Behavioral Structure as a Compositional Object

We organize candidate local relations into a directed weighted structure

$$\mathcal{B} = (V, E, c),$$

where vertices correspond to observed state instances, directed edges represent candidate action-bearing relations, and c assigns relation costs. The graph used in our implementation is therefore a concrete realization of $\mathcal{B}$, rather than the conceptual contribution by itself.

Crucially, global reachability is not directly supervised. It arises operationally through composition of locally supported relations:

$$\boxed{\text{local behavioral relations} + \text{structural composition} \implies \text{global behavioral reachability.}}$$

Given a start state s_{start} and goal state s_{goal}, decision making searches for a composition of weighted local relations connecting them. Because the composition operates over relations rather than over stored complete trajectories, the resulting decision need not coincide with any trajectory observed in $\mathcal{D}$.

This point also distinguishes a relation-centered representation from simply using a graph to select intermediate states. In Behavioral Structure, a local relation is action-bearing: (s_i, a_i, s_j) associates a source state and an action with a candidate behavioral continuation. Composition therefore acts directly on decision primitives that already contain behavioral information. The shortest-path algorithm used later is only a mechanism for composing these primitives; it is not what defines the representation.

2.3 A Testable Representation Hypothesis

The preceding distinction yields a concrete hypothesis. Suppose a desired behavior is absent as a complete trajectory in $\mathcal{D}$, while the local behavioral relations needed to realize it remain supported by different parts of the dataset. If complete trajectories are necessary composition units, destroying their continuity should strongly restrict the ability to construct the desired decision. If local behavioral relations are reusable composition units, however, the decision may remain expressible through a new composition of locally supported relations.

We therefore distinguish two notions that are easily conflated:

trajectory continuity: the complete behavior occurs as an observed sequence,

relational support: the local behavioral primitives required by the behavior are available.

Behavioral Structure is designed to exploit the second without requiring the first.

This distinction motivates the withheld-boundary intervention in Section 4.3. There, complete trajectory continuity is deliberately destroyed by removing transitions between adjacent fragments, while local behavioral evidence within the fragments is retained. The experiment therefore does not merely ask whether a model can interpolate or generate a new trajectory. It asks whether decision making can remain possible when the *observation-level organization* is disrupted but reusable *relation-level evidence* remains.

The hypothesis has an equally important boundary condition. Relation-centered composition can reorganize behavioral evidence that exists somewhere in the dataset; it does not imply recovery of behavioral primitives for which no support is available. Hence,

$$\boxed{\text{missing trajectory continuity} \neq \text{missing behavioral primitives.}}$$

We test this distinction explicitly in Section 4.5 by removing primitive relations themselves.

Finally, the structure and relation costs used in this paper are reward-independent. The test-time goal specifies a destination but does not alter the learned behavioral structure. This separation is appropriate for the sparse goal-reaching tasks considered here, where successful feasible goal-reaching is aligned with task reward. It should not be interpreted as a general claim that feasibility equals optimality. In richer tasks, task-specific reward or utility can enter the selection or planning layer without requiring the underlying behavioral relations to be redefined.

3 Behavioral Structure

Behavioral Structure consists of four stages: transition-supervised representation learning, sparse candidate topology construction, action-conditioned relation weighting, and structural planning.

3.1 TRANSITION-SUPERVISED REPRESENTATION

The first component constructs a state representation whose local geometry is informed by observed transitions rather than raw state similarity alone. Let $\bar{s}_i$ denote a normalized state. We encode it as

$$z_i = f_\theta(\bar{s}_i),$$

where f_θ is a two-layer MLP with hidden dimension 128, latent dimension 32, and ReLU activation.

The representation is trained from one-step transition evidence. For an observed transition (s_i, a_i, s_i^+), the observed successor s_i^+ provides positive relational supervision, while a randomly substituted target state provides a negative example. Importantly, the representation receives no reward, test-time goal, or explicit multi-step reachability supervision. Its role is therefore deliberately limited: it organizes states according to transition-informed local similarity so that plausible relational candidates can be proposed efficiently.

This separation between *candidate generation* and *behavioral evaluation* is important. Latent proximity alone does not imply that a transition is behaviorally executable. Two states may be close in a learned representation while the action available at the source does not support movement between them. We therefore use the representation only to define a sparse candidate topology; candidate relations are subsequently evaluated using explicit action-conditioned behavioral evidence.

This design also helps isolate the representation question studied in this work. We do not require the encoder itself to recover global reachability. Instead, local transition evidence shapes the geometry, while longer-range behavioral possibilities emerge later through structural composition.

3.2 LOCAL CANDIDATE TOPOLOGY

For every source state with an outgoing transition, we retrieve $k = 12$ nearest neighbors in latent space and explicitly retain its observed successor:

$$E = \{(i, j) : j \in \mathcal{N}_{12}(i) \cup \{i^+\}\}.$$

The graph is directed and all candidates are retained. Thus latent geometry determines where composition may be considered; it does not itself determine which routes are behaviorally preferred.

The observed outgoing action a_i of source i is attached to every candidate (i, j), yielding an action-conditioned candidate relation (s_i, a_i, s_j).

3.3 BEHAVIORAL RELATION WEIGHTING

The relation model receives

$$x_{ij} = [\bar{s}_i, \bar{a}_i, \bar{s}_j]$$

and uses a 256–64 ReLU MLP followed by a sigmoid score

$$q_\phi(\bar{s}_i, \bar{a}_i, \bar{s}_j).$$

Positive examples are observed one-step tuples (s_i, a_i, s_i^+); negatives preserve (s_i, a_i) while replacing the target with a random training state other than s_i^+. Binary cross entropy is used for training. Consequently, q_ϕ is a **discriminative behavioral-support score relative to this negative distribution**, not a calibrated estimate of $P(s_j \mid s_i, a_i)$.

We additionally measure discrepancy from the observed successor:

$$m = \text{median}_n \, \|s_n^+ - s_n\|_2, \qquad e_{ij} = \text{clip}\left(\frac{\|s_j - s_i^+\|_2}{m}, 0, 3\right).$$

The relation cost is

$$c(i, j) = -\log\big(q_\phi(\bar{s}_i, \bar{a}_i, \bar{s}_j) + 10^{-8}\big) + e_{ij}.$$

Neither reward nor the test-time goal enters this cost.

Table 1: Standard goal-directed success rate.

Method	Ant	Hopper
Decision Transformer	13%	9%
Goal-Conditioned DT	17%	9%
Behavioral Structure	**32%**	**14%**

3.4 Structural Planning and Execution

At test time the goal is mapped to its nearest graph state under normalized state distance. Dijkstra search returns

$$P^* = \arg \min_{P: i_s \rightsquigarrow i_g} \sum_{(u,v) \in P} c(u, v).$$

Each selected edge already carries its source state's observed action. These actions are executed sequentially; planning occurs once per query without test-time replanning or a separate subgoal controller.

Thus,

trajectory observations $\rightarrow$ candidate topology $\rightarrow$ action-bearing weighted relations $\rightarrow$ structural composition.

4 Experiments

We test three questions: (1) can Behavioral Structure support ordinary goal-directed control; (2) can behavioral relations be reused when complete trajectory continuity is deliberately removed; and (3) is geometric graph connectivity alone sufficient?

4.1 Setup

We evaluate sparse goal-directed continuous control on Ant-v4 and Hopper-v4. Standard evaluation uses 100 queries with target horizons $H \in \{5, 10, 15, 25, 50\}$, success threshold 0.5, and at most 250 execution steps. Unless stated otherwise, Behavioral Structure uses 100,000 transitions from 200 trajectories.

We compare with Decision Transformer (DT) (Chen et al., 2021) and a goal-conditioned DT (GC-DT) based on the official DT codebase with a minimal goal-conditioning adapter. For the controlled compositional test, we additionally use the Decision Diffuser architecture (Ajay et al., 2023), with endpoint conditioning adapted to our dataset and evaluation. Both methods receive the complete test-time goal state.

4.2 Standard Goal-Directed Control

Behavioral Structure achieves **32%** success on Ant, compared with **17%** for GC-DT and **13%** for DT. On Hopper, the corresponding rates are **14%**, 9%, and 9%. At Ant horizon $H = 50$, Behavioral Structure achieves 35%, versus 5% for GC-DT and 0% for DT.

These results establish that the relational representation supports executable decisions, but standard performance alone cannot isolate the representation hypothesis.

4.3 Withheld-Boundary Evaluation

We therefore construct a controlled intervention. Twenty-eight Ant trajectories are each divided into five disjoint 18-step fragments. Internal transitions remain available, while the four transitions joining adjacent fragments are removed, yielding 112 withheld boundaries and 2,408 training transitions.

Evaluation queries start near the beginning of one fragment and target a state near the end of a later contiguous fragment. Hence the required complete behavior is absent as a complete training trajectory, while local behavioral evidence remains distributed across fragments. We use 25 queries at each composition depth $D \in \{2, 3, 4, 5\}$, for 100 fixed queries.

Table 2: Controlled withheld-boundary evaluation on Ant.

Method	Success	Avg. Dist.	Median Dist.
Raw-State Graph	0%	16.314	15.865
Latent-Similarity Graph	1%	16.313	16.126
Decision Diffuser	2%	13.765	13.527
Behavioral Structure	**32%**	**8.530**	**7.826**

Method	$D = 2$	$D = 3$	$D = 4$	$D = 5$
Decision Diffuser	4%	4%	0%	0%
Behavioral Structure	**36%**	**20%**	**36%**	**36%**

Table 3: Graph controls under the withheld-boundary evaluation.

Method	Execution Success	Avg. Final Dist.
Raw-State Graph	0%	16.314
Latent-Similarity Graph	1%	16.313
Behavioral Structure	**32%**	**8.530**

We compare against endpoint-conditioned Decision Diffuser using its original diffusion and inverse-dynamics architecture with the adapter required for our dataset and evaluation. All 100 queries are exactly matched and no withheld-boundary transition is exposed during training.

Behavioral Structure succeeds on **32%** of queries versus **2%** for Decision Diffuser. At depths four and five it retains **36%**, while Decision Diffuser obtains 0%. We do not claim that diffusion trajectory models are generally incapable of composition; rather, this controlled intervention shows a substantial advantage for explicit relation-centered composition when required behavior crosses withheld boundaries. With only 25 queries per depth, the depth-wise numbers should not be interpreted as evidence of depth invariance.

4.4 Is a Graph Alone Sufficient?

The withheld-boundary result does not by itself establish that behavioral relation weighting is necessary. A simpler explanation is that once observations are organized into a graph, ordinary geometric connectivity may already enable composition. We therefore construct two controls that retain graph-based planning while removing the complete action-conditioned behavioral relation cost.

The **Raw-State Graph** constructs local connectivity from similarity in the original state space. The **Latent-Similarity Graph** uses the transition-supervised representation but plans according to latent similarity rather than the complete behaviorally weighted relation cost.

The execution results show that geometric organization alone is insufficient. The Raw-State Graph obtains 0% execution success, while the Latent-Similarity Graph reaches only 1%, despite using the same general graph-based composition mechanism. Behavioral Structure instead reaches 32% execution success and substantially reduces the average final distance to the goal.

The distinction is therefore not the presence of a graph or shortest-path planning. All three methods organize observations structurally and use graph-based planning. Rather, Behavioral Structure assigns candidate relations behavior-dependent costs using action-conditioned evidence and transition consistency. Latent geometry determines the candidate topology, whereas behavioral evidence determines which candidate relations are preferred during composition.

4.5 Relation Quality and Sparsification

We further test whether the learned weighting identifies useful relations within the same candidate topology. We sparsify the graph and compare **learned-score retention** with **random degree-matched retention**. Both conditions retain approximately the same number of relations; only the selection rule differs.

Table 4: Execution success under relation sparsification.

Edge Density	Random Retention	Learned-Score Retention
10%	0.0%	**25.0%**
25%	0.0%	**30.0%**
50%	1.7%	**25.0%**
75%	9.2%	**26.7%**
100%	26.7%	26.7%

The difference is largest in the sparse regime. With only 10% of candidate relations retained, learned-score retention preserves 25% execution success, whereas random retention obtains 0%. At 25% density, the corresponding rates are 30% and 0%. Behavioral-feasibility scores show the same pattern: at 10% density they are 0.614 versus 0.056, and at 25% they are 0.806 versus 0.211 for learned and random retention, respectively.

Graph connectivity again does not explain the difference. At 25% density, graph planning succeeds on 96.7% of queries under learned retention and 93.3% under random retention, while execution success differs by 30 percentage points. This indicates that the learned relation score is not merely preserving graph connectivity; it preferentially retains relations that are more useful for executable composition.

Interestingly, 25% learned retention slightly exceeds the fully dense structure (30.0% versus 26.7%). We do not interpret this as evidence that 25% density is intrinsically optimal. A more conservative explanation is that sparsification can remove weakly supported alternatives that otherwise compete during shortest-path planning.

An independent 120-query cross-trajectory evaluation shows the same qualitative pattern: Raw-State Graph achieves 0% success, Latent-Similarity Graph 0.8%, and Behavioral Structure 20.8%, with behavioral-feasibility scores of 0.109, 0.181, and 0.822, respectively.

Finally, we test the opposite regime: rather than removing complete trajectory continuity, we remove outgoing relations from 20% of primitive fragments. Behavioral Structure then falls to approximately 1% success despite high graph planning success. This exposes a clear boundary of the method:

missing complete trajectories can be overcome $\neq$ missing behavioral primitives can be overcome.

Behavioral Structure can reorganize and compose locally available evidence; it does not reliably invent behavioral primitives absent from the learned structure.

5 Related Work

Trajectory models and stitching. Decision Transformer casts RL as conditional sequence modeling (Chen et al., 2021), while diffusion-based planners model behavioral trajectories generatively (Janner et al., 2022; Ajay et al., 2023). Cross-trajectory reuse itself is not new: DiffStitch generates stitching transitions for offline-RL data augmentation (Li et al., 2024), while SSD uses conditional diffusion to stitch useful sub-trajectories for goal-conditioned offline RL (Kim et al., 2024). Our claim therefore does not rest on being the first method to recombine experience across trajectories.

Reachability and graph-based planning. Our closest conceptual antecedents combine learned reachability with planning. SoRB constructs a graph over replay-buffer observations, uses a goal-conditioned value function to define edge weights, and searches for subgoals (Eysenbach et al., 2019). LEAP plans sequences of latent subgoals whose reachability is evaluated by goal-conditioned policies/value functions (Nasiriany et al., 2019). Most closely, Graph-Assisted Stitching (GAS) learns a Temporal Distance Representation, clusters states from different trajectories into graph nodes, uses shortest-path search to select subgoals, and trains a low-level policy to reach them (Baek et al., 2025).

These works establish that learned reachability graphs and graph search are not new. The distinction investigated here is the *decision primitive*. SoRB, LEAP, and GAS use graph or latent planning

primarily to identify states/subgoals that a controller should subsequently reach. Behavioral Structure instead composes **action-bearing weighted relations** (s_i, a_i, s_j) directly, with selected relations already specifying the actions used for execution.

Our contribution is not a new graph-search algorithm; we deliberately use standard graph construction and shortest-path machinery to isolate the representation question.

Accordingly, the withheld-boundary experiment is not presented as the discovery of trajectory stitching. It is a controlled stress test of whether local action-bearing relations can serve as the compositional decision primitives after trajectory continuity is removed. In short,

$$\text{trajectory} = \text{observation unit}, \qquad (s_i, a_i, s_j) = \text{composition unit}.$$

This positioning is intentionally narrower than claiming a new notion of reachability, graph planning, or trajectory stitching. Prior work has already established all three as useful ingredients for long-horizon control. Our hypothesis concerns instead *where the compositional semantics reside*: in a sequence of states or subgoals subsequently realized by another controller, or directly in action-bearing behavioral relations that constitute the decision representation itself. The experiments are designed around this distinction rather than around novelty of the underlying graph-search machinery.

6 Discussion and Limitations

Behavioral Structure should be interpreted as a representation hypothesis rather than a claim that trajectories are unnecessary. Trajectories remain the empirical source of behavioral evidence; our question is whether they must remain the unit through which that evidence is represented and composed.

Several limitations are important. First, the current instance-level graph grows with stored states. Second, candidate topology is determined by latent kNN plus forced observed successors; behavioral evidence changes relation costs rather than binary edge existence. Third, candidate edges inherit the observed outgoing action of their source and are executed open-loop, making the method sensitive to execution error and stochastic dynamics. Fourth, the experiments cover a limited set of continuous-control environments. The withheld-boundary evaluation isolates the representation question but does not establish universal superiority over trajectory-centered or graph-based methods. In particular, direct evaluation against the broader graph-based stitching literature under a unified protocol remains future work.

The current structure and planner are reward-independent. This is suitable for our sparse goal-reaching setting but does not imply that reward is unnecessary. Richer tasks could preserve reusable behavioral relations while introducing task-specific utility at the planning layer:

$$\text{behavioral possibilities} + \text{task preference} \rightarrow \text{decision}.$$

Most importantly, our method recombines observed behavioral primitives rather than creating missing ones:

$$\text{missing complete trajectories can be overcome} \neq \text{missing behavioral primitives can be overcome}.$$

6.1 Failure Modes and Scope

The experiments suggest three distinct failure modes. First, *topological failure* occurs when candidate generation does not place a useful relation in the graph; relation weighting cannot recover an edge that was never proposed. Second, *relational failure* occurs when candidate connectivity exists but the learned behavioral cost assigns misleading preferences. The poor execution performance of the geometric graph controls demonstrates that this distinction matters empirically. Third, *execution failure* occurs when a structurally reasonable path is found but open-loop replay of source actions does not realize the intended state sequence.

These failure modes also clarify what the present results do and do not establish. The withheld-boundary experiment primarily tests whether available local behavioral evidence can be reorganized beyond the complete trajectories in which it was observed. It does not test recovery from entirely

missing local behavior, nor does it establish that an instance-level graph is the optimal implementation of relation-centered decision making. More scalable relational representations, learned topology construction, and closed-loop execution are natural extensions of the same representation hypothesis.

7 Conclusion

We asked whether decision making must be trajectory-centered. Behavioral Structure provides one alternative: trajectories supply observations, while local action-conditioned behavioral relations become reusable compositional primitives. The method deliberately uses standard graph and shortest-path machinery so that the central object under study is the representation rather than the search algorithm.

Under a controlled intervention that removes complete trajectory continuity while preserving local behavioral evidence, Behavioral Structure achieves 32% success versus 2% for endpoint-conditioned Decision Diffuser and retains 36% success at composition depths four and five. Together with geometric graph controls, these results provide initial evidence for a broader distinction:

$$\boxed{\text{how behavior is observed} \neq \text{how behavioral evidence must be represented and composed.}}$$

AI Assistance Statement

Generative AI tools were used to assist with research ideation and hypothesis refinement, experimental and methodological design feedback, code-oriented reasoning and auditing, interpretation of experimental results, literature discovery, and drafting and editing portions of the manuscript. AI-assisted outputs were checked against implementation details, experimental results, and relevant literature. The author reviewed the final methodology, claims, experiments, citations, and manuscript and takes responsibility for the final content.

References

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? In *International Conference on Learning Representations*, 2023.

Seungho Baek, Taegeon Park, Jongchan Park, Seungjun Oh, and Yusung Kim. Graph-assisted stitching for offline hierarchical reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 2391–2408, 2025.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging planning and reinforcement learning. In *Advances in Neural Information Processing Systems*, 2019.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Sungyoon Kim, Yunseon Choi, Daiki E. Matsunaga, and Kee-Eung Kim. Stitching sub-trajectories with conditional diffusion model for goal-conditioned offline rl. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13160–13167, 2024.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Guanghe Li, Yixiang Shan, Zhengbang Zhu, Ting Long, and Weinan Zhang. DiffStitch: Boosting offline reinforcement learning with diffusion-based trajectory stitching. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 28597–28609, 2024.

Soroush Nasiriany, Vitchyr Pong, Steven Lin, and Sergey Levine. Planning with goal-conditioned policies. In *Advances in Neural Information Processing Systems*, 2019.

A Additional Experimental Results

A.1 Additional Cross-Trajectory Evaluation

In an independent 120-query cross-trajectory evaluation, Raw-State Graph achieves 0% success, Latent-Similarity Graph 0.8%, and Behavioral Structure 20.8%. Their behavioral-feasibility scores are 0.109, 0.181, and 0.822, respectively.

A.2 Missing Behavioral Primitives

When outgoing relations from 20% of primitive fragments are removed, Behavioral Structure succeeds on approximately 1% of queries despite high graph planning success. This experiment distinguishes recombination of available local evidence from recovery of primitives absent from the learned structure.

B Implementation Details

Candidate topology is constructed using directed latent k-nearest-neighbor retrieval with $k = 12$, with each observed successor explicitly retained. All candidates remain in the graph. The action-conditioned discriminative score and transition discrepancy determine relation cost but do not prune edges.

The relation model is a 256–64 ReLU MLP trained by binary cross entropy. Each observed transition supplies a positive example, while one random target replacement supplies a negative example per positive per epoch.

Planning uses Dijkstra search without a heuristic. The goal is introduced only when selecting the destination graph state. Actions bound to selected source states are then executed open-loop without replanning.