GRALP: A Generative Representation Framework for Action Refinement and Latent Planning in Offline Robotic Control

Talha Zaidi [1] , **Arslan Munir** [2] , **Sardar Ali Abbas** [2]
[1]Department of Computer Science, Kansas State University
[2]Department of Electrical Engineering and Computer Science, Florida Atlantic University
tzaidi@ksu.edu, arslanm@fau.edu, abbass2024@fau.edu

Abstract

Offline robotic control requires long-horizon reasoning from fixed datasets while avoiding unsafe extrapolation beyond demonstrated behavior. We propose GRALP, a principled framework that resolves this tension by jointly enforcing support preservation and controllability at the level of temporal abstraction. GRALP adopts a deliberate architectural separation: diffusion is used exclusively as a deterministic action decoder for executing fixed latent skills, while planning and value estimation operate entirely in latent space under conservative constraints. This design enables stable value learning, controllable skill composition, and efficient planning without trajectory-level diffusion sampling at inference. Across unified D4RL benchmarks, GRALP achieves the highest average performance on Navigation, Sequential (Kitchen), and Adroit domains while remaining competitive on locomotion tasks. On contact-rich RoboSuite manipulation with human demonstrations (Lift and Pick-and-Place), GRALP achieves consistently high success rates (over 94%). These results indicate that reliable long-horizon offline control emerges when expressivity is confined to execution and decision-making operates over support-aligned latent abstractions.

1 Introduction

Learning robotic control policies from fixed datasets is increasingly important in settings where online exploration is unsafe, costly, or physically irreversible, such as contact-rich manipulation, constrained navigation, and long-horizon task execution. In offline settings, agents must compose multi-step behaviors solely from logged demonstrations while avoiding actions that deviate from the physically valid and safe behavior distribution. This creates a central challenge: long-horizon reasoning is essential for task success, yet extrapolation beyond demonstrated support can induce severe robotic failures, including unstable contacts, unsafe transitions, and geometric constraint violations. These constraints reflect open-world deployment settings, where embodied AI systems must operate under limited supervision, perceive contact-rich physical interactions [Zaidi *et al.*, 2026], and compose demonstrated behaviors in novel configurations while strictly avoiding unsafe extrapolation.

Prior offline reinforcement learning (RL) methods address distributional shift through conservative value estimation or behavior regularization (e.g., BCQ [Fujimoto *et al.*, 2019], CQL [Kumar *et al.*, 2020]), but often restrict expressivity, limiting multi-modal robotic behaviors. Implicit Q-Learning (IQL) [Kostrikov *et al.*, 2021; Hansen *et al.*, 2023] relaxes explicit conservatism but typically relies on unimodal policy representations. Sequence-modeling approaches [Chen *et al.*, 2021; Janner *et al.*, 2021] recast control as conditional prediction but lack explicit support constraints. Diffusion-based methods [Janner *et al.*, 2022; Ajay *et al.*, 2022; Janner *et al.*, 2023; Lu *et al.*, 2023] capture rich, multi-modal action distributions yet face fundamental limitations in long-horizon robotic control. Action-level diffusion policies accumulate error over long horizons, while trajectory-level diffusion planners incur substantial inference cost [Janner *et al.*, 2022; Ajay *et al.*, 2022]. Hierarchical and latent diffusion methods [Li *et al.*, 2023; Venkatraman *et al.*, 2024] further couple planning decisions to stochastic denoising, which can destabilize value estimation and compound uncertainty across robotic tasks.

We argue the core difficulty in offline robotic planning is not lacking expressive models, but the absence of a unified principle simultaneously enforcing *support preservation* and *controllability* at temporal abstraction levels. Effective latent planning requires: (i) decoded behaviors remaining near demonstrations (safe extrapolation), and (ii) latent changes producing meaningful action changes (controllable planning). Violating the former causes unsafe extrapolation; violating the latter yields posterior collapse where planning loses control authority. Existing methods typically emphasize one requirement at the expense of the other.

We propose **GRALP**, a **G**enerative **R**epresentation framework for **A**ction refinement and **L**atent **P**lanning, which resolves this tension through a key architectural principle: diffusion is used exclusively as a deterministic *action decoder, never for latent planning*. GRALP constructs a shared latent skill space trained on offline demonstrations, guided by a local principle: decoded action deviation is bounded by latent deviation from dataset support and decoder sensitivity. This principle is instantiated through conservative latent value learning (constraining planned latents to demonstrated regions) and

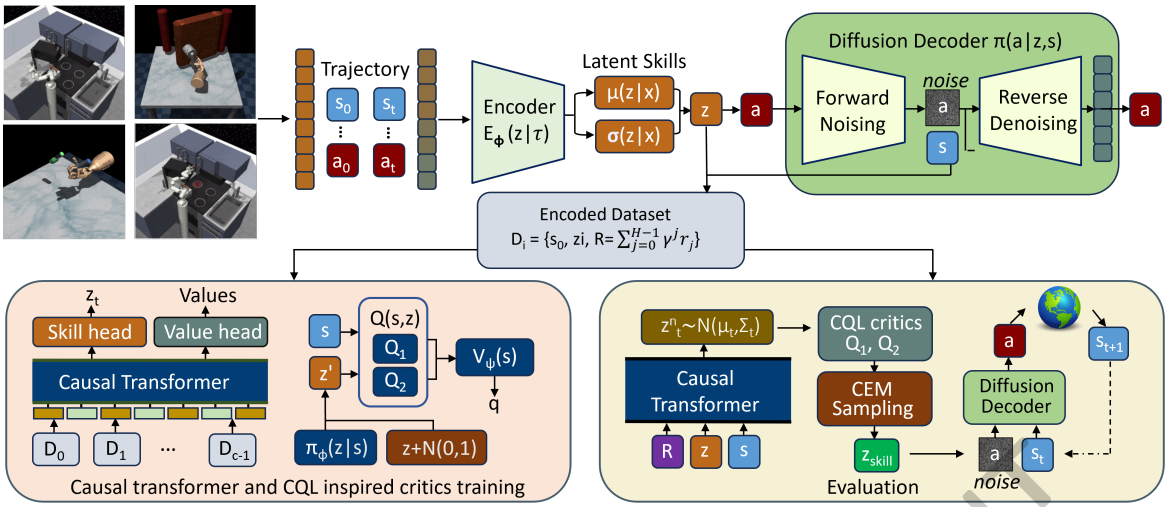

Figure 1: **Illustration of GRALP:** A Conditional Diffusion Autoencoder extracts latent skills from short action-state segments; a conservative critic learns value estimates in latent space; and a causal Transformer predicts the next skill conditioned on returns.

geometric regularization (preserving decoder sensitivity to latent perturbations). This separation enables: (i) efficient discrete planning over latents via a Transformer, (ii) consistent latent-to-action mappings for stable value estimation, and (iii) conservative value estimation directly over planning decisions, achieving controllable and support-preserving temporal abstraction without trajectory-level diffusion overhead.

Our contributions are:

1. We formulate a principled design objective for long-horizon offline robotic planning that balances *support preservation* and *controllability* at the level of temporal abstraction, providing a unified perspective for understanding common failure modes in existing approaches.

2. We introduce GRALP, a unified latent-space planning framework that operationalizes this objective through a deliberate architectural separation: diffusion is used exclusively as an action-space decoder for skill execution, while planning and value estimation operate directly in latent space, enabling efficient long-horizon composition without diffusion-based planning at inference time.

3. With extensive evaluation on D4RL benchmarks and RoboSuite manipulation with human demonstrations, we show that GRALP achieves highest average performance on Navigation, Sequential, and Adroit tasks, surpassing strongest baselines by 8.64%, 3.77%, and 3.38% respectively, while generalizing to contact-rich RoboSuite manipulation and leading across multiple task categories among compared baselines.

2 Preliminaries

2.1 Offline Latent-Space Control Setting

We consider an offline reinforcement learning setting modeled as a Markov decision process $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where the agent is given access only to a fixed dataset $\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1})\}$ collected by an unknown behavior policy. The goal is to learn a policy that maximizes the discounted return $(J(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right])$ without additional environment interaction. In robotic manipulation and navigation, such datasets typically consist of short, locally coherent motion fragments generated by low-level controllers. To expose this structure, trajectories are decomposed into fixed-length segments $\tau_t = (s_t, a_t, \ldots, s_{t+H}, a_{t+H})$, which serve as basic units for constructing temporally extended latent skills in GRALP.

2.2 Diffusion Modeling of Trajectory Segments

Diffusion models represent complex continuous distributions by progressively corrupting data with noise and learning a reverse denoising process. For a trajectory segment representation x_0, the forward diffusion step produces a noisy sample x_k via $x_k = \sqrt{\bar{\alpha}_k}\, x_0 + \sqrt{1-\bar{\alpha}_k}\, \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I)$, where $\bar{\alpha}_k$ controls the signal-to-noise ratio at diffusion step k.

GRALP employs *Denoising Diffusion Implicit Models* (DDIM) [Song *et al.*, 2020] for reconstruction, using a deterministic reverse mapping $(x_{k-1} = g_\theta(x_k, k))$ parameterized by a noise prediction network. Since the diffusion decoder is trained on empirical trajectory segments, reconstructed action sequences remain close to demonstrated behavior.

3 Methodology

We present **GRALP**, a hierarchical offline reinforcement learning framework derived from a single design objective: enabling long-horizon planning while preserving support and controllability in latent space. GRALP decomposes control into two levels: a low-level generative decoder that maps latent skills to short, dataset-consistent motion fragments, and a high-level planner that sequences skills directly in latent space (Fig. 1). Consistent with this design, diffusion is used exclusively as a deterministic action-space decoder, enabling efficient discrete planning while preserving expressive, support-aligned action generation.

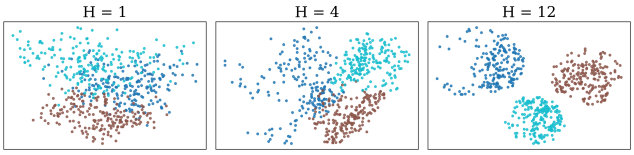

Figure 2: **PCA projections of Kitchen-Mixed latents** for horizons $H \in \{1, 4, 12\}$. Latents are diffuse at H=1, partially structured at H=4, and form well-separated clusters at H=12, indicating stronger temporal abstraction.

3.1 Design Principle: Support Preservation vs. Controllability

Effective latent planning in offline robotics must balance two competing requirements: planned latents should remain close to demonstrated behaviors to avoid unsafe extrapolation, while decoded actions must remain responsive to latent variations to enable meaningful control.

Let $f_s : \mathcal{Z} \to \mathcal{A}^H$ denote the decoder mapping a latent skill to an H-step action sequence at state s, and let $\mathcal{Z}_\mathcal{D}(s)$ denote dataset-supported latents near s (encoded from segments with start states in a local neighborhood).

Assumption 1 (Local Decoder Smoothness). For each state s, the decoder f_s is L_s-Lipschitz continuous in a neighborhood of dataset-supported latents.

Proposition 1 (Support–Controllability Bound). For any planned latent z and state s,

$$\|f_s(z) - f_s(z^*_\mathcal{D})\|_2 \;\le\; L_s \cdot \delta_s(z, s), \tag{1}$$

where $\delta_s(z, s) = \min_{z_\mathcal{D} \in \mathcal{Z}_\mathcal{D}(s)} \|z - z_\mathcal{D}\|_2$ measures deviation from dataset-supported skills, and $z^*_\mathcal{D}$ is nearest such latent.

Proof sketch. By Assumption 1, $\|f_s(z) - f_s(z')\|_2 \le L_s\|z - z'\|_2$ locally. Choosing $z' = z^*_\mathcal{D}$ yields $\|f_s(z) - f_s(z^*_\mathcal{D})\|_2 \le L_s\|z - z^*_\mathcal{D}\|_2 = L_s\,\delta_s(z, s)$. The result follows directly from local Lipschitz continuity of f_s. □

This bound exposes a fundamental trade-off: constraining $\delta_s(z, s)$ preserves dataset support, while maintaining non-degenerate L_s ensures latent variations induce meaningful action changes. GRALP explicitly enforces both through conservative latent planning and geometric regularization.

Architectural Implication. The bound motivates a clear separation of roles: diffusion is confined to deterministic action execution, yielding a well-defined regulated decoder $f_s(z)$, while long-horizon decision making operates entirely in latent space under conservative value learning.

3.2 Diffusion Action Decoder for Latent Skills

Trajectory data are segmented into fixed-length fragments $\tau = (s_t, a_t, \ldots, s_{t+H}, a_{t+H})$, each encoded into a latent skill $z \in \mathcal{Z}$ using a Conditional Diffusion Autoencoder (CDAE). The encoder $q_\phi(z|\tau)$ produces a Gaussian posterior, while a state-conditional prior $p_\omega(z|s)$ regularizes the latent space and provides proposals for planning.

Action sequences are reconstructed using a diffusion decoder trained on empirical segments. Forward noising follows

$$q(a_k \mid a_{k-1}) = \mathcal{N}(\sqrt{1 - \beta_k}a_{k-1}, \beta_k I), \tag{2}$$

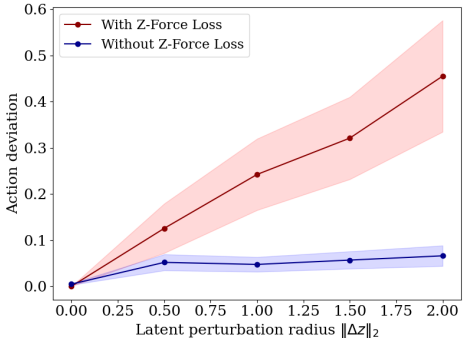

Figure 3: **Latent-action sensitivity** with and without Z-force. With Z-force (red), $\|\Delta a\|_2$ grows roughly linearly with $\|\Delta z\|_2$, reaching ≈ 0.47 at $\|\Delta z\|_2$=2.0. Without Z-force (blue), sensitivity remains low(≈ 0.07), depicting near-collapse as decoder mostly ignores latent

with a learned reverse process

$$p_\theta(a_{k-1} \mid a_k, s, z) = \mathcal{N}(\mu_\theta(a_k, k, s, z), \Sigma_\theta(k)). \tag{3}$$

At inference, deterministic DDIM sampling ensures consistent latent-to-action mappings, essential for reproducible planning and stable value-guided refinement. By confining diffusion to short-horizon action reconstruction within fixed-latent skills, GRALP avoids trajectory-level diffusion at inference, decoupling expressive execution from long-horizon planning.

To prevent the decoder from ignoring latent inputs, latent and state information are injected via Feature-wise Linear Modulation (FiLM) conditioning:

$$h' = \gamma_\theta(k, s, z) \odot h + \beta_\theta(k, s, z). \tag{4}$$

The diffusion model is trained with the standard noise-prediction objective

$$\mathcal{L}_{\text{diff}} = \mathbb{E}\big[\|\epsilon - \epsilon_\theta(a_k, k, s, z)\|_2^2\big], \tag{5}$$

encouraging decoded behaviors to remain close to the empirical data manifold. As the decoder is trained exclusively on trajectory segments from $\mathcal{D}$, latent skills selected by the planner decode into short motion fragments that remain aligned with demonstrated behaviors.

3.3 Geometric Regularization for Latent Controllability

Latent planning is effective only if variations in the latent code induce meaningful changes in decoded actions. In expressive decoders such as diffusion models, this is not guaranteed: the decoder may ignore the latent variable and reconstruct actions directly from the state input, a failure mode known as posterior collapse, which renders high-level planning ineffective.

GRALP enforces latent controllability via two complementary mechanisms. First, stochastic state dropout is applied in the decoder's FiLM conditioning, randomly masking state inputs with probability p_{drop} to force reliance on latent skills. Second, a geometric regularizer that penalizes collapsed latent directions. Let $\pi_{\text{low}}(a \mid s, z)$ denote the action sequence decoded by the diffusion model; we encourage local sensitivity to latent perturbations by penalizing small Jacobian norms:

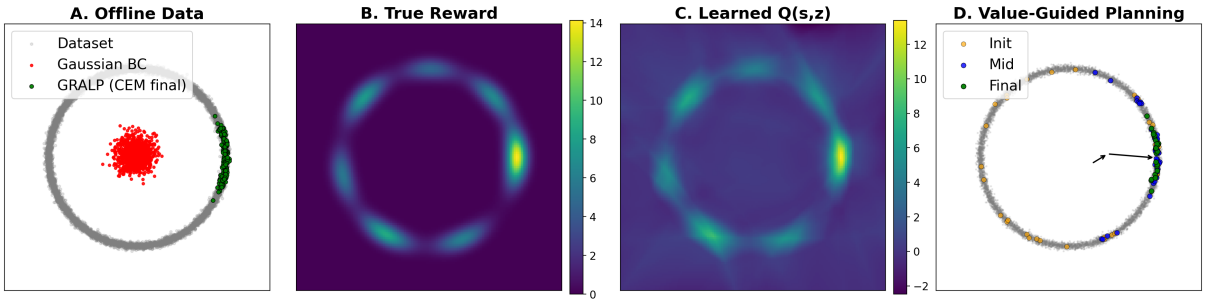

Figure 4: **Conservative value-guided planning on multi-modal 2D ring bandit.** (A) Dataset support (gray) with behavior cloning (red) and GRALP (green). (B) Reward heatmap exhibiting multiple value modes. (C) Conservative critic $Q(s, z)$ assigns high values to in-support regions and suppresses off-support areas. (D) Constrained CEM planning contracts from broad in-support exploration to highest-value region.

Algorithm 1 GRALP Training and Planning Pipeline

Input: Offline dataset $\mathcal{D}$, skill horizon H, latent dim d_z
Output: Trained skill model $(q_\phi, p_\omega, p_\theta)$, latent critics (Q_1, Q_2, V), planner π_ψ

Stage 1: Diffusion skill model training (Sec. 3.2–3.3)
Initialize encoder $q_\phi(z|\tau)$, prior p_ω, decoder $p_\theta(a|s, z)$
for $t = 1$ to T_1 **do**
  Sample trajectory segment $\tau \sim \mathcal{D}$
  Sample latent skill $z \sim q_\phi(z \mid \tau)$
  Sample diffusion timestep and apply state dropout
  Update (ϕ, ω, θ) using loss Eq. 5 and Eq. 7
end for
Freeze q_ϕ **and collect Encoded Dataset** D_z
Stage 2: Conservative latent planning (Sec. 3.4)
Initialize critics Q_1, Q_2, V and planner π_ψ
for $t = 1$ to T_2 **do**
  Sample latent transition batch from dataset D_z
  Update V via expectile regression (Eq. 11)
  Update $Q_{1,2}$ using TD targets (Eq. 8) & CQL (Eq. 10)
  Update planner π_ψ via AWR (Eqs. 11–12)
  Polyak update target networks
end for

$$\mathcal{L}_{\text{force}} = -\log\left(\|\nabla_z \pi_{\text{low}}(a \mid s, z)\|_2 + \varepsilon\right), \tag{6}$$

The log-barrier form provides:
(i) *Collapse prevention*: unbounded penalty as $\|\nabla_z f_s(z)\|_2 \to 0$ (ii) *Stability*: the penalty saturates since $\frac{d}{dg}[-\log(g+\varepsilon)] = -\frac{1}{g+\varepsilon} \to 0$ as $g \to \infty$.

By preserving non-degenerate decoder sensitivity, Z-force directly controls the Lipschitz term L_s in Proposition 1.

$$\mathcal{L}_{\text{reg}} = \beta_{\text{KL}} D_{\text{KL}}(q_\phi(z \mid \tau) \,\|\, p_\omega(z \mid s)) + \lambda \mathcal{L}_{\text{force}}. \tag{7}$$

Together, these terms maintain an expressive and responsive latent space, ensuring high-level planning retains authority over action generation (Fig. 3).

3.4 Conservative Planning in Latent Space

Once a latent skill space is established, planning reduces to selecting a sequence of latent codes whose decoded motion fragments remain feasible and supported by the offline dataset. However, even in latent space, unconstrained value learning can assign overly optimistic values to skills that are rarely or never produced by the encoder, leading to extrapolation error.

GRALP defines a latent Markov decision process where each latent skill z executes a decoded fragment of length H, yielding return R_z and successor state s'. Critic and value functions are trained offline using dataset-aligned transitions $D_z(s, z, R_z, s')$. The diffusion decoder is never used to generate training transitions and is invoked only for action execution at evaluation time, ensuring value estimation and planning remain fully latent-space operations. To ensure temporal consistency with skill execution, we use horizon Bellman targets

$$y = R_{t:t+H-1} + \gamma^H V(s_{t+H}), \tag{8}$$

where $R_{t:t+H-1} = \sum_{h=0}^{H-1} \gamma^h r_{t+h}$ is the cumulative return accrued during execution of a latent skill.

The latent action-value function is defined as

$$Q(s, z) = R_z + \gamma^H V(s'), \tag{9}$$

where $V(s)$ predicts the value of future latent decisions. Two Q-networks and a value network share a common representation for stability. The horizon H trades off temporal abstraction and critic variance; moderate values support reliable long-horizon reasoning while avoiding unstable targets.

To suppress overestimation for unsupported skills, we adapt Conservative Q-Learning to the latent domain. This penalty explicitly restricts latent value overestimation to regions supported by the offline dataset, enforcing small $\delta_s(z, s)$ in Proposition 1 and preventing extrapolative skill selection. The critic minimizes a Bellman error while penalizing high values assigned to latent proposals drawn from a distribution $\mu(z)$ that targets regions near the dataset support. Specifically, $\mu(z)$ includes Gaussian perturbations around encoder latents, local variations of dataset skills, and samples from proposal head trained to match the planner's predictions. The resulting objective enforces a conservative lower bound:

$$\mathcal{L}_{\text{CQL}}(Q_i) = \alpha\Big(\mathbb{E}_{z\sim\mu(z)}[Q_i(s, z)] - \mathbb{E}_{z\sim\mathcal{D}}[Q_i(s, z)]\Big) + \tfrac{1}{2}\mathbb{E}\big[(Q_i(s, z) - \hat{Q}_{\text{target}})^2\big], \tag{10}$$

discouraging extrapolative skills while preserving relative ordering among feasible ones. This conservative value learning

		Non-Diffusion							Diffusion-based									
		Model-Free				Model-Based			Diffusion as Planner				Diffusion as Policy					OURS
	Task Name	BC	CQL	IQL	DT	MOReL	TT	TAP	LDCQ	Diffuser	DD	HDMI	QGPO	SfBC	IDQL	DQL	DQL@1	(Mean ± Std)
Locomotion	Hopper-M	52.9	58.5	66.3	67.6	95.4	61.1	63.4	66.2	58.5	79.3	76.4	**98.0**	57.1	65.4	82.4	64.1	96.26 ± 0.4
	Hopper-M-R	18.1	95.0	94.7	82.7	93.6	91.5	87.3	86.2	96.8	100.0	99.6	96.9	86.2	92.1	**100.7**	63.1	88.41 ± 1.8
	Hopper-M-E	52.5	105.4	91.5	107.6	108.7	110.0	105.5	111.3	107.2	111.8	**113.5**	108.0	108.6	108.6	107.2	100.6	111.90 ± 2.3
	HCheetah-M	42.6	44.0	47.4	42.6	42.1	46.9	45.0	42.8	44.2	49.1	48.0	**54.1**	45.9	51.0	50.6	47.8	51.23 ± 0.3
	HCheetah-M-R	36.6	45.5	44.2	36.6	40.2	41.9	40.8	41.8	42.2	39.3	44.9	**47.6**	37.1	45.9	47.5	44.0	41.92 ± 0.42
	HCheetah-M-E	55.2	91.6	86.7	86.8	53.3	95.0	91.8	90.2	79.8	90.6	92.1	93.5	91.6	95.9	**96.1**	94.8	90.13 ± 0.9
	Walker2d-M	75.3	72.5	78.3	74.0	77.8	79.0	64.9	69.4	79.7	82.5	79.9	**86.0**	53.1	82.5	85.1	82.0	78.42 ± 0.8
	Walker2d-M-R	26.0	77.2	73.9	66.6	49.8	82.6	66.8	68.5	61.2	75.0	80.7	84.4	65.1	85.1	**94.3**	75.6	83.94 ± 1.8
	Walker2d-M-E	107.5	108.8	109.6	108.1	95.6	101.9	107.4	109.3	108.4	108.8	107.9	110.7	109.6	108.6	109.7	108.9	**111.61 ± 0.7**
	Average	51.87	77.19	76.96	74.70	72.94	78.80	74.77	76.18	75.30	81.60	82.60	**86.58**	72.70	81.68	86.30	75.60	83.76
Navigation	Maze2d-U	3.8	5.7	47.4	27.3	0	-	-	134.2	113.9	116.2	120.1	-	73.9	57.9	-	-	**143.14 ± 4.0**
	Maze2d-M	30.0	5.0	34.9	32.1	1.3	-	-	125.3	121.5	122.3	121.8	-	73.8	89.5	-	-	**145.08 ± 4.9**
	Maze2d-L	5.0	12.5	58.6	18.1	0	-	-	**150.1**	123.0	125.9	128.6	-	74.4	90.0	-	-	149.92 ± 2.9
	AntMaze-U-D	45.6	80.3	62.2	53.0	0	-	69.8	81.4	72.2	49.2	73.7	74.4	85.3	80.2	66.2	56.4	**90.75 ± 3.2**
	AntMaze-M-D	0.0	53.7	70.0	0.0	-	0.6	85	68.9	45.5	24.6	-	83.6	82.0	84.8	78.6	14.8	**87.75 ± 4.2**
	AntMaze-L-D	0.0	14.9	47.5	0.0	-	0	-	57.7	22.0	7.5	71.5	64.8	45.5	**67.9**	56.6	1.6	52.35 ± 2.5
	Average	14.06	28.68	53.43	22.78	0.32	-	-	102.94	83.01	74.28	-	-	72.48	78.38	-	-	**111.83**
Sequential	Kitchen-P	38.0	49.8	46.3	0.0	-	-	-	67.8	55.7	56.5	-	66.0	47.9	66.7	60.5	-	**67.96 ± 3.14**
	Kitchen-Mxd	51.5	51.0	51.0	2.5	-	-	-	62.3	52.5	51.8	-	45.5	45.4	66.5	62.6	-	**66.70 ± 2.8**
	Kitchen-Cmp	65.0	43.8	62.5	15.0	-	-	-	62.5	-	-	-	62.8	77.9	62.5	**84.0**	-	81.06 ± 4.3
	Average	51.5	48.2	53.26	5.83	-	-	-	64.4	-	-	-	58.1	57.06	65.23	69.03	-	**71.63**
Adroit	Pen-C	37.0	39.2	37.3	23.5	-	11.4	57.4	47.7	61.7	47.7	48.3	54.2	-	**82.3**	28.3	49.3	81.96 ± 2.5
	Pen-E	85.1	107.0	-	110.0	-	72.0	127.4	121.2	99.7	107.6	109.5	119.1	-	137.8	133.5	112.6	**146.20 ± 3.9**
	Door-C	0.0	0.4	1.6	-0.1	-	-0.1	**11.7**	1.1	0.1	9.0	9.3	11.2	-	4.4	-0.3	10.6	2.23 ± 1.2
	Door-E	34.9	101.5	-	95.5	-	94.1	104.8	105.1	103	94.0	85.9	98.8	-	105.0	104.8	93.7	**108.03 ± 0.6**
	Relocate-C	-0.3	-0.1	-0.1	-0.3	-	-0.1	-0.2	-0.2	0.0	0.15	-0.1	-0.2	-	0.0	0.1	-0.2	**0.27 ± 0.06**
	Relocate-E	101.3	95.0	-	15.0	-	10.3	105.8	104.0	102.2	87.5	91.3	102.5	-	107.0	108.5	95.2	**109.93 ± 2.1**
	Hammer-C	0.8	2.1	1.6	0.2	-	0.5	1.2	2.8	1.2	0.9	1.0	1.1	-	3.5	0.2	1.1	**4.26 ± 1.3**
	Hammer-E	125.6	86.7	-	89.7	-	15.5	127.6	45.8	103.1	106.7	111.8	123.2	-	127.6	128.3	114.8	**132.35 ± 1.16**
	Average	48.05	53.96	55.66	41.69	-	25.45	66.96	53.44	58.86	56.70	57.13	63.74	-	70.95	62.925	59.6375	**73.15**
	Pen-H	34.4	37.5	71.3	6.3	-	36.4	76.5	74.1	-	64.1	66.2	73.9	-	-	72.8	66.0	**80.92 ± 3.5**
	Door-H	0.5	9.9	4.3	3.9	-	0.1	8.8	**11.8**	-	6.9	7.1	8.5	-	-	-	8.0	7.23 ± 2.3
	Relocate-H	0.0	0.2	0.1	0.2	-	0.0	0.2	**0.3**	-	0.2	0.1	0.2	-	-	-	0.2	0.24 ± 0.9
	Hammer-H	1.2	**4.4**	1.4	0.5	-	0.8	1.4	1.5	-	1.0	1.2	1.4	-	-	-	1.3	3.55 ± 1.1
	Average	9.03	13.00	19.28	2.72	-	9.33	21.73	21.93	-	18.05	18.65	21	-	-	-	18.86	**22.98**

Table 1: Unified Evaluation Results on D4RL Benchmark. Comparison against non-diffusion (model-free/model-based) and diffusion-based (planner/policy) baselines. The maximum score in each row is **bolded** and the second highest is underlined. **M/R/E**: Medium/Replay/Expert; **U/L/D**: Umaze/Large/Diverse; **P/Mxd/Cmp**: Partial/Mixed/Complete; **H/C/E**: Human/Cloned/Expert.

constrains the support-deviation term δ_s in our design principle (Fig. 4). High-level planning is performed by a causal Transformer that conditions on recent states, previously executed skills, and an optional return-to-go signal. The planner is trained via Advantage-Weighted-Regression (AWR), i.e.,

$$A(s, z) = \min(Q_1, Q_2) - V(s) \tag{11}$$

emphasizes high-value skills while remaining anchored to conservative value estimates. The corresponding weight

$$w(s, z) = \exp\left(\frac{A(s,z)}{\tau}\right) \tag{12}$$

scales the likelihood objective with KL regularization toward the behavior distribution. At evaluation time, GRALP refines planner proposals using the Cross-Entropy Method (CEM, 3 iterations) directly in latent space. Starting from the Transformer's Gaussian proposal, CEM iteratively samples latent candidates, scores each with the conservative critic $Q(s, z)$, and fits a new Gaussian to the top elite samples. The highest-value latent is then decoded via DDIM. Algorithmic pseudocode for all training stages is provided in Algorithm 1.

4 Experimental Evaluation

We evaluate GRALP on the D4RL benchmark, spanning locomotion, navigation, sequential manipulation, and dexterous robotics control. These domains pose qualitatively different challenges: MuJoCo locomotion emphasizes short-horizon smooth control; Maze2D and AntMaze require composing locally demonstrated motions into goal-directed trajectories under sparse rewards; Kitchen tasks demand multi-stage sequential execution; and Adroit environments introduce high-dimensional, contact-rich manipulation (Fig. 5). Category-wise results in Table 1 assess whether GRALP's strengths

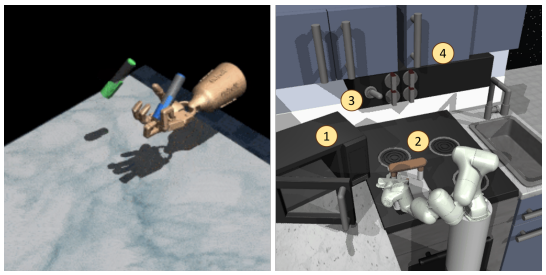

Figure 5: **Representative manipulation tasks from the D4RL benchmark suite.** Left: Adroit-Pen, a dexterous, contact-rich manipulation task learned from human demonstrations. Right: Franka Kitchen, a multi-stage manipulation task requiring sequential completion of interdependent sub-tasks.

emerge where long-horizon reasoning and trajectory composition are central. We also evaluate on RoboSuite pick-and-place tasks to examine generalization beyond D4RL.

4.1 Experimental Protocol

GRALP is trained in two stages: diffusion-based action decoder learning, followed by latent critic and causal Transformer planner optimization. The encoder uses 2-layer GRU (hidden size 256), critics are 3-layer MLPs (width 256), and the planner is an 8-layer Transformer with 8 attention heads. Based on preliminary horizon evaluation, we fix the skill horizon to $H \in \{8, 12\}$ depending on the domain and report results averaged over 10 random seeds with 25 evaluation rollouts per seed. We compare against representative model-free, model-based, diffusion-planner, and diffusion-policy baselines (Table 1). Following QGPO [Lu *et al.*, 2023], we additionally report D-QL@1, which removes the resampling trick to reflect raw diffusion decision quality. Baseline results are taken from the corresponding publications under matching protocols.

4.2 D4RL Benchmark Results

Table 1 reports unified D4RL results. GRALP attains the highest average performance on Navigation, Sequential, and Adroit tasks, while remaining competitive on Locomotion; in contrast, other baselines excel in at most one or two domains.

Navigation
GRALP achieves the highest average performance on navigation benchmarks (Maze2D and AntMaze). These environments require composing locally demonstrated motion fragments into coherent, maze-spanning trajectories absent from the dataset. GRALP outperforms both diffusion-based planners and policies, particularly on umaze and medium variants that exhibit severe distributional shift, consistent with conservative latent value learning that restricts value extrapolation to unsupported regions.

Sequential Manipulation
On the Kitchen suite, which requires correct temporal ordering and subtask composition, GRALP achieves the highest category-wise average performance. Performance gains are largest on Partial and Mixed variants, where sparse feedback demands composing reusable motion fragments, indicating that latent skills capture reusable structure without task-specific heuristics.

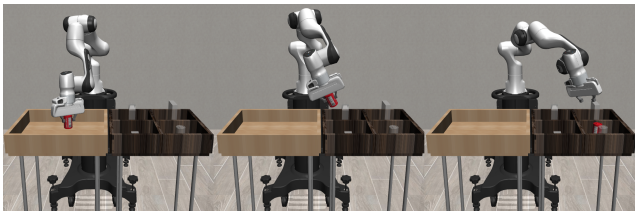

Figure 6: **RoboSuite Pick-and-Place.** GRALP performs multi-stage grasping, transport, and placement tasks.

Adroit Manipulation
GRALP also achieves the strongest overall average on the Adroit benchmark across cloned, expert, and human demonstration settings. These tasks involve high-dimensional, contact-rich dynamics that require fine-grained control while benefiting from temporal abstraction. The results indicate that GRALP preserves local control while enabling effective skill-level planning, avoiding degradation observed in hierarchical approaches to dexterous manipulation.

Locomotion
On MuJoCo locomotion, GRALP remains competitive with leading offline RL methods but does not uniformly outperform approaches optimizing actions directly. This is expected, as these domains emphasize smooth, reactive control over short horizons, where long-horizon planning provides limited advantage. Importantly, GRALP incurs no performance penalty despite its hierarchical structure.

4.3 Generalization to Contact-Rich Manipulation

Beyond D4RL, we evaluate GRALP on RoboSuite manipulation tasks, including **Lift** and **Pick-and-Place**, using offline human demonstrations collected via teleoperation. RoboSuite employs a 7-DoF Panda manipulator with substantially different kinematics and action spaces than Adroit's Shadow Hand and navigation agents. Lift evaluates single-stage grasp acquisition under precise contact constraints, while Pick-and-Place additionally requires sequencing grasp, transport, and placement. Across 10 seeds with 25 rollouts per seed, GRALP achieves a success rate of 96.6% ± 2.1% on Lift and 94.3% ± 3.4% on Pick-and-Place. Together with results on navigation and dexterous manipulation benchmarks, these experiments span diverse embodiments and contact regimes, demonstrating that GRALP's latent skill abstraction generalizes across morphologies while preserving support-aligned execution.

4.4 Ablation Studies

Table 2 reports ablations on four representative tasks spanning sparse-reward navigation, compositional manipulation, and dexterous control. Each removal isolates the role of a component in the support-controllability design principle.

Latent Controllability (L_s)
Removing Z-force substantially degrades performance across all tasks, with largest drops in AntMaze and Kitchen, indicating reduced sensitivity of decoded actions to latent perturbations. State dropout exhibits a complementary effect,

Ablation	AntMaze-M-D	Kitchen-Mxd	Pen-C	Maze2d-L
GRALP (Full)	87.75	66.14	81.96	149.9
W/O Z-force	63.82	40.75	68.38	115.46
W/O State Dropout	65.19	32.46	71.22	133.36
W/O Both	40.48	29.48	51.02	95.85
W/O CQL	37.67	35.65	44.72	73.64
Critic Ignore	61.19	43.28	48.33	135.49
H=1	34.28	30.03	42.71	118.02

Table 2: Component ablations on representative tasks.

with larger degradation in high-dimensional manipulation tasks where reliance on latent structure is critical. Removing both regularizers causes severe collapse, confirming that both mechanisms jointly preserve decoder responsiveness and prevent posterior collapse.

Support Preservation (δ_s)

Removing conservative regularization (W/O CQL) produces the largest drop across tasks, particularly in sparse-reward navigation and long-horizon manipulation. Without conservatism, the latent critic overestimates unsupported skills, causing the planner to select infeasible latent actions, which directly validates the necessity of constraining latent deviation from dataset support.

Critic-Guided Refinement

Without refinement (Critic Ignore), performance degrades most on sparse-reward tasks such as AntMaze, while denser environments are less affected, suggesting latent-space local search is particularly useful when credit assignment is difficult and value uncertainty is high.

Temporal Abstraction

Setting H=1 collapses GRALP to action-level planning and results in consistent performance degradation, demonstrating that fixed-length latent skills are essential for effective long-horizon reasoning. As shown in Fig. 7, performance consistently peaks for intermediate horizons ($H \in [8, 12]$). Short horizons provide insufficient abstraction, while longer horizons introduce excessive critic variance, revealing a clear abstraction-stability trade-off shared across navigation, manipulation, and dexterous control domains.

4.5 Mechanism Validation

We complement quantitative benchmarks and ablations with targeted diagnostics probing the support-controllability principle. Each experiment isolates a specific failure mode predicted by the design inequality in Section 3.

Conservative Critic Behavior

The ring-bandit experiment in Fig. 4 validates conservative value learning in latent space. The critic assigns low values to latents outside dataset support while preserving relative ordering among feasible skills. When combined with constrained CEM-based planning, latent proposals refine to the highest-value region, confirming that conservative regularization constrains the support-deviation term δ_s without collapsing skill diversity.

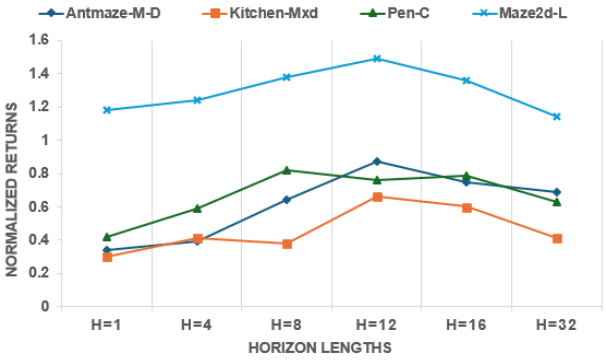

Figure 7: **Performance vs skill horizon H across four tasks.**

Latent Controllability via Z-Force

Fig. 3 illustrates the effect of geometric regularization on decoder sensitivity. Without Z-force, latent perturbations produce minimal changes in decoded actions, indicating collapse. With Z-force, perturbations induce structured action responses, confirming non-degenerate local sensitivity L_s.

Emergence of Structured Latent Skills

PCA visualizations in Fig. 2 show how latent structure evolves with skill horizon. At H=1, representations are diffuse and overlapping, while increasing horizons yield progressively more organized and separable clusters around H=12, aligning with performance trends in Fig. 7.

Long-Horizon Composition

Open-world navigation often requires composing locally demonstrated motion fragments into globally coherent plans that were never observed end-to-end in the dataset. Figure 8 visualizes GRALP stitching fragmented AntMaze data into maze-spanning rollouts. Despite fixed-interval skill transitions, GRALP navigates narrow passages and multi-turn corridors without trajectory-level diffusion planning, confirming that latent skill sequencing enables stable long-horizon behavior while preserving data support.

Together, these diagnostics link GRALP's gains to conservative support preservation, maintained decoder sensitivity, and reliable temporal skill composition.

5 Related Work

Offline reinforcement learning must contend with distributional shift from finite datasets, where optimistic value estimates can induce out-of-support actions. Prior work addresses this via behavior regularization and conservative value learning (e.g., BCQ [Fujimoto *et al.*, 2019], TD3+BC [Fujimoto and Gu, 2021], CQL [Kumar *et al.*, 2020], IQL [Kostrikov *et al.*, 2021]), or by recasting control as conditional sequence modeling (e.g., Decision Transformer [Chen *et al.*, 2021]), which avoids explicit bootstrapping but offers limited guarantees for compositional long-horizon planning. Hierarchical and latent-skill approaches reduce temporal complexity by planning over abstractions such as subgoals or latent skills [Mandlekar *et al.*, 2020; Rosete *et al.*, 2023; Kim *et al.*, 2024; Li *et al.*, 2022; Zhou *et al.*, 2020], while model-based methods

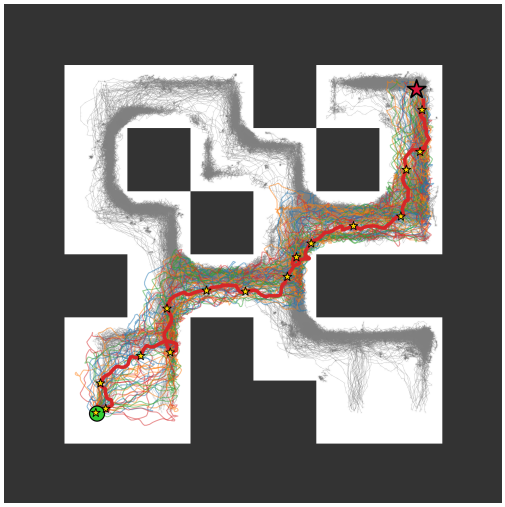

Figure 8: **Trajectory stitching on AntMaze-Medium-Diverse.** Gray traces show fragmented dataset coverage. Colored trajectories are GRALP rollouts, with a bold red path indicating a successful start-to-goal trajectory (start: green circle, goal: red star). Yellow stars mark skill transitions every H=12 steps. GRALP composes maze-spanning paths not present as single dataset trajectories.

enable longer-range reasoning through learned dynamics or rollout-based planning (e.g., MOReL [Kidambi *et al.*, 2020], Trajectory Transformer [Janner *et al.*, 2021], TAP [Jiang *et al.*, 2023]). Diffusion models have been applied either as trajectory planners generating full action sequences at inference [Janner *et al.*, 2022; Ajay *et al.*, 2022; Li *et al.*, 2023; Venkatraman *et al.*, 2024] or as expressive stochastic policies [Wang *et al.*, 2023; Chen *et al.*, 2023; Lu *et al.*, 2023; Hansen *et al.*, 2023], while goal-conditioned reinforcement learning frames long-horizon control via conditioning on desired outcomes but suffers from sparse rewards and generalization challenges [Liu *et al.*, 2022]. In contrast, GRALP confines diffusion to deterministic action execution within latent skills, while long-horizon decision making operates in latent space via conservative value learning and skill-level planning, avoiding trajectory-level diffusion at inference.

6 Conclusion

We presented GRALP, a principled framework for offline robotic planning that preserves dataset support while enabling controllable long-horizon decision making through a clear separation of roles: diffusion is used solely for deterministic action execution within latent skills, while planning and value estimation operate directly in latent space under conservative constraints. Across long-horizon benchmarks, GRALP achieves the highest average performance on Navigation, Sequential Manipulation, and Adroit tasks while remaining competitive on locomotion. RoboSuite experiments further demonstrate generalization to contact-rich manipulation from human demonstrations without trajectory-level diffusion planning. While our evaluation focuses on high-fidelity simulation, GRALP is designed with physical deployment constraints in mind: conservative latent planning mitigates unsafe extrapolation, morphology-agnostic skill abstraction promotes cross-platform generalization, and deterministic execution improves reproducibility. Extending to physical platforms with sim-to-real techniques is an important next step.

Reproducibility: Hyperparameters, protocols and implementation are available at *github.com/talhazaidi13/GRALP*.

Acknowledgments

This work was supported in part by the United States Department of Agriculture (USDA) National Institute of Food and Agriculture (NIFA) under Award Number 2023-67021-44838. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the USDA NIFA.

References

[Ajay *et al.*, 2022] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? In *The Eleventh International Conference on Learning Representations (ICLR)*, 2022.

[Chen *et al.*, 2021] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems (NeurIPS)*, 34:15084–15097, 2021.

[Chen *et al.*, 2023] Huayu Chen, Cheng Lu, Chengyang Ying, Hang Su, and Jun Zhu. Offline reinforcement learning via high-fidelity generative behavior modeling. *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

[Fujimoto and Gu, 2021] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems (NeurIPS)*, 34:20132–20145, 2021.

[Fujimoto *et al.*, 2019] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning (ICML)*, pages 2052–2062. PMLR, 2019.

[Hansen *et al.*, 2023] Philippe Hansen, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *International Conference on Learning Representations (ICLR)*, 2023.

[Janner *et al.*, 2021] Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems (NeurIPS)*, 34:1273–1286, 2021.

[Janner *et al.*, 2022] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning (ICML)*, 2022.

[Janner *et al.*, 2023] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. Diffusion policies as an expressive policy class for offline reinforcement learning.

In *International Conference on Learning Representations (ICLR)*, 2023.

[Jiang *et al.*, 2023] Zhengyao Jiang, Tianjun Zhang, Michael Janner, Yueying Li, Tim Rocktäschel, Edward Grefenstette, and Yuandong Tian. Efficient planning in a compact latent action space. *International Conference on Learning Representations (ICLR)*, 2023.

[Kidambi *et al.*, 2020] Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems (NeurIPS)*, 33:21810–21823, 2020.

[Kim *et al.*, 2024] Donghoon Kim, Minjong Yoo, and Honguk Woo. Offline policy learning via skill-step abstraction for long-horizon goal-conditioned tasks. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, (IJCAI-24)*, pages 4282–4290, 8 2024. Main Track.

[Kostrikov *et al.*, 2021] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *ArXiv*, abs/2110.06169, 2021.

[Kumar *et al.*, 2020] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems (NeurIPS)*, 33:1179–1191, 2020.

[Li *et al.*, 2022] Jinning Li, Chen Tang, Masayoshi Tomizuka, and Wei Zhan. Hierarchical planning through goal-conditioned offline reinforcement learning. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):10216–10223, 2022.

[Li *et al.*, 2023] Wenhao Li, Xiangfeng Wang, Bo Jin, and Hongyuan Zha. Hierarchical diffusion for offline decision making. In *International Conference on Machine Learning (ICML)*, pages 20035–20064. PMLR, 2023.

[Liu *et al.*, 2022] Minghuan Liu, Menghui Zhu, and Weinan Zhang. Goal-conditioned reinforcement learning: Problems and solutions. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, (IJCAI-22)*, pages 5502–5511, 7 2022. Survey Track.

[Lu *et al.*, 2023] Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 22825–22855. PMLR, 2023.

[Mandlekar *et al.*, 2020] Ajay Mandlekar, Fabio Ramos, Byron Boots, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Dieter Fox. Iris: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4414–4420. IEEE, 2020.

[Rosete *et al.*, 2023] Erick Rosete, Oier Mees, Gabriel Kalweit, Joschka Boedecker, and Wolfram Burgard. Latent plans for task-agnostic offline reinforcement learning. In *Conference on Robot Learning (CoRL)*, pages 1838–1849. PMLR, 2023.

[Song *et al.*, 2020] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[Venkatraman *et al.*, 2024] Siddarth Venkatraman, Shivesh Khaitan, Ravi Tej Akella, John Dolan, Jeff Schneider, and Glen Berseth. Reasoning with latent diffusion in offline reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2024.

[Wang *et al.*, 2023] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2023.

[Zaidi *et al.*, 2026] Syed Ahsan Masud Zaidi, Lior Shamir, William Hsu, Scott Dietrich, and Talha Zaidi. Graze: Grounded refinement and motion-aware zero-shot event localization. *arXiv preprint arXiv:2604.01383*, 2026.

[Zhou *et al.*, 2020] Guojing Zhou, Hamoon Azizsoltani, Markel Sanz Ausin, Tiffany Barnes, and Min Chi. Hierarchical reinforcement learning for pedagogical policy induction (extended abstract). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, (IJCAI-20)*, pages 4691–4695, 7 2020. Sister Conferences Best Papers.