Software Engineering and Applications 软件工程与应用, 2025, 14(3), 535-549
Published Online June 2025 in Hans. https://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2025.143047

Hans汉斯

基于关键状态的扩散模型轨迹规划方法

杜潇含，李　烨

上海理工大学光电信息与计算机工程学院，上海

收稿日期：2025年3月29日；录用日期：2025年5月30日；发布日期：2025年6月11日

摘　要

在离线强化学习的轨迹规划任务中，传统基于自回归的规划方法因误差逐级累积效应而限制了模型性能。近年来，扩散模型凭借其出色的分布建模能力被引入该领域，以缓解误差累积问题。然而，现有方法在高维动作空间生成长时序轨迹时仍面临性能不足的挑战。为此，本文提出了一种基于关键状态的扩散模型轨迹规划方法，通过提取原始轨迹中的关键状态特征，并结合条件扩散生成模型进行轨迹规划，将传统的自回归式轨迹规划范式转化为基于关键状态的条件生成问题。在确保生成轨迹时序连续性的同时，提升了模型轨迹规划的性能。在D4RL基准测试的Gym-Mujoco、Maze2d、AntMaze和Adroit等多个环境中进行的实验表明，本文方法在轨迹规划性能和算法鲁棒性方面均优于现有方法。

关键词

离线强化学习，扩散模型，轨迹规划，Transformer，变分自编码器

Key-State-Conditioned Diffusion Models for Trajectory Planning

Xiaohan Du, Ye Li

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai

Received: Mar. 29th, 2025; accepted: May 30th, 2025; published: Jun. 11th, 2025

Abstract

In trajectory planning for offline reinforcement learning, conventional autoregressive planning methods suffer from performance limitations due to error accumulation effects. While diffusion models have recently been introduced to this domain to mitigate error accumulation through their exceptional distribution modeling capabilities, existing approaches still face performance

文章引用: 杜潇含, 李烨. 基于关键状态的扩散模型轨迹规划方法[J]. 软件工程与应用, 2025, 14(3): 535-549.
DOI: 10.12677/sea.2025.143047

challenges when generating long-horizon trajectories in high-dimensional action spaces. To address this, we propose a Key-State-Conditioned Diffusion Models for Trajectory Planning method that integrates key states with diffusion models. Our approach extracts critical state features from original trajectories and combines them with conditional diffusion generative models for trajectory planning, effectively transforming the traditional autoregressive planning paradigm into a key state-conditioned generation problem. This method not only maintains temporal continuity in generated trajectories but also significantly enhances planning performance. Extensive experiments conducted on multiple D4RL benchmark environments, including Gym-Mujoco, Maze2d, AntMaze, and Adroit, demonstrate that our method outperforms existing approaches in both trajectory planning performance and algorithmic robustness.

Keywords

Offline Reinforcement Learning, Diffusion Model, Trajectory Planning, Transformer, Variational Autoencoder

1. 引言

强化学习(Reinforcement Learning, RL)在机器人控制[1] [2]、自动驾驶[3]等序列决策任务中展现出巨大潜力。然而，传统在线强化学习面临安全风险高、试错成本大等固有缺陷。在此背景下，离线强化学习(Offline Reinforcement Learning)通过利用静态数据集进行策略优化，有效规避了实时交互需求，逐渐发展成为该领域的主流范式[4]。学界已提出多种创新方法：Fujimoto 等人提出批次约束 Q 学习(Batch-Constrained Q-Learning, BCQ) [5]算法，通过行为克隆与 Q 学习的协同机制显式约束策略空间；Kumar 等人提出保守 Q 学习算法(Conservative Q learning, CQL) [6]，创新性地在价值函数中引入正则化项以逼近真实 Q 值的下界；Kidambi 等人提出的基于模型的离线强化学习算法(Model-based Offline Reinforcement Learning, MOReL) [7]则通过构建环境模型来补偿离线交互的缺失。

然而，传统离线强化学习方法在复杂轨迹规划任务中仍面临诸多挑战：自回归式规划方法受限于单步预测误差在时序决策中的累积效应，其复合误差会导致生成轨迹逐渐偏离真实环境动态[8]。此外，基于单步 Q 值估计的方法难以有效应对稀疏奖励场景下的信用分配问题(Credit Assignment Problem, CAP)，同时受限于离线数据集的次优轨迹分布，传统方法易受分布偏移的影响[9]。

生成式模型的技术突破为离线强化学习开辟了新路径。研究者通过将序列建模与传统规划方法相结合，突破了传统策略迭代的框架，将序贯决策转化为直接的序列建模问题，展现出独特的优势。轨迹 Transformer 算法(Trajectory Transformer, TT) [10]率先引入 Transformer 架构[11]，利用其自注意力机制有效捕捉状态–动作序列的长期依赖关系。决策 Transformer (Decision Transformer, DT) [12]进一步将强化学习重构为序列预测任务，通过嵌入网络将历史轨迹编码为连续表征。

近年来，扩散模型(Diffusion Model)在图像生成领域获得显著成就[13]，也为轨迹规划提供了新工具[14]。Janner 等人提出了基于扩散模型结构的条件生成建模算法 Diffuser [15]，通过反向扩散过程生成满足约束条件的轨迹序列，开创了基于扩散模型的规划范式。相较于传统方法，这类方法通过联合建模完整轨迹的概率分布，有效缓解了自回归方法的累积偏差和分布偏移问题。扩散模型在建模高维连续动作

空间时展现出的多模态捕捉能力，使其成为复杂轨迹生成的有效工具。

然而，现有基于扩散模型的轨迹规划方法仍存在关键瓶颈：

1) 时序特征学习的不足：现有方法缺乏对原始轨迹时序特征的有效建模，导致难以平衡轨迹规划长度与质量。短视距规划易受稀疏奖励和局部不确定性的影响，造成轨迹规划出现偏差；而长视距规划则要求模型具备更强的跨步依赖建模能力。

2) 轨迹生成的局限性：部分方法因缺乏价值引导，难以生成高价值轨迹，且环境约束不充分会导致生成的轨迹可能违反物理限制。

3) 生成式模型噪声问题：无分类器引导机制可能导致模型倾向于生成低概率区域轨迹，这些偏离主分布的样本会显著影响策略的可靠性。

针对上述挑战，本研究提出了一种基于关键状态的扩散模型轨迹规划方法(Key-State-Conditioned Diffusion Models for Trajectory Planning, KSDP)，本文的主要贡献如下：

- 构建了一个融合 Transformer 时序建模与 β-VAE 特征压缩的混合架构，从原始轨迹中提取包含状态转移规律和语义上下文的关键特征。
- 构建了基于关键状态条件的扩散模型，通过价值函数引导生成高回报状态序列，并耦合逆动力学模型确保动作轨迹的物理可行性。
- 提出基于核密度估计的自适应筛选策略，通过动态阈值调整有效抑制生成模型的分布外噪声。

2. 基于关键状态的扩散模型轨迹生成

2.1. 关键状态提取

在本研究方法中，关键状态被定义为能够有效表征原始轨迹中跨 T 个时间步长状态序列的显著性特征。这些关键状态不仅需要捕捉轨迹的核心动态信息，还需编码其上下文时序关联特征，从而为后续扩散模型提供结构化条件输入，以生成高保真度的状态轨迹。在连续控制任务中，轨迹数据的复杂性和时序依赖性对特征提取提出了较高要求。关键状态的提取旨在减少冗余信息的影响，突出对轨迹规划具有决定性意义的状态特征。近年来，Transformer 架构因其在自然语言处理领域的显著成功而备受关注，其核心在于通过自注意力机制(Self-Attention)有效建模序列元素的长距离依赖关系，这一特性使其在处理长时序数据时表现出色，尤其适用于捕捉轨迹数据中的长期动态模式。因此，我们首先使用 Transformer 模型初步提取关键状态的上下文时序特征。

数据集中的轨迹样本的形式为 $\tau=\left(s_1,a_1,r_1,s_2,a_2,r_2,\cdots,s_T\right)$，其中 $s\in\mathbb{R}^{d_s}$ 表示状态，$a\in\mathbb{R}^{d_s}$ 表示动作，$r\in\mathbb{R}^{d_s}$ 表示奖励。首先，我们需要定义模型的输入形式及其编码目标。轨迹 τ 是一个长度为 T 的序列，轨迹的每个时间步由状态、动作和奖励组成，每个时间步的输入定义为 $x_t=\left[s_t,a_t,r_t\right]$。然而动作空间、状态空间和奖励空间的维度大小往往存在差异。为了统一模型的输入维度，我们通过独立的全连接层将各模态映射到统一的隐空间 d_h，再融合为统一的时间步输入：

$$e_s^{(t)}=W_s\cdot s_t+b_s,e_a^{(t)}=W_a\cdot a_t+b_a,e_s^{(t)}=W_r\cdot r_t+b_r \tag{1}$$

其中 $W_s\in\mathbb{R}^{d_h\times d_s}$，$W_a\in\mathbb{R}^{d_h\times d_a}$，$W_r\in\mathbb{R}^{d_h\times d_r}$ 为嵌入权重矩阵，b 为偏置向量。通过上述嵌入层，状态、动作和奖励都被映射到相同的维度 d_h，输入的嵌入层如图 1 所示。原始数据轨迹经过嵌入后的序列可以表示为 $E\left[e_1,e_2,\cdots,e_T\right]\in\mathbb{R}^{T\times d_h}$，$e_t$ 表征了时间步 t 的完整信息。

Transformer 模型摒弃了传统循环神经网络(RNN)的递归结构，完全依赖自注意力机制处理序列数据。然而，输入元素的顺序变化不会影响注意力权重的计算结果，导致模型无法区分序列中元素的时序关系，所以必须显式引入位置信息以捕捉轨迹的马尔可夫性及长期依赖关系。

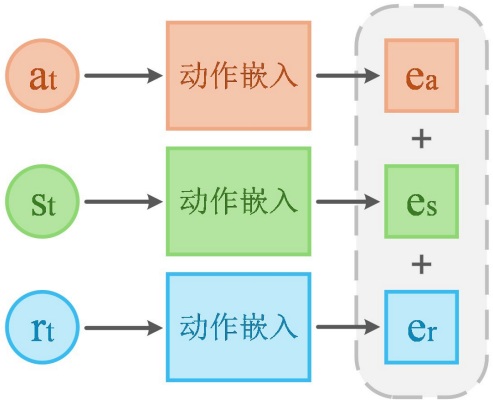

Figure 1. Embedding layer schematic diagram
图 1. 嵌入层示意图

传统的绝对位置编码(Absolute Positional Encoding)通过为每个时间步添加固定的正弦–余弦函数或可学习向量来表示位置。然而，这种方法在长序列中可能难以充分捕捉动态的时序关系，尤其当状态、动作和奖励之间的依赖随相对位置变化时。为此，我们采用 Shaw 等人提出的相对位置编码(Relative Positional Encoding, RPE) [16]。与传统的绝对位置编码不同，该方法通过建模序列中任意两位置 i 和 j 的相对距离 $j-i$，增强模型对长序列结构的感知能力。为控制计算复杂度，设定最大相对距离 $k_{\max}$，超出范围的距离被截断：

$$\text{clipped\_distance} = \text{clip}\left(j-i,-k_{\max},k_{\max}\right) \tag{2}$$

其中，$k_{\max}$ 表示考虑的最大相对距离，超出此范围的距离被截断，相对距离被限制在 $\left[-k_{\max},k_{\max}\right]$ 范围内，减少了模型参数量，并假设远距离依赖的影响较小。在标准的自注意力机制中，注意力权重由查询(Query)和键(Key)的点积计算：

$$\text{Attention}\left(Q,K,V\right) = \text{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_h}}\right)V \tag{3}$$

其中 $Q=XW_Q$，$K=XW_K$，$V=XW_V$ 分别是查询、键和值矩阵，W_Q W_K W_V 是可学习的权重。这种计算仅依赖于 Q 和 K 的内容信息，忽略了时间步之间的相对位置关系。相对位置编码通过在注意力分数中引入位置偏置项，显式建模 i 和 j 之间的相对距离。具体而言，在计算 QK^{T} 时引入一个相对位置表示矩阵 RPE，从而在注意力分数中显式建模位置关系：

$$\text{Attention}\left(Q,K,V\right) = \text{softmax}\left(\frac{QK^{\mathrm{T}}+RPE}{\sqrt{d_h}}\right)V \tag{4}$$

其中，$RPE\in\mathbb{R}^{T\times T}$ 是一个相对位置偏置矩阵，矩阵中的元素 $RPE_{i,j}$ 表示位置 i 和 j 之间的相对位置信息，通过神经网络学习生成。此机制赋予模型动态选择关键时间步的能力，而非简单地对所有时间步进行均等处理。嵌入序列通过多头自注意力层处理，生成全局特征表示 M。多头机制通过不同子空间并行计算注意力，提升特征表达能力：

$$\text{head}_m = \text{Attention}\left(QW_m^Q,KW_m^K,VW_m^V\right) \tag{5}$$

$$M = \text{Concat}\left(\text{head}_1,\text{head}_2,\cdots,\text{head}_m\right)W_O \tag{6}$$

其中，W_i^Q，W_i^K，W_i^V 为查询、键和值的投影矩阵，m 为注意力头数，$W_O\in\mathbb{R}^{d_h\times d_h}$ 是输出的线性变换矩

阵。在多头自注意力机制完成序列的全局特征交互后，输出 M 通过残差连接(Residual Connection)与层归一化(Layer Normalization)接入由全连接层组成前馈神经网络，通过 ReLU 激活函数引入非线性表达能力。自注意力层和前馈神经网络层组成了一个基本的上下文特征提取模块，通过重复堆叠构成了完整的编码器结构，如图 2 所示。

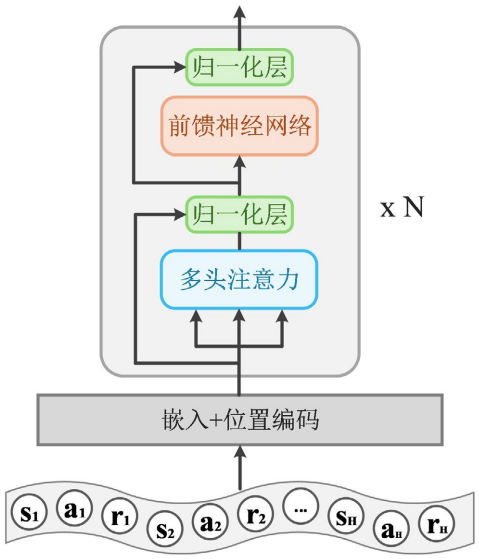

Figure 2. Transformer encoder architecture diagram
图 2. Transformer 编码器结构图

Transformer 编码器输出的 $M \in \mathbb{R}^{T\times d_h}$ 蕴含丰富的上下文信息，但由于其保留了输入序列的维度，仍可能包含冗余或不关键的信息。为进一步提炼关键状态特征，我们将 M 输入至 β-变分自编码器(β-Variational Autoencoder, β-VAE) [17]模型中，以生成隐变量 $z \sim q(z \mid M)$。其目的是压缩轨迹特征，同时保留足够的信息用于后续生成任务。解码器部分被建模为逆动力模型，解码器的输入为相邻状态和关键状态特征 z，输出为模型预测产生这个状态转移的动作 a，即 $q(a_t \mid s_t, s_{t+1}, z)$。逆动力模型的设计不仅作为解码器辅助编码器训练，还作为轨迹规划中的动作轨迹生成器，这部分将在下一小节详细讲解。β-VAE 模型的训练目标结合了重构损失和 KL 散度，确保重构精度和隐变量分布的正则化：

$$L_\beta = E_{a_t \sim D}\left[\sum_{t=1}^{T}\left\|a_t - D_\psi\left(a_t \mid s_t, s_{t+1}, z\right)\right\|_2^2 - \beta \cdot D_{KL}\left(E_\phi\left(z \mid M\right) \| p(z)\right)\right] \tag{7}$$

其中第一项为解码器预测动作和原始轨迹中动作的重构损失，表示逆动力模型在给定潜在变量 z 和状态序列 s 的情况下重构动作序列 a 的能力。第二项为 KL (Kullback-Leibler)散度项，约束了关键状态特征分布 $E_\phi(z \mid M)$ 与先验分布 $p(z)$ 之间的接近程度。$\beta \in [0,1]$ 为超参数，用于平滑重构损失和 KL 散度之间的权重，当 $\beta = 1$ 时，β-VAE 退化为传统的 VAE。通过调整 β 的值，β-VAE 模型能够在重构质量和拟合潜在变量分布之间取得更好的平衡。通过 Transformer 和 β-VAE 编码器，我们从原始轨迹 τ 中提取到了关键状态的紧凑表示 z，其捕获了轨迹的高级语义信息，如策略模式与环境动态等，且有效滤除了噪声，为生成任务提供紧凑且信息丰富的关键状态表示。

2.2. 扩散轨迹生成

现有方法大多直接利用扩散模型生成状态－动作轨迹，这种方式虽然直观，但在强化学习场景中存

在局限性。在强化学习的任务环境中，状态通常具有连续性，而动作则呈现出更大的多样性，且本质上往往是离散的。在机器人控制场景中，以关节力矩表示的动作序列通常表现出较高的频率和平滑性，这显著增加了预测和建模的难度[18]。为应对这一问题，我们首先使用扩散模型生成状态序列，再使用上节提到的逆动力模型(Inverse-Dynamics) [19]，根据状态序列推断出动作序列，如图 3 所示。

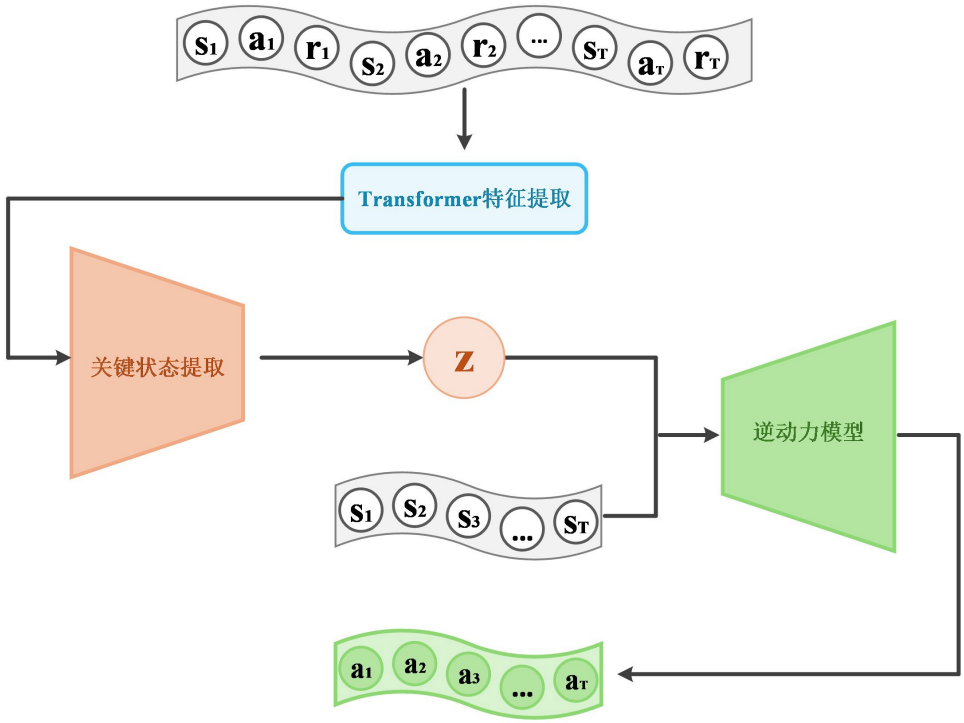

Figure 3. Training flowchart of KSDP algorithm
图 3. KSDP 算法的训练流程图

具体而言，扩散模型 G_θ 用于生成 T 个时间步长的状态轨迹序列，其中 k 表示去噪过程中的时间步长：

$$x^k(\tau)=\left(s_t,s_{t+1},\cdots,s_{t+T-1}\right)^k \tag{8}$$

在生成状态轨迹后使用逆动力学模型预测动作：

$$a_t=D_\psi\left(a_t\mid s_t,s_{t+1},z\right) \tag{9}$$

我们将强化学习中的轨迹规划问题定义为条件扩散模型的序列生成问题，其中上标表示扩散过程的时间步，下标表示强化学习中的时间步：

$$\max_\theta E_{\tau\sim D}\left[\log p_\theta\left(x^0(\tau)\mid y(\tau)\right)\right] \tag{10}$$

我们使用关键状态 z 作为轨迹生成模型的条件，通过条件信息 $y(\tau)$ 生成轨迹，将扩散过程限制在轨迹状态的上下文信息范围内。扩散模型的前向过程是一个马尔可夫过程，通过逐步向原始轨迹数据 x^0 添加高斯噪声，使其分布逐渐趋向各向同性的高斯分布。该过程定义为马尔可夫链：

$$q\left(x^{1:K}\mid x^0\right)=\prod_{k=1}^{K}q\left(x^k\mid x^{k-1}\right) \tag{11}$$

$$q\left(x^k\mid x^{k-1}\right)=N\left(x^k;\sqrt{1-\beta^k}x^{k-1},\beta^kI\right) \tag{12}$$

其中，β^k 是时间步 k 的噪声调度参数，控制每一步添加的噪声强度。通过重参数化技巧，可直接从原始状态轨迹 τ^0 采样任意时刻 K 的噪声轨迹，前向过程的闭式解为：

$$q\left(x^K \mid x^0\right)=N\left(x^k;\sqrt{\bar{\alpha}^k}x^0,\left(1-\bar{\alpha}^k\right)I\right) \tag{13}$$

其中 $\alpha^k=1-\beta^k$，方差调度采用余弦退火策略保证训练稳定性。原始状态轨迹数据在 $k\to K$ 时逐渐趋向于标准高斯分布 $x^K\sim N\left(0,I\right)$。反向过程的目标是从纯噪声 $x^K\sim N\left(0,I\right)$ 开始，逐步去噪以恢复原始轨迹 x^0。该过程同样被建模为马尔可夫过程，其条件分布为：

$$p_\theta\left(x^{0:K}\mid z\right)=p\left(x^K\right)\prod_{k=1}^{K}p_\theta\left(x^{k-1}\mid x^k,z\right) \tag{14}$$

$$p_\theta\left(x^{k-1}\mid x^k,z\right)=N\left(x^{k-1};\mu_\theta\left(x^k,k,z\right),\Sigma^k I\right) \tag{15}$$

其中，μ_θ 是参数化的均值函数，由神经网络预测。Σ^k 通常设为固定的协方差矩阵 β^k 以简化计算。反向去噪的均值可以通过噪声预测网络建模：

$$\mu_\theta\left(x^k,k,z\right)=\frac{1}{\sqrt{\alpha^k}}\left(x^k-\frac{\beta^k}{\sqrt{1-\bar{\alpha}^k}}\varepsilon_\theta\left(x^k,k,z\right)\right) \tag{16}$$

为平衡生成轨迹的多样性与条件约束，我们采用无分类器扩散(Classifier-free Diffusion) [20]框架。在每轮训练中，我们从均匀分布 $k\sim\mathcal{U}\left(1,K\right)$ 中随机采样时间步 k，根据前向过程 $q\left(x^K\mid x^0\right)$ 将采样的原始状态轨迹 x^0 转换为加噪样本 x^K。随后，神经网络 ε_θ 以 x^K、k 和 z 为输入，预测前向过程中添加的噪声 ε。同时引入条件丢弃机制：在每次前向传播中，以概率 β 将条件 z 替换为无效条件 $\varnothing$，从而使模型同时学习条件和无条件分布。模型的损失函数基于 DDPM [21]简化后的扩散模型目标，即预测噪声的均方误差。给定真实噪声 $\varepsilon\sim N\left(0,I\right)$，损失函数定义为：

$$L\left(\theta\right)=E_{k,\tau\in D}\left[\left\|\varepsilon-\varepsilon_\theta\left(x^k\left(\tau\right),\left(1-\beta\right)z+\beta\varnothing,k\right)\right\|^2\right] \tag{17}$$

通过最小化该损失，模型能够在生成符合条件 z 的轨迹的同时，在无条件情况下保持生成结果的多样性。通过调整指导强度 β，可以灵活控制条件对生成结果的影响，允许在多样性(无条件生成)和条件一致性(有条件生成)之间进行权衡。然而，这种方式无法显式优化任务目标，无法直接引导模型生成更高回报的轨迹。

为解决这一问题，我们额外训练了一个轨迹价值预测器 V_ς，通过关键状态特征 z 来预测扩散生成轨迹的累计回报，以此作为轨迹生成中的价值梯度引导。特别地，我们在累计回报预测中隐式引入折扣因子，确保其与价值函数的定义一致，从而提升引导精度。在训练时使用离线数据中长度为 T 的轨迹的累计回报作为监督信号：

$$V\left(\tau\right)=\sum_{t=1}^{T}\gamma^{t-1}R\left(s_t,a_t\right) \tag{18}$$

轨迹价值预测器 V_ς 训练目标为最小化预测值与真实折扣回报之间的重构损失：

$$L_\varsigma=E_{\tau\sim D}\left[\left\|V_\varsigma\left(z\right)-V\left(\tau\right)\right\|_2^2\right] \tag{19}$$

其中 D 表示离线数据集，z 是编码器提取的关键状态特征。最终采样过程，结合无分类器方法和轨迹价值梯度引导方式实现：

$$\hat{\varepsilon}=\omega\varepsilon_\theta\left(x^k,z,k\right)+\left(1-\omega\right)\varepsilon_\theta\left(x^k,\varnothing,k\right)-\sqrt{1-\bar{\alpha}^k}\nabla V_\varsigma\left(z\right) \tag{20}$$

其中 ω 表示无分类器方法在条件扩散模型中的引导权重。设置 $\omega=1$ 时，模型等效于标准条件生成模型；设置 $\omega>1$ 时，增强条件信息的影响，生成结果更贴近条件 z。$\bar{\alpha}^k$ 表示累积的噪声方差，随时间步增加而减小，则 $\sqrt{1-\bar{\alpha}^k}$ 随时间步增大而增大。价值梯度项的引导强度自动适应扩散过程中的时间步 k，无需手动设置超参数，从而在扩散后期强化价值引导效果。在扩散后期，带噪轨迹 τ^k 已接近真实分布，此时价值梯度引导更可靠，能够更精确地优化任务目标，避免早期过强引导引入的偏差。通过在扩散生成过程中融合无分类器引导和轨迹价值梯度引导，模型既能生成符合关键状态约束的轨迹，又能产出高价值的样本。

2.3. 轨迹规划

在模型训练结束后的应用阶段，KSDP 算法的轨迹规划流程如图 4 所示。

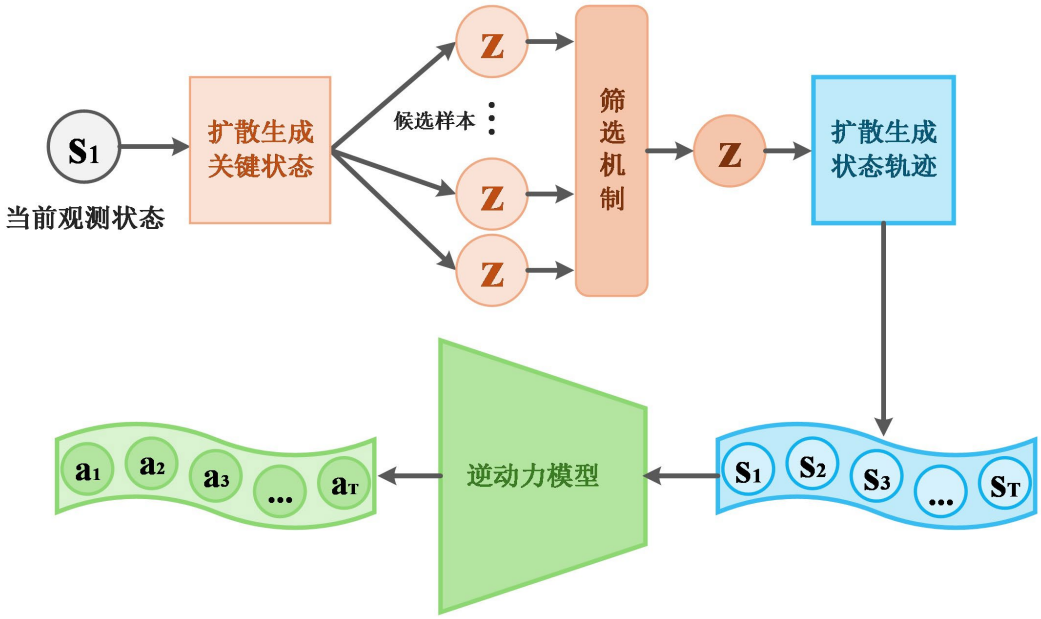

Figure 4. Flowchart of trajectory planning in KSDP algorithm
图 4. KSDP 算法的轨迹规划流程图

我们从初始状态 s_t 开始进行轨迹规划，为了获取关键状态特征，我们需要对先验分布 $p(z|s_t)$ 建模，即需要训练一个模型将当前状态映射到关键状态特征。关键状态特征 z 是一个低维的抽象表示，考虑到扩散模型对多模态分布的强大建模能力，我们使用扩散模型来实现这一过程，即 $P_\sigma(z|s_t)$。

我们将编码器生成的关键状态特征 z 作为标签，通过直接预测原始的关键状态特征的方式来计算重构损失，从而训练扩散模型 P_σ，Jun [22]等人的研究表明，这种方式在隐空间中比预测噪声 ε 训练模型效果更好。

$$L(\sigma)=E_{z^0\sim E_\phi(z|M),z^K\sim q(z^K|z^0)}\left(\left\|z^0-\mu_\sigma\left(z^K,s_t,k\right)\right\|^2\right) \tag{21}$$

值得注意的是，在轨迹规划的流程中，需要用到两次扩散模型：用于生成关键状态的 P_σ 和用于生成轨迹的 G_θ，这是两个不同参数的网络。

对于 P_σ 我们用无分类器引导的方式来简单高效的训练模型，然而一些研究表明[23]，无分类器引导方式可能促使生成模型倾向于生成分布中低概率区域的轨迹。这些样本往往偏离训练数据的主流分布，

从而为生成模型引入噪声。

为解决这一问题，我们设计了一种筛选机制，以确保生成的关键状态特征 z 更贴近数据分布的特性。Pearce 等人[23]提出了一种基于核密度估计(Kernel Density Estimation, KDE)的动作采样方法。我们在此基础上对其进行了改进，并将其应用于筛选扩散模型生成的关键状态特征。核密度估计是一种非参数估计方法，与参数估计不同，它不预先假设数据服从特定的分布形式，而是直接通过数据本身来估计分布。KDE 通过计算每个候选样本 z 在特征空间中的密度值来评估其代表性。密度较高的样本通常位于数据分布的高密度区域，因此更可能被视为高质量的样本，适合用于后续的任务。

传统的核密度估计通常基于欧氏距离计算样本间的相似性，这种方法在面对简单一维或低维数据时效果良好，但其局限在于忽略了数据维度之间的相关性。特别是在关键状态特征空间这种相对高维的情景中，欧氏距离可能导致密度估计不够精确。为此，我们引入马氏距离(Mahalanobis Distance)替代欧氏距离，马氏距离通过融入数据的协方差矩阵，能够更准确地反映数据点之间的真实距离，从而提升密度估计的精确性和筛选效果。改进后的核密度估计公式定义为：

$$\hat{p}(z)=\frac{1}{n}\sum_{i=1}^{n}K_M\left(\frac{z-z_i}{h}\right) \tag{22}$$

其中 z 是待评估的候选样本，z_i 是 n 个候选样本中的第 i 个样本，h 是带宽(bandwidth)，用于控制核函数的覆盖范围，h 越大覆盖范围越广，密度估计越平滑。核函数 $K_M(\cdot)$ 是 KDE 的核心，我们使用高斯核函数，其定义为：

$$K_M\left(\frac{z-z_i}{h}\right)=\frac{1}{\left(2\pi h^2\right)^{d/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2h^2}(z-z_i)^{\mathrm{T}}\Sigma^{-1}(z-z_i)\right) \tag{23}$$

核函数的作用是基于马氏距离计算 z_i 和 z 的相似度。d 是 z 的维度。$(z-z_i)^{\mathrm{T}}\Sigma^{-1}(z-z_i)$ 是通过协方差矩阵 Σ 计算的马氏距离，能够捕捉维度间的相关性。指数部分能使高斯核在 z 接近 z_i 时赋予较高的权重，随着距离增加权重呈指数衰减，能够有效捕捉数据分布的局部结构。公式的系数为归一化因子，确保核函数的输出满足概率密度的性质。协方差矩阵计算公式为：

$$\Sigma=\frac{1}{n-1}\sum_{i=1}^{n}(z_i-\mu)(z_i-\mu)^{\mathrm{T}} \tag{24}$$

$$\mu=\frac{1}{n}\sum_{i=1}^{n}z_i \tag{25}$$

通过改进的 KDE 概率密度估计公式，我们对每个生成的 z 样本计算其概率密度 $\hat{p}(z)$。最终，最优样本 z^* 通过以下方式筛选得出：

$$z^*=\underset{z\in\{z_1,z_2,\cdots,z_n\}}{\arg\max}\ p(z) \tag{26}$$

在获得筛选后的关键状态特征后，其作为扩散模型 G_θ 的条件输入，指导模型生成状态序列，并且利用逆动力模型生成动作序列，其中第一个动作被智能体执行与环境交互。当发生状态转移后，根据当前状态 s_{t+1} 重复轨迹规划流程。

3. 实验结果与分析

3.1. 实验环境和数据集

本研究在 D4RL [24]数据集上进行实验，该数据集是离线强化学习领域的标准基准，提供多样化的任

务环境以评估算法性能。实验涉及 Gym-Mujoco 任务(如 HalfCheetah、Hopper、Walker2d)，聚焦智能体的基础运动控制，如奔跑和跳跃；Adroit 任务(如 Pen、Door、Hammer)，挑战高维动作空间中的复杂机械臂操作，如抓取和开门；Maze2D 任务，测试智能体在二维迷宫中的路径规划能力；以及 AntMaze 任务，控制四足机器人 Ant 在复杂迷宫中完成长期导航，结合路径规划与机器人控制。实验严格遵循 D4RL 的标准化评估协议，通过在多场景、多难度级别上的组合测试，系统地验证了算法在不同数据分布下的泛化能力。通过上述任务和数据集的设置，本研究旨在全面评估所提算法在离线强化学习中的性能表现，涵盖从基础运动控制到复杂操作和长期导航的多种场景。

3.2. 对比实验

在本节中，我们通过对比实验来评估 KSDP 方法在离线强化学习数据集中的表现。我们选择了多种基线方法，包括模仿学习方法行为克隆(BC)；基于值的方法批量约 Q 学习(BCQ) [5]，保守 Q 学习(CQL) [6]，隐式 Q 学习(IQL) [9]；传统轨迹规划方法 MPPI [25]，MoReL [7]，HiGoC [26]，MBOP [27]；基于 Transformer 的生成模型方法轨迹变换器(TT) [10]，决策变换器(DT) [12]；基于扩散模型的方法 Diffuser [15]，DD [28]。这些基线方法涵盖了离线强化学习轨迹规划方法的主流范式，以确保对 KSDP 的全面评估，实验结果如下。

Gym-Mujoco 是一个经典的连续控制任务数据集，其中高维的连续动作空间为模型决策提出了挑战。尤其是 replay 和 medium 数据中包含大量次优轨迹，对模型的鲁棒性要求较高。如表 1 所示，KSDP 在 Gym-Mujoco 数据集的平均得分达到 82.7，显著高于所有基线方法，包括 Diffuser (77.5)、DT (74.7)和 TT (78.9)。特别是在 medium-replay 中，KSDP 的平均得分 72 相比于 Diffuser 的 67.3 约提升了 7%的性能。KSDP 的关键状态特征提取机制能够有效过滤次优数据中的噪声，结合逆动力解码器生成高质量的动作序列。这使其在面对次优数据和动态变化时表现出较强的鲁棒性。

Table 1. Comparative experimental results on Gym-Mujoco datasets
表 1. Gym-Mujoco 数据集中的对比试验结果

Datasets	BC	MBOP	MoReL	TT	DT	Diffuser	KSDP
halfcheetah-medium-expert-v2	55.2	**105.9**	53.3	95.0	86.8	88.9	89.3
walker2d-medium-expert-v2	107.5	70.2	95.6	101.9	**108.1**	106.9	107.4
hopper-medium-expert-v2	52.5	55.1	108.7	110.0	107.6	103.3	**110.5**
halfcheetah-medium-v2	42.6	44.6	42.1	46.9	42.6	42.8	**47.1**
walker2d-medium-v2	75.3	41.0	77.8	79.0	74.0	79.6	**80.7**
hopper-medium-v2	52.9	48.8	**95.4**	61.1	67.6	74.3	92.4
halfcheetah-medium-replay-v2	36.6	42.3	40.2	41.9	36.6	37.7	**42.5**
walker2d-medium-replay-v2	26.0	9.7	49.8	**82.6**	66.6	70.6	79.2
hopper-medium-replay-v2	18.1	12.4	93.6	91.5	82.7	93.6	**94.3**
Average	51.9	47.8	72.9	78.9	74.7	77.5	**82.7**

Maze2D 是一个二维导航任务数据集，该任务环境是一个典型了稀疏奖励环境，只有当智能体到达目标时才会获得奖励反馈，这显著增加了模型的规划难度。相比其他数据集，虽然动作–状态空间的维度较低，但路径规划复杂。此外，在 Multi2D 设置下的每个 episode 开始时目标位置是随机初始化的，由于目标位置的随机性，环境的复杂度和不确定性更高，智能体需要具备更强的适应性和泛化能力。如表 2 所

示，KSDP 在 Maze2D 数据集上平均得分达到 92.4，为所有基线方法中最高。AntMaze 数据集使用 MuJoCo 的 Ant 机器人进行导航，迷宫环境和 Maze2D 相同，机器人运动涉及多关节协调，增加了控制难度。如表 3 所示，KSDP 在 AntMaze 数据集上的平均得分达到 82.7，证明了在稀疏奖励任务中，KSDP 借助逆动力解码器同样可以适应高维的动作空间控制任务，展现了对高维状态和复杂动态的处理能力。KSDP 在稀疏奖励环境下的优异表现归功于其条件扩散模型和关键状态引导机制。扩散模型能够生成覆盖长时程的高质量轨迹，而关键状态提取帮助模型聚焦于导航中的重要决策点。

Table 2. Comparative experimental results on AntMaze datasets
表 2. AntMaze 数据集中的对比试验结果

Datasets	MPPI	IQL	HiGoC	Diffuser	DD	KSDP
Maze2D-U-Maze-3	14.4	23.2	61.2	82.6	83.9	**89.8**
Maze2D-U-Maze-3	5.7	19.8	59.8	87.8	85.8	**92.2**
Maze2D-Large-2	3.9	31.1	45.4	87.9	87.3	**93.4**
Multi2D-U-Maze-3	17.8	16.5	67.9	85.4	86.9	**95.1**
Multi2D-Medium-2	8.1	8.9	52.4	85.6	88.2	**92.6**
Multi2D-Large-2	4.5	10.3	42.1	89.3	**91.7**	91.2
Average	9.1	18.3	54.8	86.4	87.3	**92.4**

Table 3. Comparative experimental results on Maze2D datasets
表 3. Maze2D 数据集中的对比试验结果

Datasets	IQL	HiGoC	Diffuser	KSDP
AntMaze-U-Maze	62.2	**91.2**	76.0	85.2
AntMaze-Medium	70.0	79.3	45.5	**82.6**
AntMaze-Large	47.5	67.3	22.0	**80.3**
Average	59.9	79.3	47.8	**82.7**

Adroit 数据集是离线强化学习中最具挑战性的任务之一，涉及超高维手部操作任务，同时需要模拟手的物理交互，动态建模难度大。如表 4 所示，KSDP 在 pen-cloned 任务中获得最高得分 47.7，优于 CQL 和 IQL 方法，但在 hammer 和 door 任务中得分较低(分别为 2.8 和 1.8)，所有方法均表现不佳。KSDP 在 pen-cloned 中的表现显示其在高难度任务中仍有潜力，关键状态提取和扩散模型能够在一定程度上捕捉任务的关键特征。手部操作任务需要极高的动作精度和复杂的动态建模，KSDP 当前的设计在这方面的优化不足，导致性能受限。我们分析，Adroit 的超高维动作空间和物理交互复杂性，超出了 KSDP 对于当前特征提取和压缩表征的能力，模型难以生成满足任务要求的精确动作，未来可进一步优化 KSDP 在此类任务中的表现。

Table 4. Comparative experimental results on Adroi datasets
表 4. Adroi 数据集中的对比试验结果

Datasets	BC	BCQ	IQL	CQL	KSDP
pen-cloned	37.0	44.0	37.3	39.2	**47.7**
hammer-cloned	0.6	0.4	2.1	2.1	**2.8**
door-cloned	0.0	0.0	1.6	0.4	**1.8**

3.3. 消融实验

3.3.1. 关键状态提取模块消融实验和可视化分析

为了评估关键状态特征提取模块中各组成部分对轨迹生成性能的贡献，我们设计并开展了消融实验。本实验旨在分析Transformer编码器和β-VAE编码器在KSDP模型中的作用及其对生成轨迹质量的影响。通过控制变量的方式，所有模型在相同的训练和评估条件下进行，确保结果的公平性和可比性。我们设计了以下三种版本的关键状态特征提取模块进行对比实验：

β-VAE 编码器提取关键状态：移除 Transformer 编码器，直接将原始轨迹输入 β-VAE 编码器提取关键状态特征。为适配序列输入，我们使用 RNN 网络在 β-VAE 前添加了一个简单的序列编码层。

随机关键状态：不使用任何编码器，直接将 Transformer 嵌入层的轨迹向量通过全连接层映射到关键状态特征维度，将其输入条件扩散模型作为条件。

KSDP：输入轨迹首先通过 Transformer 编码器处理，生成上下文增强的特征序列，随后输入 β-VAE 编码器，压缩为关键状态特征。这些特征作为条件输入条件扩散模型，最终生成完整轨迹。

实验基于 D4RL 数据集中的 Gym-Mujoco 任务开展，实验结果如表 5 所示。

Table 5. Ablation study results of Key-State extraction module
表 5. 关键状态提取模块消融实验结果

Datasets	β-VAE 提取关键状态	随机关键状态	KSDP
halfcheetah-medium-v2	42.3	34.7	**47.1**
walker2d-medium-v2	76.5	68.4	**80.7**
hopper-medium-v2	88.4	74.2	**92.4**
halfcheetah-medium-replay-v2	37.1	31.6	**42.5**
walker2d-medium-replay-v2	76.6	64.3	**79.2**
hopper-medium-replay-v2	88.9	78.5	**94.3**

实验结果表明，KSDP 取得了最佳表现，显著优于随机关键状态特征的模型。这一结果验证了 Transformer 编码器和 β-VAE 编码器在关键状态提取中的重要性和协同效应。KSDP 充分利用了 Transformer 的自注意力机制捕捉轨迹中重要状态信息和长期依赖关系，以及 β-VAE 的潜空间压缩能力提取关键信息。扩散模型通过关键状态的引导显著提升了生成轨迹的质量。

与仅使用 β-VAE 编码器的模型相比，KSDP 的性能提高了约 6%。这表明 Transformer 编码器在理解轨迹序列的上下文和动态关系方面至关重要，仅靠简单的 RNN 网络无法替代其功能。Transformer 通过自注意力机制能够直接建模序列中任意两个元素之间的关系，克服了 RNN 在处理长序列时的梯度消失问题，能够更好地提取上下文特征。

为深入分析关键状态特征的分布特性，我们先将关键状态特征映射到对应任务的状态空间，然后使用 t-SNE (t-distributed Stochastic Neighbor Embedding)将其高维特征降维至二维空间进行可视化。如图 5 所示，图左为原始数据集中状态空间，图右为提取到的关键状态空间。

如图所示，关键状态特征能够很好地表征原始数据分布，同时呈现出清晰的聚类结构。不同轨迹类别的特征分布明显分离，表明其成功捕获了轨迹的语义信息和动态模式，凸显了 Transformer 编码器和 β-VAE 编码器协同作用的显著效果。

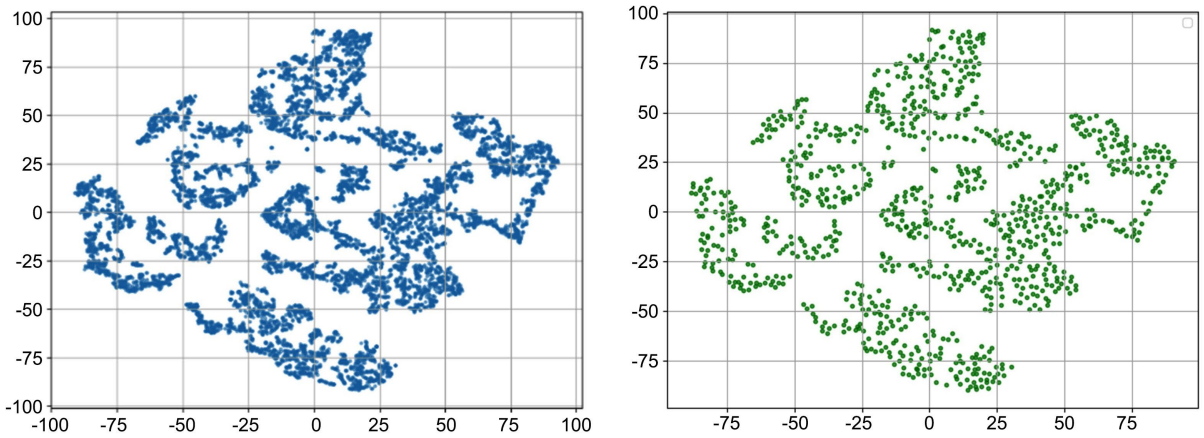

Figure 5. Original trajectory spatial distribution map (left), key state spatial distribution map (right)
图 5. 原始轨迹空间分布图(左)，关键状态空间分布图(右)

3.3.2. 轨迹价值引导消融实验

在 KSDP 方法中，轨迹价值预测模块通过预测关键状态所表征轨迹的价值 V，为扩散模型的采样过程提供价值梯度引导，从而改进了传统无分类器引导的轨迹生成方式。为了评估该模块对生成轨迹质量的贡献，我们设计了消融实验，对比 KSDP 基准模型与移除价值引导的变体模型，分析不同引导策略对性能的影响。具体而言，我们设置了以下两种版本的模型进行对比：

无价值引导模型：移除轨迹价值预测器，仅依赖无分类器引导进行轨迹采样。这种配置依赖扩散模型的原始生成能力，未引入额外的价值信息。

KSDP：同时使用无分类器引导和轨迹价值引导。

实验基于 D4RL 数据集中的 Maze2D 任务进行评估。该任务具有稀疏奖励和长时程依赖的特点，因此适合检验轨迹生成质量的优劣。实验结果如表 6 所示。

Table 6. Ablation study results of trajectory value guidance
表 6. 轨迹价值引导消融实验结果

Datasets	无价值引导模型	KSDP
Maze2D-U-Maze-3	83.1	**89.8**
Maze2D-Medium-2	86.5	**92.2**
Maze2D-Large-2	84.7	**93.4**

实验结果表明，KSDP 在所有 Maze2D 子任务中的表现均优于无价值引导模型，平均分数提升约 8.3%。在复杂度最高的 Maze2D-Large-2 中，KSDP 得分比无价值引导模型提高 10%。这种性能提升可以归因于轨迹价值引导的引入。Maze2D 任务的稀疏奖励特性要求模型生成能够有效连接起点和目标的长时程轨迹，而无分类器引导仅依赖于数据分布的先验知识，容易生成偏离最优路径的轨迹。相比之下，KSDP 通过价值预测模块为扩散模型提供了额外的优化方向，使得采样过程更倾向于生成高价值的轨迹。这种引导机制在高复杂度迷宫环境尤为显著，因为更复杂的迷宫需要更强的目标导向能力，而价值梯度恰好弥补了无分类器引导在这方面的不足。

本消融实验展示了轨迹价值引导在扩散模型轨迹生成中的重要性。通过结合无分类器引导和轨迹价值梯度引导的方式，为生成高价值轨迹提供了强有力的支持。

3.3.3. 筛选机制消融实验

在 KSDP 算法的轨迹规划过程中，关键状态筛选机制是其核心组成部分。该机制利用基于核密度估计的方法，从扩散模型生成的多个关键状态候选样本中筛选出最优样本，以减少噪声并确保关键状态符合数据分布特性。为了评估这一机制的作用，我们设置了以下两种版本的模型进行对比：

无筛选机制模型：移除筛选机制，直接使用扩散模型生成的一个关键状态样本作为轨迹生成的条件输入。

KSDP：保留完整的筛选机制。

为确保实验的可控性，两种模型在状态轨迹生成和动作执行阶段保持一致，仅在关键状态筛选机制的有无上有所差异。实验在 D4RL 数据集中的 AntMaze 任务上进行，该任务环境状态空间更加复杂，在关键状态特征生成时容易产生噪声，实验结果如表 7 所示。

Table 7. Ablation study results of the selection mechanism
表 7. 筛选机制消融实验结果

Datasets	无筛选机制模型	KSDP
AntMaze-U-Maze	84.7	**85.2**
AntMaze-Medium	80.4	**82.6**
AntMaze-Large	76.1	**80.3**

实验结果表明，KSDP 在所有 AntMaze 子任务中的成功率均高于无筛选机制模型。在 U-Maze 和 Medium 环境下，无筛选机制模型与 KSDP 分数相差不大，我们分析在相对简单的任务环境，即使没有筛选机制，模型也能生成较为合理的轨迹，因此分数下降幅度较小。在 Large 环境下，KSDP 分数相较于无筛选机制模型有较大提升。这表明在高维复杂任务中，筛选机制的作用尤为重要，能够有效减少噪声、优化关键状态的选择，从而提升轨迹规划的成功率。

4. 结论

本文提出了一种基于关键状态的扩散模型轨迹规划方法，该方法利用 Transformer 和 β-VAE 模型从原始轨迹中提取出关键状态特征，并以此作为条件驱动扩散模型生成状态轨迹，随后通过逆动力模型生成相应的动作序列。在轨迹规划过程中，引入了基于核密度估计的关键状态筛选机制，有效降低了模型生成过程中的噪声干扰。在离线强化学习标准数据集 D4RL 中的多个任务环境进行的实验表明，KSDP 的模型性能优于目前的先进方法，且在次优数据集和复杂环境中的鲁棒性更高。最后，通过对比实验和消融实验，在多个数据集上验证了 KSDP 的性能优势，充分展示了其在轨迹规划任务中的有效性和鲁棒性。未来可探索更高效的扩散模型架构，以解决其计算复杂性并缩短模型采样时间。

参考文献

[1] Singh, B., Kumar, R. and Singh, V.P. (2021) Reinforcement Learning in Robotic Applications: A Comprehensive Survey. *Artificial Intelligence Review*, **55**, 945-990. https://doi.org/10.1007/s10462-021-09997-9

[2] Tang, C., Abbatematteo, B., Hu, J., *et al*. (2024) Deep Reinforcement Learning for Robotics: A Survey of Real-World Successes. arXiv:2408.03539.

[3] Wang, Z., Yan, H., Wei, C., Wang, J., Bo, S. and Xiao, M. (2024) Research on Autonomous Driving Decision-Making Strategies Based Deep Reinforcement Learning. *Proceedings of the* 2024 4*th International Conference on Internet of Things and Machine Learning*, Nanchang, 9-11 August 2024, 211-215. https://doi.org/10.1145/3697467.3697643

[4] Levine, S., Kumar, A., Tucker, G., *et al*. (2020) Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. arXiv:2005.01643.

[5] Fujimoto, S., Meger, D. and Precup, D. (2019) Off-Policy Deep Reinforcement Learning without Exploration. *International Conference on Machine Learning*, Long Beach, 10-15 June 2019, 2052-2062.

[6] Kumar, A., Zhou, A., Tucker, G., *et al*. (2020) Conservative Q-Learning for Offline Reinforcement Learning. *Advances in Neural Information Processing Systems*, **33**, 1179-1191.

[7] Kidambi, R., Rajeswaran, A., Netrapalli, P., *et al*. (2020) Morel: Model-Based Offline Reinforcement Learning. *Advances in Neural Information Processing Systems*, **33**, 21810-21823.

[8] Zhan, X., Zhu, X. and Xu, H. (2022) Model-Based Offline Planning with Trajectory Pruning. *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, Vienna, 23-29 July 2022, 3716-3722. https://doi.org/10.24963/ijcai.2022/516

[9] Kostrikov, I., Nair, A. and Levine, S. (2021) Offline Reinforcement Learning with Implicit Q-Learning. arXiv:2110.06169.

[10] Janner, M., Li, Q. and Levine, S. (2021) Offline Reinforcement Learning as One Big Sequence Modeling Problem. *Advances in Neural Information Processing Systems*, **34**, 1273-1286.

[11] Vaswani, A., Shazeer, N., Parmar, N., *et al*. (2017) Attention Is All You Need. arXiv: 1706.03762.

[12] Chen, L., Lu, K., Rajeswaran, A., *et al*. (2021) Decision Transformer: Reinforcement Learning via Sequence Modeling. *Advances in Neural Information Processing Systems*, **34**, 15 084-15097.

[13] Esser, P., Kulal, S., Blattmann, A., *et al*. (2024) Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. arXiv: 2403.03206.

[14] Zhu, Z., Zhao, H., He, H., *et al*. (2023) Diffusion Models for Reinforcement Learning: A Survey. arXiv:2311.01223.

[15] Janner, M., Du, Y., Tenenbaum, J.B., *et al*. (2022) Planning with Diffusion for Flexible Behavior Synthesis. arXiv:2205.09991.

[16] Shaw, P., Uszkoreit, J. and Vaswani, A. (2018) Self-Attention with Relative Position Representations. arXiv:1803.02155.

[17] Higgins, I., Matthey, L., Pal, A., *et al*. (2017) Beta-Vae: Learning Basic Visual Concepts with a Constrained Variational Framework. *International Conference on Learning Representations*, Toulon, 24-26 April 2017, 1-13.

[18] Tedrake, R. (2009) Underactuated Robotics: Learning, Planning, and Control for Efficient and Agile Machines Course Notes for MIT 6.832. Working Draft Edition, 1-13.

[19] Pathak, D., Mahmoudieh, P., Luo, G., *et al*. (2018) Zero-Shot Visual Imitation. arXiv:1804.08606.

[20] Ho, J. and Salimans, T. (2022) Classifier-Free Diffusion Guidance. arXiv:2207.12598.

[21] Ho, J., Jain, A. and Abbeel, P. (2020) Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems*, **33**, 6840-6851.

[22] Jun, H. and Nichol, A. (2023) Shape: Generating Conditional 3D Implicit Functions. arXiv:2305.02463.

[23] Pearce, T., Rashid, T., Kanervisto, A., *et al*. (2023) Imitating Human Behaviour with Diffusion Models. arXiv:2301.10677.

[24] Fu, J., Kumar, A., Nachum, O., *et al*. (2020) D4RL: Datasets for Deep Data-Driven Reinforcement Learning. arXiv:2004.07219.

[25] Williams, G., Drews, P., Goldfain, B., Rehg, J.M. and Theodorou, E.A. (2016) Aggressive Driving with Model Predictive Path Integral Control. 2016 *IEEE International Conference on Robotics and Automation* (*ICRA*), Stockholm, 16-21 May 2016, 1433-1440. https://doi.org/10.1109/icra.2016.7487277

[26] Li, J., Tang, C., Tomizuka, M. and Zhan, W. (2022) Hierarchical Planning through Goal-Conditioned Offline Reinforcement Learning. *IEEE Robotics and Automation Letters*, **7**, 10216-10223. https://doi.org/10.1109/lra.2022.3190100

[27] Argenson, A. and Dulac-Arnold, G. (2020) Model-Based Offline Planning. arXiv:2008.05556.

[28] Ajay, A., Du, Y., Gupta, A., *et al*. (2022) Is Conditional Generative Modeling All You Need for Decision-Making? arXiv:2211.15657.