arXiv:2304.02532v2 [cs.LG] 1 Jun 2023

使用基于分数的扩散策略进行目标条件模仿学习

Moritz Reuss、Maximilian Li、Xiaogang Jia 和 Rudolf Lioutikov，Intuitive Robots 实验室，德国卡尔斯鲁厄理工学院

摘要—**我们提出了一种基于基于分数的扩散模型（SDMs）的新型策略表示。我们将这种新型策略表示应用于目标条件模仿学习（GCIL），以便从大型、未经筛选的数据集中学习通用的目标指定策略，而无需奖励。我们提出的新型目标条件策略架构"基于分数的扩散策略行为生成"（BESO）将生成式基于分数的扩散模型用作其策略。BESO 将得分模型的学习与推理采样过程解耦，因此能够采用快速采样策略，仅用 3 个去噪步数生成目标指定行为；其他基于扩散模型的策略则需要 30+ 步。此外，BESO 表达能力很强，能够有效捕捉游戏数据解空间中的多模态特征。与 Latent Plans 或 C-BeT 等先前方法不同，BESO 无需依赖复杂的层次化策略或额外的聚类，即可有效学习目标条件行为。最后，我们展示了如何利用无分类器引导，通过游戏数据学习目标无关策略。据我们所知，这是首项同时满足以下条件的工作：a）使用这种解耦的基于分数的扩散模型来表示行为策略；b）在 GCIL 领域学习基于基于分数的扩散模型的策略；c）提供一种从游戏数据中同时学习目标相关策略和目标无关策略的方法。我们通过详细的仿真实验评估 BESO，并展示它在具有挑战性的基准测试中始终优于多种最先进的目标条件模仿学习方法。此外，我们还提供了广泛的消融研究和实验，以展示该方法在目标条件行为生成方面的有效性。演示数据和代码可在 https://intuitive-robots.github.io/beso-website 获取。**

I. 引言

目标条件行为学习旨在训练能够处理各种日常任务的通用具身智能体。应对这一挑战的一种常见方法是目标条件模仿学习（GCIL）。GCIL 仅需离线数据集，无需额外奖励或昂贵的环境交互即可进行训练。然而，GCIL 通常需要一组预先定义的任务，并且每项任务都需要大量带标签且经过分段的专家轨迹，这既耗费成本又耗费时间。此外，它对新场景和不同任务的泛化能力较弱。从游戏中学习（LfP） [21] 并非教智能体学习数量有限的预定义目标，而是提供了一种有效方式，用于收集与任务无关、由远程操控、未经筛选且形式自由的数据集。这类数据集包含丰富而有意义的多模态环境交互，覆盖状态空间的不同区域。LfP 不会手动标注轨迹，而是将每条轨迹中的随机序列与一个或多个未来状态（即相应轨迹的目标状态）配对。目标条件策略从收集到的游戏交互数据中提炼出有用的目标导向行为。然而，从游戏数据中学习仍是一项开放性挑战，部分原因在于演示数据具有多模态特性，例如，同一任务可以通过截然不同的方式完成，而不同任务也可能以非常相似的方式完成。

从这些数据集中有效学习行为需要策略能够保持这种多模态解决方案，并且具有足够的表达能力，以保持接近离线数据中所见的状态-动作分布，从而执行长时程技能。大多数先前的工作试图通过结合生成模型来应对这一挑战，例如变分自编码器 (VAEs) [12, 25, 32] 和生成式预训练变换器 (GPTs) [6, 35], 以及额外的模型和网络，以显式编码多模态性或层级结构。然而，这些方法需要补充网络或在其架构内分离技能执行和规划，因为策略表达不足或无法处理观察到的行为的多模态性。此外，通常需要多个学习目标，例如用于低层和高层策略，这带来了额外的调优挑战。

我们提出了一种新颖的方法，**BE**havior Generation using**Sc**ore-based Diffusion models (BESO)，它擅长仅从无奖励的离线数据集中学习目标条件策略。BESO 使用基于分数的扩散模型 (SDMs) [37, 15, 41, 17], , 这是一类新的生成模型，通过正向随机微分方程 (SDE) 逐步将数据扩散到噪声。通过训练一个称为分数或去噪模型的神经网络来近似分数函数，可以逆转 SDE，通过迭代采样过程从噪声中生成新样本。

我们展示了使用基于分数的扩散模型对目标条件动作分布进行建模的多项优势。首先，我们证明，基于分数的扩散模型的表达能力及其捕获多模态分布的能力，是实现有效条件行为生成的关键。在多个具有挑战性的目标条件基准测试中，包括条件化的 Relay Kitchen 和 Block-Push 环境 [6], , BESO始终优于 C-BeT 和潜在运动计划等最先进的方法 [6, 21]。其次，通过利用基于分数的扩散模型的分类器自由引导训练，BESO能够有效地同时学习两种策略：目标依赖策略和目标无关策略；测试时，这两种策略既可以结合使用，也可以单独使用。第三，我们的模型只需一个训练目标即可轻松、稳定地训练，无需额外奖励。这与隐式行为克隆（IBC） [10], 等其他最先进的生成模型形成对比，后者或

层次化策略 [12]。第四，与变分自编码器或基于能量的模型（EBM）等其他生成模型不同，基于分数的扩散模型不会限制模型架构的选择 [10]。因此，我们采用一种新颖的 Transformer 架构，并结合预条件技术，根据一系列观测和期望的目标状态合成分步动作。最后，BESO能够快速扩散生成新动作。当前基于扩散的策略 [30] 通常需要 30+ 个去噪步数才能预测单个动作并取得良好效果，而我们提出的 BESO 在具有挑战性的 GCIL基准上表现出色，以仅需 3 个去噪步数的优势超越了最先进的目标条件策略。我们通过采用基于分数的扩散模型领域的最新进展，将训练过程与推理过程分离 [17] ，并应用专为快速扩散推理设计的新型数值求解器 [38, 19]，实现了这一点。因此，我们系统地评估了基于分数的扩散模型在快速、有效地生成分步动作方面的关键组成部分。

总结我们的贡献如下：

- BESO 是一种基于分数的扩散模型的新型策略表示，可利用未经筛选的游玩数据有效生成目标条件行为
- 使用基于无分类器引导的扩散策略，同时从游戏数据中学习依赖目标和不依赖目标的策略
- 系统评估了结合基于分数的扩散策略实现快速、高效动作生成的关键组件，并进行了广泛的实验和消融研究

II. 相关工作

扩散生成模型。基于分数的生成模型（SGM） [39, 40] 和去噪扩散概率模型（DDPM） [37, 15] 是基于分数的扩散模型（SDM）的两种不同变体。这些模型通过不断增加的高斯噪声破坏数据分布，并利用神经网络学习逆转这种破坏，从噪声中生成新的数据样本。随机微分方程（SDE）框架统一了这两种不同的模型 [41]。SDE 将扩散过程描述为连续时间过程，而非使用离散的噪声水平。BESO 遵循 Karras 等人提出的 SDE 表述[17]。为了从扩散模型中采样新样本，需要反转按 T 时间步离散化的 SDE。SDE 包含一个具有相同边缘分布的概率流 常微分方程（ODE），从而实现快速采样 [41]。ODE 求解器在推理过程中不会添加噪声，因此可以减少函数评估次数并加快采样速度 [19]。使用专为扩散推理设计的专用数值 ODE 求解器还可进一步加快采样 [15, 17, 20]。SDM 在多种任务中取得了最先进的结果，包括图像生成 [17]、基于文本的图像合成 [7, 33]和人体动作生成 [42]。

目标条件模仿学习（GCIL）。它是模仿学习的一个子领域，其中每条演示数据都附带一个或多个目标状态，用以指明该演示数据所对应的任务。目标状态包含学习方法可利用的信息，有助于区分不同的演示数据。因此，目标条件策略，即将目标状态纳入条件集合的策略，可以根据给定的目标状态相应地调整行为。同样，目标状态也通过目标条件强化学习（GCRL）扩展了强化学习领域：智能体获得的不是专家示范，而是奖励信号。通常，这些奖励信号很难定义，尤其是在复杂任务和环境中；在这种情况下，提供演示数据往往是更自然的选择。此外，GCRL 所需的策略展开在现实场景中通常代价高昂。近期研究探讨了目标条件离线强化学习，这种方法在训练期间无需进行这些代价高昂的滚动执行。

从游戏中学习。 从从游戏中学习 (LfP)[21]旨在从多样化的无标签状态-动作轨迹中学习由目标指定的行为。传统的模仿学习数据集通常由狭窄状态空间中的单模态、分段专家轨迹构成。另一方面，游戏数据的特点是包含未分段的多模态轨迹。这使得学习有意义的行为更具挑战性，因为策略需要能够应对解决同一任务的多种方式，区分解决不同任务的相似方式，同时还需要具备长时域规划能力，以实现遥远未来的目标。以往的研究致力于从游戏数据中提取表征，以有效开展下游策略学习 [47]，或使用条件变分自编码器（条件变分自编码器） [12, 21, 25, 24]学习技能的自监督表征，并将其称为潜在计划。研究人员还探索了将基于Transformer的架构作为任务无关行为学习 [6, 35]的策略类别。另一类研究则侧重于数据本身，通过从以物体为中心的交互中学习，而非随机采样序列，来改进 LfP [3]。

策略学习中的生成模型。模仿学习可以表述为一个状态占用匹配问题，其目标是学习一种策略，使其匹配专家演示数据的状态占用分布。如今，可以通过现代生成模型架构来近似未知的专家演示数据。一种常见方法是使用生成对抗网络（GAN） [13, 11]。这类方法由一个生成器策略和一个判别器组成：生成器策略学习模仿专家观察到的行为，判别器则区分真实与伪造轨迹。这些方法在训练过程中需要大量滚动执行，这在我们的设定中并不可行。其他方法使用条件变分自编码器（CVAEs） [23, 32, 12, 24, 34]学习潜在嵌入，以表示潜在技能。近期研究还将基于能量的模型用作行为克隆的条件策略 [10]。归一化流也已被提出作为一种策略表示 [36]。

机器人中的扩散生成模型。大多数将扩散模型应用于机器人领域的方法都聚焦于离散的 DDPM 变体 [15]。DDPM 扩散模型已用于离线强化学习，以生成状态—

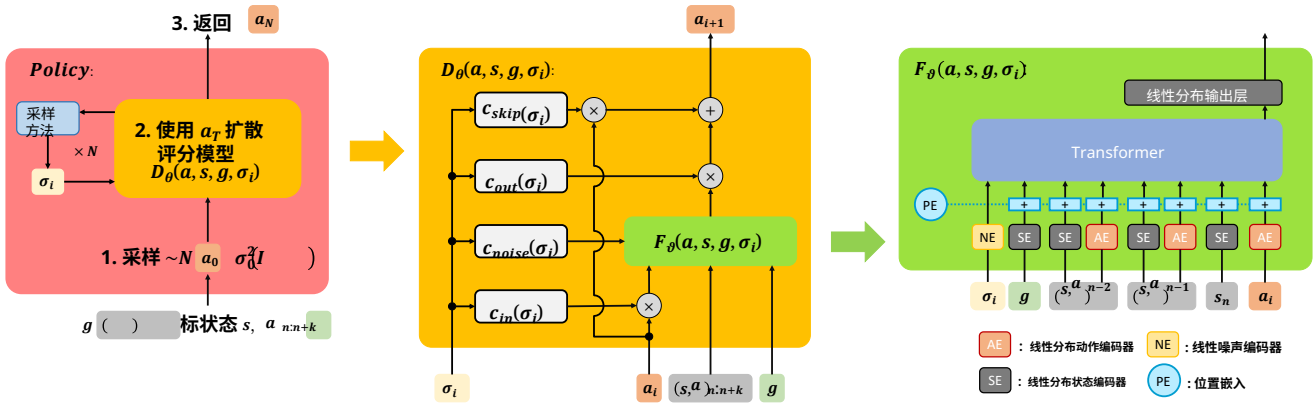


图 1. BESO 的动作生成过程及所使用的模型架构概览。**左侧：** 使用扩散过程对下一步动作进行去噪的一般动作生成过程：给定当前观测 s 和期望的目标状态 g，在 N 步内完成。**中间：** 高层评分模型及其预条件层和跳跃连接。**右侧：** 内部去噪评分模型采用 Transformer 架构和因果掩码，根据先前观测、动作以及目标序列，迭代预测去噪后的动作。

动作或仅状态轨迹，并采用大型 U-Net 架构[16, 1]。DDPM 也被用作基于步数的离线强化学习场景中的策略正则化方法，并与学习得到的 Q 函数结合使用 [45]。最近，基于分数的生成模型已被用于合成抓取位姿配置的代价函数 [43]。此外，研究者提出了条件式基于分数的生成模型，用于在逆强化学习中学习奖励函数 [18]。与 BESO 最相关的工作是 Diffusion Policy [5] 和 Diffusion-BC [30], , 两者都提出将条件式离散 DDPM 用作行为克隆的新策略类别。Diffusion-BC 通过 50 个随机采样步数合成新动作。为提升性能，Diffusion-BC 在最低噪声水平下额外使用X个推理步数，且不添加额外噪声。然而，这种方法会使动作生成进一步变慢。BESO 利用概率流 常微分方程，并结合快速、确定性采样器和经过优化的噪声水平。因此，BESO 在每次动作预测中所需的函数评估次数显著更少。

III. 问题表述与方法

本节将介绍我们使用基于分数的扩散模型进行目标条件行为生成的方法。

A. 问题表述

GCIL 的目标是从未经筛选的游玩数据中学习一个通用的目标条件策略。给定一组无结构、任务无关的轨迹，$\mathcal{T} = \left\{\boldsymbol{\tau}_k | \boldsymbol{\tau}_k = ((\boldsymbol{s}_n^k, \boldsymbol{a}_n^k))_{n=1}^{N_k}\right\}$，每条轨迹都可以拆分为一组元组，其中包含子轨迹序列和目标状态 $\mathcal{D}_k = \left\{ \boldsymbol{o}\ \boldsymbol{g} \mid \boldsymbol{o} = \boldsymbol{s}_n\ \boldsymbol{a}_n\ {}_{n=i}^{i_\dagger}\ \boldsymbol{g} = \boldsymbol{s}_n\ {}_{n=j}^{j_\dagger}\ \boldsymbol{s}_n\ \boldsymbol{a}_n \in \boldsymbol{\tau}_k \right\}$ (,) (,), (), (,)，其中 $i \le i_\dagger < j \le j_\dagger$ 分别表示序列和目标状态的起始与结束步数。按照这一表述

可以看出，目标状态必须是与该序列来自同一条轨迹的一个或多个状态，并且必须从该序列结束后的某一步开始。集合 $\mathcal{D}_k$可以包含相互重叠的序列，最终的游戏数据集表示为 $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$。为简单起见，$\boldsymbol{o}_k$ 和 $\boldsymbol{g}_k$ 的索引仅表示序列与目标状态彼此对应，而 $(\boldsymbol{s}_n, \boldsymbol{a}_n) \in \boldsymbol{o}$ 中的索引表示序列中的相对时间步。现在，将序列 $\boldsymbol{o}_k$ 中通向目标状态 $\boldsymbol{g}_k$ 的状态-动作对视为达到 $\boldsymbol{g}_k$[12, 25]的最优行为。目标条件策略试图最大化游戏数据集上的对数似然目标

$$\mathcal{L}_{\text{play}} = \mathbb{E}_{(\boldsymbol{o},\boldsymbol{g})\in\mathcal{D}}\left[\sum_{(\boldsymbol{s},\boldsymbol{a})\in\boldsymbol{o}} \log \pi_\theta\left(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{g}\right)\right]. \tag{1}$$

由于演示数据具有多模态特性，即多条轨迹可以通向同一个目标状态，因此要成功实现这一目标，需要策略能够表征此类多模态行为。

B. 基于分数的扩散策略

现在，我们的目标是学习游戏数据集 $\mathcal{D}$ 及其所给示范背后的策略分布 $\pi_{\mathcal{D}}(\boldsymbol{a}|\boldsymbol{s}, \boldsymbol{g})$。为此，我们定义一个连续扩散过程，通过逐步添加高斯噪声，将游戏数据集中的样本映射为中间分布 p_t, $t \in [0, T]$，其初始分布为 $p_0 = \pi_{\mathcal{D}}$，最终分布为 p_T。

连续扩散过程可以用随机微分方程（SDE） [41]描述。本文采用与近期提出的一种形式类似的方式定义该随机微分方程 [17]:

$$d\mathbf{a} = \left(\beta_t\sigma_t - \dot{\sigma}_t\right)\sigma_t\nabla_a \log p_t(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{g})dt + \sqrt{2\beta_t}\sigma_t d\omega_t, \tag{2}$$

其中，$\nabla_{\boldsymbol{a}} \log p_t(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{g})$表示分数函数，$\omega_t$ 是标准维纳过程，可将其理解为

无穷小高斯噪声。噪声调度器记为 σ_t，而 $\beta(t)$描述当前噪声被新噪声替换的相对速率。在我们的方法中，我们采用 $\sigma_t(t) = t$，这是一种已被证明在图像生成中有效的方法 [17]。在每个时间步 t 及其对应的噪声水平下，都存在相应的边缘分布 $p_t(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{g})$；该分布是向来自$p_{\text{play}}$的采样注入高斯噪声后得到的结果。这可以表示为 $p_t(\boldsymbol{a}_t|\boldsymbol{a}) = \mathcal{N}(\boldsymbol{a}, \sigma_t^2\mathbf{I})$。扩散过程最终的动作分布是一个已知且易于处理的先验分布 $\boldsymbol{a}_T = p_T$。在没有任何游戏数据分布信息的情况下，选择一个非结构化的高斯分布$p_T = \mathcal{N}(\mathbf{0}, \sigma_T^2\mathbf{I})$。

对于 BESO，我们尤其关注 SDE [4] 中的概率流 常微分方程 (ODE)p_t。该 ODE 在每个时间步都与 SDE 具有相同的边缘分布$p_t(\boldsymbol{a}|\boldsymbol{s},\ \boldsymbol{g})$，但不会额外注入随机噪声。令 $\beta(t) = 0$，即可从式 (2) 得到概率流ODE：

$$d\boldsymbol{a} = -\dot{\sigma}_t\sigma_t\nabla_{\boldsymbol{a}}\log p_t(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{g})dt \tag{3}$$

负分数函数 $-\nabla_{\boldsymbol{a}}\log\ p_t(\boldsymbol{a}|\boldsymbol{s},\ \boldsymbol{g})$ 指定了当前边缘分布 $p_t(\boldsymbol{a}|\boldsymbol{s},\ \boldsymbol{g})$ 的向量场。该向量场指向数据密度较低的区域，其尺度由当前噪声水平 $\dot{\sigma}_t$ 与其变化量 $\dot{\sigma}_t$ 的乘积决定。

算法 1 BESO 训练

要求： 游戏 数据集 $\mathcal{L}_{\text{play}}$，序列大小 c_o，Goal 序列大小 c_g
要求： 得分模型 $\mathbf{D}_\theta(\boldsymbol{a}, \boldsymbol{s}, \boldsymbol{g}, \sigma_t)$
要求： 噪声分布 $\sigma \sim$ 对数逻辑斯蒂分布(α, β)
对于 $i \in \{0, ..., N_{\text{train 步数}}\}$ **do**
  采样 $(\boldsymbol{o}, \boldsymbol{g}) \sim \mathcal{L}_{\text{play}}$
  采样 $\boldsymbol{\epsilon} \sim \mathcal{N}(\sigma_{\text{mean}}, \sigma_{\text{std}}^2\mathbf{I})$
  $\mathcal{L}_{D_\theta} \leftarrow \mathbb{E}_{\sigma,\boldsymbol{a},\boldsymbol{\epsilon}}\left[\alpha(\sigma_t)\|D_\theta(\boldsymbol{a}+\boldsymbol{\epsilon}, \boldsymbol{s}, \boldsymbol{g}, \sigma_t) - \boldsymbol{a}\|_2^2\right]$
结束循环

C. 扩散训练

为了通过数值近似反向常微分方程来生成新的采样结果，我们需要准确估计扩散过程中所有边缘分布的分数函数 $\nabla_{\boldsymbol{a}}\log\ p_t(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{g})$。为此，我们使用一个神经网络 $D_\theta(\boldsymbol{a}, \boldsymbol{s}, \boldsymbol{g}, \sigma_t)$，使其匹配所有边缘分布 $p_t(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{g})$ 的分数。

$$\nabla_{\boldsymbol{a}}\log p_t(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{g}) = \left(D_\theta(\boldsymbol{a}, \boldsymbol{s}, \boldsymbol{g}, \sigma_t) - \boldsymbol{a}\right)/\sigma_t^2. \tag{4}$$

神经网络使用去噪分数匹配目标进行训练 [44, 39],：我们向动作添加高斯噪声，并最小化网络输出与原始动作之间的差异：

$$\mathcal{L}_{D_\theta} = \mathbb{E}_{\sigma_{\mathbf{t}},\boldsymbol{a},\boldsymbol{\epsilon}}\left[\alpha(\sigma_t)\|D_\theta(\boldsymbol{a}+\boldsymbol{\epsilon}, \boldsymbol{s}, \boldsymbol{g}, \sigma_t) - \boldsymbol{a}\|_2^2\right], \tag{5}$$

其中，$\boldsymbol{a}$是一个动作样本，而 $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_t^2\mathbf{I})$表示高斯噪声。各个噪声水平下的损失为根据 $\alpha(\sigma_t)$加权，当前 σ_t 从噪声训练分布 p_{train}中采样。我们使用位置参数为 α 、尺度参数为 β的截断对数逻辑斯蒂分布： $p_{\text{train}} \sim$ LogLogistic(α, β)，取值范围为 $\{\sigma_{\min}, \sigma_{\max}\}$。训练过程总结于算法 1。这样，我们便能有效地学习扩散过程的噪声条件分数函数，并使用概率流常微分方程从条件密度 $p_t(\boldsymbol{a}|\boldsymbol{s},\boldsymbol{g})$中生成样本。

D. 使用确定性采样器进行高效动作生成

我们的策略通过从先验分布 $\boldsymbol{a}_T \sim \mathcal{N}(\mathbf{0}, \sigma_T^2\mathbf{I})$ 中采样来生成新动作，并在方程（3）中用训练得到的模型替代分数函数，从而对反向常微分方程或随机微分方程进行数值模拟。该过程首先从先验分布 $\boldsymbol{a}_T \sim \mathcal{N}(\mathbf{0}, \sigma_T^2\mathbf{I})$ 中随机选取一个采样结果，然后对该采样结果进行迭代去噪。使用随机采样结果作为起点，即使底层常微分方程是确定性的，也能够生成多样且具有多峰分布的动作。通过将微分方程离散化，可以从 T 开始到 0 对常微分方程进行数值求解。在动作预测过程中，我们在 N 个离散噪声水平上迭代地对采样结果进行去噪。BESO 采用 DDIM 求解器进行快速的确定性采样，具体描述见算法 2 [20, 38],。该求解器是一种基于指数积分方法的一阶确定性采样器。附录 B 的第 B 节详细比较了当前最先进的扩散采样器，并得出结论：DDIM 的总体性能最佳。关于噪声影响的额外评估表明，在动作预测任务中，常微分方程求解器的性能可与随机微分方程变体相媲美。我们在第 B 节进行的消融研究表明，BESO 只需三个去噪步数即可高精度地生成动作。进一步增加推理步数只能小幅提升性能，却会显著减慢采样过程。因此，我们发现 3 步在计算效率和性能之间达到了最佳平衡。在推理过程中，我们可以调整噪声范围和离散时间步的分布。基于经验评估，对于大多数应用，我们决定使用噪声范围为 $\sigma \in \{0.005, 1\}$ 的指数时间步。

IV. 目标引导的基于分数的扩散策略

本节介绍 BESO 的两种变体，用于合成目标条件行为的动作。

条件策略 (C-BESO)。我们通过直接使用基于分数的生成模型学习目标与状态条件分布，定义了一种目标条件扩散策略 $\pi(\boldsymbol{a}|\boldsymbol{s},\ \boldsymbol{g})$。与标准的目标条件行为克隆不同，我们的扩散策略能够捕捉游戏数据中存在的多种解，同时具有足够的表达能力来实现长期目标。

目标分类器自由引导策略（CFG-BESO）。此外，我们将 BESO 与扩散模型中一种流行的条件控制方法——无分类器引导（CFG） [14] 相结合。

算法 2 基于 DDIM 的动作生成过程
适用于 BESO [19, 38] 的采样器（DPM-1）

需要： 当前状态 $\boldsymbol{s}$，目标 $\boldsymbol{g}$
需要： 分数去噪模型 $D_\theta(\boldsymbol{a}, \boldsymbol{s}, \boldsymbol{g}, \sigma)$
要求： 离散时间步 $t_{i\in\{0,..,N\}}$
需要： 噪声调度器 $\sigma_i = t_i$
要求： $f_\beta(t) = -\log(t)$，$f_t(\beta) = \log(-\beta)$
绘制采样 $\boldsymbol{a}_0 \sim \mathcal{N}(\mathbf{0}, \sigma_0^2\mathbf{I})$
对于 $i \in \{0, ..., N-1\}$ **do**
  $\mathbf{d}_i \leftarrow \big(\boldsymbol{a}_i - D_\theta(\boldsymbol{a}_i, \boldsymbol{s}, \boldsymbol{g}, \sigma_i)\big)/\sigma_i$
  $\beta_{t_i}, \beta_{t_{i+1}} \leftarrow f_\beta(t_i)\ f_\beta(t_{i+1})$
  $h_i \leftarrow \beta_{t_i} - \beta_{t_{i-1}}$
  $\boldsymbol{a}_{i+1} \leftarrow (\frac{t_{i+1}}{t_i})\boldsymbol{a}_i - \big(e^{(-h_i)} - 1\big)\mathbf{d}_i$
结束循环
返回 $\mathbf{a}_N$

我们通过对目标 $\boldsymbol{g}$ 应用 0.1 的 dropout 比率来训练目标条件扩散策略 $\pi(\boldsymbol{a}|\boldsymbol{s}, \boldsymbol{g})$，这也会在我们的目标条件模型中训练一个隐式的目标无关策略 $\pi(\boldsymbol{a}|\boldsymbol{s})$。生成过程使用组合梯度来执行去噪过程

$$\nabla_a \log p_{t,\lambda}(\boldsymbol{a}|\boldsymbol{s}, \boldsymbol{g}) = \lambda\nabla_a \log p_t(\boldsymbol{a}|\boldsymbol{s}, \boldsymbol{g}) + (1-\lambda)\nabla_a \log p_t(\boldsymbol{a}|\boldsymbol{s}), \tag{6}$$

其中，引导因子 λ 用于平衡目标条件梯度与无目标条件梯度的影响。在扩散模型文献中，λ 通常介于 2 到 7。5 之间，以引导扩散模型朝目标条件分布 $\pi(\boldsymbol{a}|\boldsymbol{s}, \boldsymbol{g})$ 发展。与其他条件方法相比，CFG 已展现出显著的性能提升 [14, 19, 28]。尽管 CFG 也已成功应用于离线强化学习中的仅状态轨迹生成 [1], ，近期关于行为克隆的研究表明，在基于步进的动作生成中，CFG 的表现明显不如更简单的条件化方法 [30]。我们在实验部分详细分析了 CFG 在目标引导动作生成中的应用。

A. 模型架构

训练基于分数的扩散模型面临的主要挑战之一，是噪声水平 $\sigma_t \in \{0.001, 40\}$ 的范围很大。为应对这一挑战，我们采用了一种改进的架构 [17] ，其中包括额外的跳跃连接和两个预条件化层；这些层以当前噪声水平 σ_t 为条件。

$$D_\theta(\boldsymbol{a}|\boldsymbol{s}, \boldsymbol{g}, \sigma_t) = c_{\text{skip}}(\sigma_t)\boldsymbol{a} + c_{\text{out}}(\sigma_t)F_\theta(c_{\text{in}}(\sigma_t)\boldsymbol{a}, \boldsymbol{s}, \boldsymbol{g}, c_{\text{noise}}(\sigma_t)), \tag{7}$$

条件函数的详细说明见附录 A 节，并在图 1 中进行了可视化。

这些额外的跳跃连接有助于得分模型将输出缩放至广泛的噪声水平 σ_t，既可以通过估计去噪样本 $\boldsymbol{a}_{t-1}$，也可以直接预测噪声 ϵ，或预测介于两者之间的内容。我们提出的方法 BESO 将带有因果掩码的基于 Transformer 的架构用作内部模型 $F_\theta(\boldsymbol{a}, \boldsymbol{s}, \boldsymbol{g}, \sigma_t)$。这

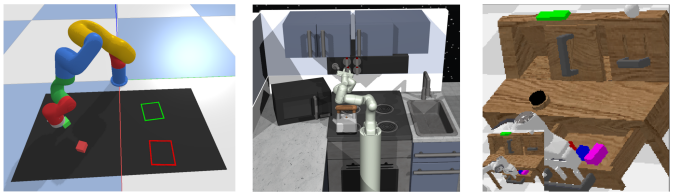


图 2. 用于测试 BESO 性能的仿真环境：多模态 Block-push（左）；Relay Kitchen（中）；CALVIN（右）

使我们的模型能够学习观测与动作之间的时间关系，从而提升整体性能。图 1 展示了我们所提架构的详细概览。三个线性嵌入层将状态 $\boldsymbol{s}_n$、噪声 σ_t 和带噪声的动作 $\boldsymbol{a}_n$ 编码为维度相同的线性表示，$l_{\boldsymbol{s}}(\boldsymbol{s})$ $l_{\boldsymbol{a}}(\boldsymbol{a})$ $l_\sigma(\sigma)$。此外，位置嵌入信息也会添加到这些线性表示中。噪声嵌入会与期望的未来状态以及所有状态-噪声-动作对拼接成一个较长的序列，供模型使用。在训练期间，模型会推断输入序列中所有时间步的去噪动作，但在推理时仅使用最后一个预测动作。为利用 Transformer 中的因果掩码，我们将目标序列拼接在当前观测序列之前 [6],，从而形成一个目标状态序列。

V. 评估

我们的实验旨在回答以下几个关键问题：**I)** BESO 在目标条件环境中的表现是否能与最先进的基线方法相媲美？**II)** 要实现性能良好的扩散策略快速采样，关键组成部分是什么？**III)** 无分类器引导是否适用于目标条件行为合成？为回答这些问题，我们在若干具有挑战性的仿真基准上评估了 BESO。首先，我们将 BESO 的性能与其他最先进方法进行了比较。随后，我们考察了 BESO 各组成部分对性能的贡献。

A. 基线方法

我们将 BESO 与多种最先进方法进行了比较：

- **目标条件行为克隆（GCBC）**学习一种单峰策略，该策略由简单的多层感知机（MLP）表示，并使用均方误差损失 [21]进行训练。
- **接力模仿学习（RIL）**是一种分层策略，它学习一个高层子目标生成器，用于对低层 MLP 策略进行条件化 [12]。
- **潜在运动计划（LMP）**是一种分层目标条件策略，由 Seq2Seq CVAE 和动作解码器策略组成 [21]。我们采用了一种改进的 KL 加权项和一个变换器编码器，研究表明，这些方法能够提升 LMP 的性能 [24]。
- **条件隐式行为克隆（C-IBC）**使用基于能量的模型作为隐式策略 [10]。我们

		GCBC	C-IBC	LMP	RIL	C-BeT	CX-Diff	C-BESO	CFG-BESO
Block-Push	奖励	0.13 (± 0.04)	0.46 (± 0.06)	0.04 (± 0.03)	0.06 (± 0.01)	0.91 (± 0.03)	0.93 (± 0.03)	***0.96*** (± 0.02)	***0.97*** (± 0.02)
	结果	0.13 (± 0.04)	0.29 (± 0.10)	0.04 (± 0.03)	0.02 (± 0.01)	0.87 (± 0.07)	0.90 (± 0.04)	***0.93***(± 0.02)	0.88 (± 0.04)
Relay-Kitchen	奖励	2.65 (± 0.25)	0.50 (± 0.09)	1.45 (± 0.22)	0.31 (± 0.15)	2.73 (± 0.28)	3.64 (± 0.14)	***3.98***(± 0.07)	***3.98*** (± 0.07)
	结果	2.57 (± 0.26)	0.45 (± 0.08)	1.41 (± 0.22)	0.23 (± 0.11)	2.69 (± 0.28)	3.35 (± 0.15)	***3.75***(± 0.08)	***3.47*** (± 0.08)

表 I：条件化推箱子任务和厨房环境上的均值与标准差，基于 10 个随机种子，每个随机种子运行 100 次。尽管仅使用 3 个推理步数，C-BESO 和 CFG-BESO 的表现始终优于所有基线方法。CX-Diff 使用 3 个推理步数时，在 Relay-Kitchen 上取得 2.74(±0.26) 的结果。BESO 的两种变体在不同随机种子下的偏差都很小，表明其具有较强的鲁棒性。

使用 IBC 的目标条件化扩展来研究所选生成模型架构的重要性。

- **Conditional-Behavior Transformer（C-BeT）** 是一种类似 GPT 的基于 Transformer 的策略，通过预测离散动作标签以及连续偏移向量来学习多模态行为 [35, 6]。动作标签通过 K-means 聚类预先确定。
- **Diffusion-X（CX-Diff）** [30] 是一种经过改进、推理效率更高的 DDPM [15] 策略。它采用随机采样，并在噪声水平最低时增加 X额外的推理步数，以便在 50+X 步内合成动作。尽管其表现略逊于密切相关的 KDE-Diff[30] ，但计算成本显著更低。

为确保对所有方法的公平评估，我们在调整方法特定超参数的同时，尽可能保持通用超参数（例如层大小和数量）的一致性。基线架构和超参数的详细摘要见附录C节。此外，我们在Kitchen和推箱子任务上对所有模型进行了评估，每个模型使用10个随机种子和100次滚动执行。鉴于训练CALVIN模型的高计算成本和时间，我们将测试方法限制为3个随机种子，并限制了基线数量。

B. 仿真实验

我们在三个仿真基准上评估了BESO与基线方法，如图2所示：

- **CALVIN基准 [25]:** 我们使用了LfP基准，其数据集由6小时非结构化游戏数据组成。我们将所有方法限制为使用单个静态RGB图像作为观察输入，并预测相对笛卡尔动作作为输出 [34]。我们从一个目标图像评估了方法在单个任务和连续2个任务上的表现，两种变体均以训练分布之外的目标图像为条件，这些图像中末端执行器不在正确位置。
- **Block-Push环境 [10]:** 我们使用了经过调整的目标条件变体 [6]。Block-Push环境由一台XArm机器人组成，它必须将两个方块（一个红色、一个绿色）推入红色和绿色的方形目标区域。数据集由确定性控制器采集的1000条演示组成，包含4种可能的目标配置。每个方块被推入任一目标区域，方法即可获得0.5分，最高得分为1.0分。
- **Relay Kitchen环境 [12]:** 这是一个多任务厨房环境，包含水壶、门和灯等物体，智能体可以与之交互。数据由566条人工采集轨迹组成，每条轨迹包含按顺序执行的4项技能。我们采用了 [6] 中描述的相同实验设置，以便进行公平比较。模型使用预先定义的目标状态进行评估，每次试验包含4项任务。每项正确完成的任务得1分，最高得4分。

这些方法根据两个指标进行评估：**结果** 用于评估每次滚动过程中达成了多少个期望目标，而**奖励**则通过对达成环境中定义的任意目标给予分数，来衡量整体表现。

C. 仿真结果

我们在 Relay-Kitchen 和 Block-Push 环境中将 BESO 与基线方法进行了比较。结果汇总于表 I。如表所示，在两个任务中，BESO 在 10 个随机种子下始终优于其他方法。此外，BESO 的低方差也表明我们的方法具有较强的鲁棒性。在各基线方法中，Diffusion-X 在厨房任务上表现出色，而 C-BeT 在推箱子环境中表现突出。扩散策略表现优异，在厨房和推箱子任务中均超过了所有其他基线方法，而 C-BeT 在推箱子环境中的表现与之相当。考虑到 BESO 在两个环境中都只使用了 3 个去噪步数，而 CX-Diff 使用了 50(20) + 8 步，BESO 的表现就更加出色。相比之下，CX-Diff 在去噪步数限制为 3 步时，在厨房环境中的平均结果仅为 2.74(±0.26)。这凸显了 BESO 的架构结合改进的噪声调度和采样器，仅用 3 个去噪步数便能取得良好结果的优势。在现代台式电脑上，BESO 预测一个动作约需 0.012 秒，而 CX-Diffusion 模型平均需要 0.15 秒。这使 BESO 的速度提升了 10 倍以上。

在更具挑战性的仿真环境 CALVIN 环境中，BESO 展现了泛化至未见过的目标状态的能力，在 13 个困难的单项任务上取得了最佳整体性能。每项任务都以训练期间未见过的单张目标图像作为条件，并且末端执行器并未位于相应任务的附近。这带来了巨大挑战，因为模型必须推断环境状态的变化，并执行所需的

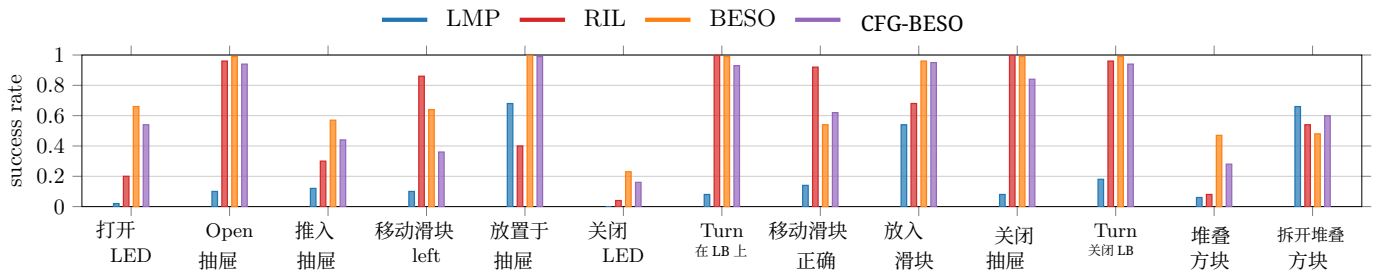


图 3. 所有测试模型在 CALVIN 环境中以单张目标图像为条件执行单个高难度任务的平均成功率，该目标图像不包含靠近所需任务的机器人末端执行器。

任务，而不能依赖图像中末端执行器的位置来获得指导。该实验的结果总结于图 4 中，各项任务的具体成功率总结于图 3 中。如图所示，BESO 在各项高难度任务上均取得了最佳整体性能，展现了其同样泛化至未见过的目标状态的能力。RIL 是表现第二好的模型，并且在 2 项任务上的平均性能略高。

此外，这些模型还在以单张目标图像解决两项任务上进行了评估。与第一项任务类似，末端执行器位于远离两项任务的不同位置。在此情况下，BESO 及其无分类器引导（CFG）变体再次优于其他模型，尽管 CFG 变体的性能略低。结果表明，BESO 能够通过从交互轨迹的随机窗口中学习，有效学习有意义的行为以解决下游短期和长期目标。这进一步支持了 BESO 学习多模态和富有表现力的动作分布的能力是从游戏中有效学习的关键这一结论。此外，该实验展示了 BESO 从视觉数据中有效学习的能力。总体而言，我们的结果表明 BESO 与最先进的基线具有竞争力，并且能够有效地从游戏数据中学习，使其成为目标条件行为学习的一种有前景的方法。因此，我们可以肯定地回答问题 **I**）。

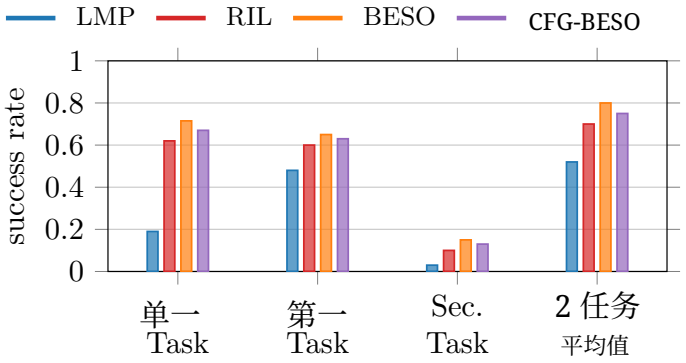


图 4. 目标条件策略在 CALVIN 环境中的平均表现。第一列显示 13 项单项任务的平均成功率。其他三列显示所有模型在以包含 2 项任务的单个目标图像为条件时的平均成功率。

D. BESO 设计选择

我们通过评估 BESO 的不同组成部分来回答问题**II**，以研究它们对整体性能的贡献。

条件化方法。 首先，我们评估了不同的方法，以根据期望的目标状态对行为生成进行条件化。我们测试了 FiLM 条件化 [31]，以及 C-BeT [6] 中使用的顺序条件化方法。FiLM 需要额外的多层感知机模型，这些模型将目标作为输入，并对 Transformer 层中的潜在表示进行缩放。顺序条件化方法只需将期望的目标状态纳入序列开头，如图 1 的模型概览所示。我们使用

使用相同的Transformer评分模型，并在方块-推送和Kitchen环境上以10个随机种子进行了评估。FiLM条件化

这导致与顺序条件化方法相比，Block-Push 和厨房环境的平均结果分别从 0.93 降至0.91，以及从 3.76降至 3.4。此外，FiLM 方法增加了整体模型容量。因此，BESO 使用顺序条件化方法。

采样算法。 BESO 通过从其高斯先验分布 p_T 生成的样本出发，使用其学习的分数模型对反向常微分方程进行数值近似来生成动作。我们研究了扩散研究中使用的几种数值采样算法，如 DDIM[38]，DPM [19]，DPM++ [20] 和 Heun [17]，，以评估它们对 BESO 性能的贡献。这些采样器在 Block-Push 和厨房环境中以不同数量的去噪步数进行了评估。结果表明，各个采样器之间的性能差距很小，其中 DDIM 实现了最佳的整体性能。令人惊讶的是，二阶 Heun求解器的平均性能最差。该实验的详细结果总结在附录中的表 VII 和表 VIII 中。总体而言，BESO 对采样步数和所选采样器类型具有鲁棒性，从 3 到 50 个推理步数保持相似的性能。

		确定性的	随机性
方块-Push	奖励	0.97 (± 0.02)	0.97 (± 0.02)
	结果	0.93 (± 0.02)	0.92 (± 0.03)
中继-Kitchen	奖励	3.95 (± 0.10)	4.03 (± 0.07)
	结果	3.73 (± 0.11)	3.80 (± 0.08)
CALVIN	高难度任务	0.71 (± 0.01)	0.68 (± 0.03)
	2 任务	0.79 (± 0.03)	0.79（± 0.02）

表 II 随机噪声注入与未注入时，噪声注入对目标条件行为生成的影响评估，结果为两个采样器的平均值。

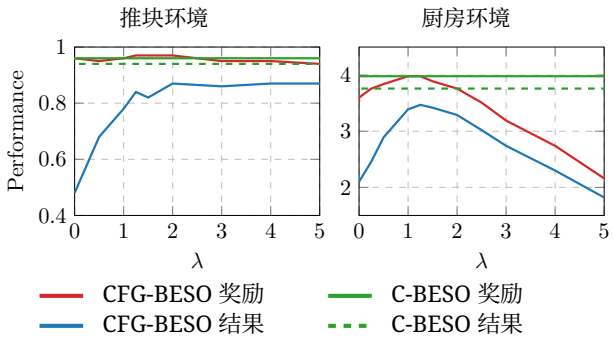


图 5. 使用 CFG 方法进行基于游戏数据的目标条件行为学习比较。对于 CFG-BESO，我们使用不同的 λ 值，对 10 个随机种子分别进行 100 次滚动执行评估。BESO 的 CFG 变体在两个环境中的平均结果略差，但奖励相近。使用 $\lambda = 0$ 时，我们可以恢复出一个无条件策略，该策略执行随机滚动时奖励较高、结果较低。

随机采样与确定性采样。当前扩散模型文献支持随机采样器相比确定性采样器具有更好整体性能的假设 [17, 41]。我们针对基于步进的动作生成测试了这一假设。我们使用 2 种采样算法 DPM++(2S) 和欧拉采样器 [19, 17]，评估了相同模型，每种算法分别带和不带噪声注入。噪声调度通过祖先采样策略执行，如 DPPM 变体 [15, 41] 中所用并在算法 C 中描述。实验再次在所有环境中进行。如表 II 所示，结果表明添加噪声对基于步进的扩散策略的动作生成并无显著益处。随机采样器仅在厨房环境中提高了平均性能。与图像合成 [7] 等常见扩散应用相比的差异，可能源于高维图像空间，使得生成过程更加困难并需要更多步数才能获得良好结果。在这些高维空间中，错误更可能发生并随时间累积。在推理过程中添加噪声有助于模型纠正梯度近似的错误，从而获得更好的整体性能 [17]。相比之下，基于步进的动作分布的维度显著低于图像生成的高维潜空间，因此添加噪声似乎并不利于基于步进的策略的平均性能，这得到了我们实验结果的支持。

无分类器引导（CFG）。最后，我们通过评估无分类器引导（CFG）对采用目标条件策略的基于步进的动作生成的影响，来探讨问题**III** 。图 5 所示的实验结果表明，在基于步进的设置中，CFG 是一种有效的目标条件化方法。平均结果为
方块推送和厨房任务的表现略逊于标准目标条件变体，但平均奖励相同。CFG-BESO 也能在基于图像的 CALVIN 环境中有效学习，并取得与标准目标条件变体相近的表现。CFG 模型在 $\lambda = 0$ 下的表现表明，CFG-BESO 能够学习出表现良好的无条件策略 $\pi(\boldsymbol{a}|\boldsymbol{s})$。图 5 中较低的平均结果表明，该策略忽略目标状态，仅依据当前状态来获取高奖励。这使 CFG-BESO 相较于常见的游戏型策略具有独特优势。然而，CFG 存在权衡：它会略微降低更多样化滚动执行的平均结果。实证评估表明，对于大多数经过测试的环境，最佳 λ 值为 1.25。在动作空间维度较高的环境中，使用更高的值进行实验会导致平均表现下降，表明动作生成不稳定。我们推测，目标条件化提供的引导仅在滚动执行的某些步骤中至关重要，具体而言，是策略决定要解决哪项任务时。

VI. 结论

我们提出了 BESO，这是一种新的目标条件行为生成策略表示，采用基于分数的扩散模型。我们利用基于分数的扩散模型的表达能力和多模态特性，从离线、无奖励的游戏数据集中学习与任务无关的行为，无需层次结构或额外的聚类。此外，我们展示了无分类器引导在序列设置中同时学习目标相关策略和目标无关策略的有效性。在多个 GCIL 基准上的实验表明，BESO 显著优于多种最先进的 GCIL 算法。消融研究验证了 BESO 中能够实现快速、确定性行为生成的关键组件。仅使用 3 个去噪步数时，它的表现也优于标准 DDPM 策略，缓解了扩散采样速度慢这一既有缺点。

尽管 BESO 作为独立策略表现出色，但它也具有灵活性，可作为动作预测策略无缝集成到其他层次化框架中。作为传统行为克隆策略的一种实用替代方案，BESO 凭借扩散模型固有的独特特性脱颖而出。未来，我们计划将 BESO 扩展用于语言引导的行为生成，为人类提供更直观的目标引导。

VII. 致谢

本文所述工作由德国研究基金会（DFG）资助，项目编号为 448648559。

参考文献

[1] Anurag Ajay、Yilun Du、Abhi Gupta、Joshua B. Tenen- baum、Tommi S. Jaakkola 和 Pulkit Agrawal。条件生成建模是否就是决策所需的一切？载于国际学习表征会议，2023年。网址：https://openreview.net/forum? id=sP1fo2K9DFG.[2] Brenna D Argall、Sonia Chernova、Manuela Veloso 和 Brett Browning。机器人示范学习综述。《机器人与自主系统》，57（5）：469–483，2009年。[3] Suneel Belkhale 和 Dorsa Sadigh。PLATO：通过以物体为中心的游戏预测潜在可供性。载于第六届机器人学习年会，2022年。网址：https://openreview.net/forum?id=UAA5bNospA0。[4]Ricky TQ Chen、Yulia Rubanova、Jesse Bettencourt 和 David K Duvenaud。神经常微分方程。神经信息处理系统进展，31，2018年。[5] Cheng Chi、Siyuan Feng、Yilun Du、Zhenjia Xu、Eric Cousineau、Benjamin Burchfiel 和 Shuran Song。扩散策略：通过动作扩散实现视觉运动策略学习。载于机器人学：科学与系统（*RSS*）论文集，2023年。[6]Zichen Jeff Cui、Yibin Wang、Nur Muhammad Mahi Shafiullah 和 Lerrel Pinto。从游戏到策略：基于未经筛选的机器人数据进行条件行为生成。载于国际学习表征会议，2023年。网址：https://openreview.net/forum?id=c7rM7F7jQjN。[7] Prafulla Dhariwal 和 Alexander Nichol。扩散模型在图像合成方面胜过生成对抗网络。神经信息处理系统进展，34：8780–8794，2021年。[8] Benjamin Eysenbach、Soumith Udatha、Ruslan Salakhut- dinov 和 Sergey Levine。模仿过去的成功可能非常次优。载于神经信息处理系统进展，2022年。网址：https://openreview.net/forum?id=iqCO3jbPjYF。[9] Benjamin Eysenbach、Tianjun Zhang、Sergey Levine 和 Ruslan Salakhutdinov。将对比学习作为目标条件强化学习。载于神经信息处理系统进展，2022年。网址：https://openreview.net/forum?id=vGQiU5sqUe3。[10] Pete Florence、Corey Lynch、Andy Zeng、Oscar A Ramirez、Ayzaan Wahid、Laura Downs、Adrian Wong、Johnny Lee、Igor Mordatch 和 Jonathan Tompson。隐式行为克隆。载于机器人学习会议，第158–168页。PMLR，2022年。[11]Justin Fu、Katie Luo 和 Sergey Levine。通过对抗式逆强化学习学习鲁棒奖励

. 在国际学习表征会议

，*2018*年。URL https://openreview.net/forum? id=rkHywl-A-.[12] Abhishek Gupta、Vikash Kumar、Corey Lynch、Sergey Levine 和 Karol Hausman。接力策略学习：通过模仿学习和强化学习解决长时程任务。机器人学习会议（*CoRL*），2019年。[13] Jonathan Ho 和 Stefano Ermon。生成对抗模仿学习。神经信息处理系统进展，29，2016年。[14] Jonathan Ho 和 Tim Salimans。无分类器扩散引导。载于*NeurIPS 2021*年深度生成模型与下游应用研讨会，2021年。[15] Jonathan Ho、Ajay Jain 和 Pieter Abbeel。去噪扩散概率模型。神经信息处理系统进展，33：6840–6851，2020年。[16]Michael Janner、Yilun Du、Joshua Tenenbaum 和 Sergey Levine。使用扩散进行规划，以实现灵活的行为合成。载于国际机器学习会议，2022年。[17] Tero Karras、Miika Aittala、Timo Aila 和 Samuli Laine。阐明基于扩散的生成模型的设计空间。载于 Alice H. Oh 、 Alekh Agarwal 、 Danielle Belgrave 和 Kyunghyun Cho 编，神经信息处理系统进展，2022年。URL https://openreview.net/forum?id=k7FuTOWMOc7。[18] Kuno Kim、Akshat Jindal、Yang Song、Jiaming Song、Yanan Sui 和 Stefano Ermon。使用神经密度模型进行模仿。神经信息处理系统进展，34：5360–5372，2021年。[19] Cheng Lu、Yuhao Zhou、Fan Bao、Jianfei Chen、Chongxuan Li 和 Jun Zhu。Dpm-solver++：用于扩散概率模型引导采样的快速求解器。*arXiv* 预印本 *arXiv:2211.01095*，2022年。[20] Cheng Lu、Yuhao Zhou、Fan Bao、Jianfei Chen、Chongxuan Li 和 Jun Zhu。DPM-solver：约 10 步内完成扩散概率模型采样的快速常微分方程求解器。载于神经信息处理系统进展，2022年。URL https://openreview.net/forum?id=2uAaGwlP V。[21] Corey Lynch、Mohi Khansari、Ted Xiao、Vikash Kumar、Jonathan Tompson、Sergey Levine 和 Pierre Sermanet。从游戏中学习潜在计划。载于机器人学习会议，第 1113–1132 页。PMLR，2020年。[22] Yecheng Jason Ma、Jason Yan、Dinesh Jayaraman 和 Osbert Bastani。通过 f-优势回归进行离线目标条件强化学习。载于第三十六届神经信息处理系统会议，2022年。URL https://openreview.net/forum?id= h29VprPHD。[23] Ajay Mandlekar、Danfei Xu、Roberto Mart´ın-Mart´ın、Silvio Savarese 和 Li Fei-Fei。GTI：从人类示范中学习跨长时程任务的泛化能力。载于机器人学：科学与系统论文集，2020年7月。doi：10.15607/RSS.2020.XVI.061。[24] Oier Mees、Lukas Hermann 和 Wolfram Burgard。在非结构化数据上进行语言条件机器人模仿学习时，什么因素最重要

. *IEEE*机器人与

自动化快报 *(RA-L)*, 7(4):11205–11212, 2022年.[25] Oier Mees, Lukas Hermann, Erick Rosete-Beas 和 Wolfram Burgard。Calvin：用于长时程机器人操作任务中语言条件策略学习的基准。*IEEE*机器人与自动化快报, 2022年.[26] Lina Mezghani, Sainbayar Sukhbaatar, Piotr Bojanowski, Alessandro Lazaric 和 Karteek Alahari。通过自监督奖励塑形离线学习目标条件策略。载于 *CoRL*-机器人学习会议, 2022年.[27]Alexander Quinn Nichol 和 Prafulla Dhariwal。改进的去噪扩散概率模型。载于国际机器学习会议, 第 8162–8171 页。PMLR, 2021年.[28]Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever 和 Mark Chen。GLIDE：利用文本引导扩散模型实现逼真图像生成与编辑。载于 Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu 和 Sivan Sabato（编），第*39*届国际机器学习会议论文集，第 162 卷，机器学习研究论文集，第 16784–16804 页。PMLR, 2022年7月17–23日.[29] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters 等。模仿学习的算法视角。机器人学基础与趋势®, 7(1-2):1–179, 2018年.[30]Tim Pearce, Tabish Rashid, Anssi Kanervisto, Dave Bignell, Mingfei Sun, Raluca Georgescu, Sergio Valcarcel Macua, Shan Zheng Tan, Ida Momennejad, Katja Hofmann 和 Sam Devlin。利用扩散模型模仿人类行为。载于国际学习表征会议, 2023年。URL https: // openreview.net/forum?id=Pv1GPQzRrC8.[31]Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin 和 Aaron Courville。FiLM：利用通用条件层进行视觉推理。载于*AAAI*人工智能会议论文集，第 32 卷, 2018年.[32] Karl Pertsch, Youngwoon Lee 和 Joseph Lim。利用学习到的技能先验加速强化学习。载于机器人学习会议，第 188–204 页。PMLR, 2021年.[33] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser 和 Bj¨orn Ommer。利用潜在扩散模型进行高分辨率图像合成。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第 10684–10695 页，2022年.[34] Erick Rosete-Beas, Oier Mees, Gabriel Kalweit, Joschka Boedecker 和 Wolfram Burgard。面向任务无关离线强化学习的潜在计划。载于第六届机器人学习年会, 2022年。URL https://openreview.net/forum?id=ViYLaruFwN3.[35] Nur Muhammad Mahi Shafiullah, Zichen Jeff Cui, Ariuntuya Altanzaya 和 Lerrel Pinto。行为变换器：用一块石头克隆 k 模式。In 第三十六届神经信息处理系统会议，2022年。URL https://openreview.net/forum?id=agTr-vRQsa.[36] Avi Singh、Huihan Liu、Gaoyue Zhou、Albert Yu、Nicholas Rhinehart 和 Sergey Levine。Parrot：用于强化学习的数据驱动行为先验。In国际学习表征会议, 2020年。[37]Jascha Sohl-Dickstein、Eric Weiss、Niru Maheswaranathan 和 Surya Ganguli。使用非平衡热力学的深度无监督学习。In 国际机器学习会议，第 2256–2265页。PMLR，2015年。[38] Jiaming Song、Chenlin Meng 和 Stefano Ermon。去噪扩散隐式模型。In*ICLR*，2021年。[39] Yang Song 和 Stefano Ermon。通过估计数据分布的梯度进行生成建模。神经信息处理系统进展，32，2019年。[40] Yang Song 和 Stefano Ermon。训练基于分数的生成模型的改进技术。神经信息处理系统进展，33:12438–12448，2020年。[41] Yang Song、Jascha Sohl-Dickstein、Diederik P Kingma、Abhishek Kumar、Stefano Ermon 和 Ben Poole。通过随机微分方程的基于分数的生成建模。In 国际学习表征会议，2020年。[42]Guy Tevet、Sigal Raab、Brian Gordon、Yoni Shafir、Amit Haim Bermano 和 Daniel Cohen-or。人体运动扩散模型。In 国际学习表征会议，2023年。URL https: // openreview.net/forum?id=SJ1kSyO2jwu.[43] Julen Urain、Niklas Funk、Jan Peters 和 Georgia Chalvatzaki。Se(3)-diffusionfields：通过扩散学习关节抓取和运动优化的平滑代价函数。*IEEE*国际机器人与自动化会议 *(ICRA)*，2023年。[44] Pascal Vincent。分数匹配与去噪自编码器之间的联系。神经计算，23(7): 1661–1674，2011年。doi: 10.1162/NECO a 00142.[45] Zhendong Wang、Jonathan J Hunt 和 Mingyuan Zhou。扩散策略作为离线强化学习的一种表达性策略类别。In 国际学习表征会议，2023年。URL https: // openreview.net/forum?id=AHvFDPi-FA.[46]Rui Yang、Yiming Lu、Wenzhe Li、Hao Sun、Meng Fang、Yali Du、Xiu Li、Lei Han 和 Chongjie Zhang。重新思考目标条件监督学习及其与离线强化学习的联系。In 国际学习表征会议，2021年。[47] Sarah Young、Jyothish Pari、Pieter Abbeel 和 Lerrel Pinto。用于表征学习的游戏互动。In*2022*年 *IEEE/RSJ*国际智能机器人与系统会议 *(IROS)*，第992–999页，2022年。doi: 10.1109/IROS47612.2022.9981307。

超参数	Block-Push	Relay-Kitchen	CALVIN
隐藏维度	240	360	768
隐藏层	4	6	6
窗口大小	5	5	2
目标窗口大小	1	2	1
注意力头数	12	6	4
注意力丢弃率	0.05	0.3	0.2
残差丢弃率	0.05	0	0.1
学习率	1e-4	1e-4	1e-4
优化器	Adam	Adam	AdamW
去噪步数	3	3	5
$\sigma_{\max}$	1	1	1
$\sigma_{\min}$	0.05	0.005	0.005
σ_{data}	0.5	0.5	0.5
最佳 λ 用于 CFG	2	1.25	1.25
噪声分布	对数逻辑斯蒂分布	对数逻辑斯蒂分布	对数逻辑斯蒂分布
EMA	True	True	True
视觉编码器	None	None	ResNet-18
采样器类型	DDIM	DDIM	DDIM
噪声调度器	Exp	Exp	Exp
批次大小	1024	1024	64
训练步数（千）	60	40	120

表 III 所有测试环境中不同模型架构最重要超参数概览。

	Block-Push		Relay Kitchen	
	奖励	结果	奖励结果	
指数分布	0.96 (± 0.02)	0.93 (± 0.02)	3.86 (± 0.09)	3.74 (± 0.09)
线性分布	0.97 (± 0.01)	0.94（± 0.01）	3.98 (± 0.08)	3.76 (± 0.10)
IDDPM [27]	0.95 (± 0.01)	0.92 (± 0.01)	3.88 (± 0.08)	3.67 (± 0.09)
Karras [17]	0.96 (± 0.02)	0.93 (± 0.03)	3.96 (± 0.08)	3.75 (± 0.07)
VE [41]	0.95 (± 0.01)	0.93 (± 0.01)	3.98 (± 0.09)	3.75 (± 0.10)
VP [41]	0.64 (± 0.17)	0.61 (± 0.16)	3.21 (± 0.12)	2.95 (± 0.14)

表 IV

目标条件行为生成的时间步函数影响评估：对 10 个随机种子进行平均，每个随机种子执行 100 次滚动执行。所有模型均使用 DDIM 求解器，并采用 3 个去噪步进行比较。

附录

A. *BESO* 超参数

BESO的关键超参数总结列于表III。我们观察到，Transformer特定的超参数如丢弃率需要根据任务进行调整，而一般的扩散超参数在不同任务中保持一致。

预条件。我们使用 Karras 等人 [17] 提出的预条件函数：

- $c_{\text{跳过}} = \sigma^2_{\text{数据}} / \left(\sigma^2_{\text{数据}} + \sigma_t^2\right)$
- $c_{\text{输出}} = \sigma_t \sigma_{\text{数据}} / \sqrt{\sigma^2_{\text{数据}} + \sigma_t^2}$
- $c_{\text{输入}} = 1 / \sqrt{\sigma^2_{\text{数据}} + \sigma_t^2}$
- $c_{\text{噪声}} = 0.25 \ln(\sigma_t)$

归一化。 BESO 在动作的扩散范围为 $[-1, 1]$ 、噪声范围为 0.005、1 时表现最佳。我们在所有三种环境中都采用了这一噪声范围，并相应地缩放动作输出。对于具有更大扩散范围的动作，例如 $[-3, 3]$，建议将噪声范围扩大到更高的值：$\{0.4, 40\}$ ，以获得最佳性能。对于输入，我们建议将数据归一化为均值为 0、标准差为 1。

训练噪声分布。在训练期间，噪声值从预定义的噪声分布 $P(\sigma)$ 中采样。扩散文献 [17] 中使用的标准分布是对数正态分布，它引入了两个额外的超参数 σ_{std}， $\sigma_{\max}$，需要进行额外调优。我们的实验表明，先前工作 [17] 推荐的值并不适用于动作扩散。因此，我们选择了对数逻辑斯蒂分布 LogLogistic($\alpha = 0.5$, $\beta = 0.5$)，该分布不需要额外参数，并且在我们的所有实验中都表现良好。

优化。 在优化方面，我们在实验中采用了常用的 Adam 或 AdamW 优化器，标准学习率为 $1e{-}4$。此外，我们使用指数移动平均（EMA）来优化模型的权重。

时间步。 一个重要的选择是时间步函数，它决定噪声水平如何分布在离散步上。表 IV 中总结的实证评估表明，指数时间步对 BESO 的平均效果最佳。不过，线性调度器 [41] 和 Karras 调度器[17] 等其他离散化方法也能取得相近的结果，并可提升单项任务的性能。

建议。 对于新任务，我们建议从噪声范围 $\{0.005, 1\}$ 开始，并搭配指数时间步和 DDIM 求解器。若要获得最佳性能，值得尝试其他采样器，例如 Euler Ancestral，以及线性时间步。

B. 采样器消融

我们在不同环境中评估了多种最先进的常微分方程采样器及其随机微分方程对应方法。为了确定最适合条件行为生成的求解器，我们分析了不同环境中 10 个随机种子的平均表现，每个种子进行 100 次滚动执行。总体而言，我们将求解器分为一阶和二阶两类：一阶求解器为 Euler [15] ，测试的二阶求解器为 Heun [17]。测试的采样器包括：

- **Euler 常微分方程（Euler）：** 一种一阶常微分方程采样器，来自[17] ，不包含额外的噪声添加和去除。算法总结见 C。
- **Euler-Ancestral（EA）：** 标准 DDPM 采样器的连续时间版本 [15]，由 [41]提出。
- **2二阶 Heun 求解器 (Heun)：** 一种使用 Heun 方法的二阶常微分方程求解器 [17]。
- **DPM：** 一种专为在少量推理步数中进行合成而设计的指数型常微分方程积分器求解器 [20]。我们使用二阶方法。
- **DDIM：** DPM 的一阶变体，已单独引入 [38, 20] ，并专为快速推理和 CFG 而设计。
- **DPM-Ancestral：** DPM 的随机变体，带有祖先噪声注入。
- **DPM++(2S):** 二阶的改进版本

用于基于无分类器引导的条件扩散模型的DPM采样器，仅需单次推理步骤 [19]

- **DPM++(2M)：**一种改进版的二阶 DPM 采样器，适用于基于无分类器引导的条件扩散模型 [19], ，它是一种每步使用两次模型预测的二阶方法。

此前有多项研究在图像生成背景下比较了常微微分方程采样器的性能 [17, 19]。然而，这些比较未必完全具有参考意义，因为图像生成任务面临着动作合成所不涉及的独特挑战和要求。为确保公平比较，我们在多个仿真环境中使用相同的模型评估所有采样器，并根据每个环境下的 100 次运行报告其平均性能。这样，我们就能准确比较各确定性求解器在基于步进的动作生成中的效果。厨房环境的结果见表 VII，Block Push 的性能见表 VIII。如两表所示，一阶指数积分求解器 DDIM 的总体表现最佳。增加推理步数对平均性能没有显著影响，甚至会降低部分采样器的平均结果。总体而言，所有受评估采样器之间的性能差异都很小。

超参数	Block Push	Relay Kitchen
隐藏维度	128	128
隐藏层	6	6
训练步数	5000	1000
噪声尺度	0.3	0.3
Loss	InfoNCE	InfoNCE
训练样本	64	64
噪声收缩	0.5	0.5
学习率	0.001	0.001

表 V GC-IBC 在两种环境中所使用的超参数概览。

超参数	Block-Push	Relay-Kitchen	CALVIN
解码器隐藏维度	{128, 256, 512, **1024**}	{128, 256, 512}	2048
混合分量数	10	10	10
类别数	{10, 32, 64, 128, **256**}	10	10
策略丢弃率	{0.1,**0.2**, 0.3}	{0.1,**0.2**, 0.3}	0.1
规划特征	{16, **32**, 64, 128}	{16, 32, **64**, 128}	32
计划识别特征	{64, 128, **256**, 512}	{64, 128, 256, **512**}	2048
重新规划频率	{5, **10**, 16, 32}	{5, 10, 16, **32** }	2
规划器隐藏层	2	2	2
窗口大小	{10, 16, 32, **48**}	{10, **16**, 32, 48, 64}	16
目标窗口大小	1	1	1
KL-β	{0.001,0.005,**0.01**}	{0.001, 0.005, **0.01**}	0.01
学习率	{0.001,**0.0005**, 0.0001}	{0.001, **0.0005**, 0.0001}	0.0001
优化器	Adam	Adam	Adam

表 VI 潜在计划超参数扫描概述及各测试仿真环境评估所用最终参数

C. 基线方法实现

基于多层感知机的模型包含 4 层，每层有 512 个神经元，并使用 ReLU 激活函数。所有扩散模型都采用相同的 Transformer 主干网络，C-BeT 则使用其推荐的参数。训练期间，MLP 模型使用 Adam 优化器，学习率为 0.001；Transformer 模型使用 $1e-4$。MLP 模型的批次大小为 512，Transformer 模型为 1024；BeT 例外，其批次大小为 64，与 [6] 中的推荐一致。

GCBC 对于 GCBC 模型，目标与状态拼接后输入 4 层多层感知机架构，丢弃率为 0.1。

GC-IBC GC-IBC 模型采用与 GCBC 相同的多层感知机架构，并使用 InfoNCE 损失以及额外的能量正则化和 Wasserstein 梯度损失进行优化。在实验中，添加 $\lambda = 0.005$ 的惩罚项以限制平均能量，提高了训练稳定性[10]。鉴于 IBC 有大量可调超参数，我们进行了超参数搜索以确定最佳参数。需要指出的是，EBM 模型的结果对初始随机种子非常敏感，我们很难让模型获得一致的结果。相关研究也报告了 IBC 性能方面的类似观察 [30, 5]。

C-BeT 对于 C-BeT的性能，我们对所有测试环境使用 Cui 等人推荐的参数。 [6] 我们报告的结果略差于原始工作中报告的结果，因为原始工作没有对 10 个随机种子进行平均。

潜在运动计划 LMP 模型在 Kitchen 和 Block Push 环境中进行了评估，并通过大规模超参数搜索找到性能最佳的配置。扫描参数及所选参数的详细概述如表 VI 所示。在 CALVIN 环境中，使用了先前工作提出的参数 [34]。我们使用了来自 [24], 的改进 LMP 变体，称为 HULC，它使用不同的 KL 散度加权项以及 Transformer 模型 Seq2Seq CVAE。

RIL 对于 Kitchen 和 Block Push 的低层策略，我们使用 4 层，每层 512 个神经元。对于 CALVIN 任务，我们使用来自 [34] 的基线版本，并在训练中保持超参数相同。

Diffusion-X 来自 [30] 的基线使用与我们在 III 中报告的 Transformer 模型相同的超参数，以保证公平比较。Diffusion-X 在 Kitchen 任务上使用 50 个推理步数，并在最低噪声水平额外进行 10 个微调步数，而我们在 Block-Push 环境中使用 20 个推理步数，并额外进行 8 个微调步数。Diffusion-X 使用带有祖先噪声调度器的离散欧拉采样方法变体，该方法报告于算法 C [15, 41]。此外，它在最低噪声水平应用 X 额外去噪步数。

算法 3 祖先噪声调度器 f_{ANC} [41, 15]

要求： t_{from}, $t_{\text{to}} \leftarrow$ 最小值(t_{to}, $\sqrt{t_{\text{to}}^2 - t_{\text{from}}^2}$)

$t_{\text{up}} \leftarrow \sqrt{t_{\text{to}}^2 - t_{\text{from}}^2}$

$t_{\text{down}} \leftarrow \sqrt{(t_{\text{to}}^2 - t_{\text{from}}^2)}$

return t_{down}, t_{up}

	步数	Euler	Heun	DDIM	DPM	DPM++(2S)	DPM++(2M)
奖励	3	3.87 (± 0.09)	3.80 (± 0.03)	3.92 (± 0.07)	3.86 (± 0.08)	3.88 (± 0.08)	3.89 (± 0.08)
	5	3.87 (± 0.07)	3.84（± 0.06）	3.88 (± 0.06)	3.90 (± 0.09)	3.87（± 0.06）	3.87 (± 0.06)
	10	3.85 (± 0.08)	3.86 (± 0.09)	3.88 (± 0.06)	3.87 (± 0.10)	3.88 (± 0.07)	3.89 (± 0.05)
	20	3.86 (± 0.10)	3.91 (± 0.08)	3.87 (± 0.07)	3.88 (± 0.09)	3.89 (± 0.07)	3.88 (± 0.06)
	50	3.82 (± 0.08)	3.93 (± 0.04)	3.88 (± 0.06)	3.82 (± 0.10)	3.67 (± 0.04)	3.89 (± 0.06)
结果	3	3.66 (± 0.09)	3.62 (± 0.07)	3.69 (± 0.07)	3.67 (± 0.10)	3.67 (± 0.09)	3.67 (± 0.08)
	5	3.66 (± 0.07)	3.66 (± 0.06)	3.67 (± 0.08)	3.67 (± 0.08)	3.66 (± 0.07)	3.66（± 0.08）
	10	3.65 (± 0.06)	3.63 (± 0.06)	3.67 (± 0.07)	3.66 (± 0.07)	3.66 (± 0.08)	3.67 (± 0.07)
	20	3.64 (± 0.07)	3.65 (± 0.09)	3.66 (± 0.09)	3.66 (± 0.07)	3.68 (± 0.09)	3.67 (± 0.08)
	50	3.62 (± 0.04)	3.67 (± 0.04)	3.67 (± 0.09)	3.62 (± 0.07)	3.67 (± 0.07)	3.67 (± 0.08)

表VII 在厨房环境中对确定性采样器的性能比较，每个随机种子进行100次滚动执行，并对 10 个随机种子取平均值。我们测试了在对数正态噪声分布上训练的 BESO，使用 $\sigma_{\text{MEAN}} = -2$、$\sigma_{\text{STD}} = -2$、$\sigma_{\text{MAX}} = 33$、$\sigma_{\text{MIN}} = 0.39$，并采用指数时间步。

	步数	Euler	Heun	DDIM	DPM	DPM++(2S)	DPM++(2M)
奖励	3	0.95 (± 0.02)	0.92 (± 0.02)	0.96 (± 0.02)	0.96 (± 0.02)	0.95 (± 0.03)	0.97 (± 0.02)
	5	0.94 (± 0.04)	0.95 (± 0.02)	0.96 (± 0.02)	0.97 (± 0.01)	0.94 (± 0.02)	0.93 (± 0.02)
	10	0.97 (± 0.03)	0.93 (± 0.02)	0.96 (± 0.01)	0.95（± 0.03）	0.96 (± 0.02)	0.96 (± 0.03)
	20	0.98（± 0.02）	0.96 (± 0.03)	0.98 (± 0.02)	0.96（± 0.03）	0.96 (± 0.02)	0.97 (± 0.03)
	50	0.98 (± 0.01)	0.96 (± 0.01)	0.97 (± 0.02)	0.97 (± 0.05)	0.97 (± 0.01)	0.94 (± 0.05)
结果	3	0.94 (± 0.02)	0.90 (± 0.05)	0.94 (± 0.04)	0.94（± 0.01）	0.92 (± 0.03)	0.94 (± 0.03)
	5	0.91 (± 0.06)	0.93 (± 0.03)	0.95 (± 0.02)	0.95 (± 0.02)	0.91 (± 0.03)	0.93 (± 0.03)
	10	0.94 (± 0.02)	0.91 (± 0.04)	0.95 (± 0.02)	0.91 (± 0.04)	0.94 (± 0.02)	0.96 (± 0.02)
	20	0.96 (± 0.02)	0.94 (± 0.03)	0.95（± 0.04）	0.95（± 0.04）	0.93 (± 0.03)	0.96 (± 0.03)
	50	0.98 (± 0.01)	0.95 (± 0.02)	0.95 (± 0.01)	0.93 (± 0.03)	0.94（± 0.03）	0.92 (± 0.06)

表 VIII 确定性采样器在 Block Push 环境上的性能比较，在 10 个随机种子上平均，每个种子进行 100 次滚动执行。我们测试了在对数正态噪声分布上训练的 BESO，使用 $\sigma_{\text{MEAN}} = -0.17$、$\sigma_{\text{STD}} = -2$、$\sigma_{\text{MAX}} = 40.5$、$\sigma_{\text{MIN}} = 0.39$，并使用指数时间步。

算法 4 确定性一阶欧拉采样器 [17]

需要： 当前状态 $\boldsymbol{s}$，目标 $\mathbf{g}$
需要： 分数去噪模型 $D_\theta(\mathbf{a}, \boldsymbol{s}, \mathbf{g}, \sigma)$
需要： 噪声调度器 $\sigma_t = \sigma(t_i)$
要求： 离散时间步 $t_{i\in\{0,..,N\}}$
绘制采样 $\mathbf{a}_0 \sim \mathcal{N}(\mathbf{0}, \sigma_0^2\mathbf{I})$
对于 $i \in \{0, ..., N-1\}$ **do**
  $\mathbf{d}_i \leftarrow \big(\mathbf{a}_i - D_\theta(\mathbf{a}_i, \boldsymbol{s}, \mathbf{g}, \sigma_i)\big)/\sigma_i$
  $\mathbf{a}_{i+1} \leftarrow \mathbf{a}_i + (t_{i+1} - t_i)\mathbf{d}_i$
结束 for
return $\mathbf{a}_N$

算法 5 随机一阶欧拉采样器 [17]

要求： 当前状态 $\boldsymbol{s}$，目标 $\mathbf{g}$
需要： 分数去噪模型 $D_\theta(\mathbf{a}, \boldsymbol{s}, \mathbf{g}, \sigma)$
要求： 噪声调度器 $\sigma_i = t_i$，f_{ANC} 来自算法 C
要求： 离散时间步 $t_{i\in\{0,..,N\}}$
绘制采样 $\mathbf{a}_0 \sim \mathcal{N}(\mathbf{0}, \sigma_0^2\mathbf{I})$
对于 $i \in \{0, ..., N-1\}$ **do**
  $\mathbf{d}_i \leftarrow \big(\mathbf{a}_i - D_\theta(\mathbf{a}_i, \boldsymbol{s}, \mathbf{g}, \sigma_i)\big)/\sigma_i$
  $t_{\text{down}}, t_{\text{up}} \leftarrow f_{\text{ANC}}(t_i, t_{i+1})$
  $\mathbf{a}_{i+1} \leftarrow \mathbf{a}_i + (t_{\text{down}} - t_i)\mathbf{d}_i$
  $\epsilon_{\text{up}} \sim \mathcal{N}(\mathbf{0}, \sigma^2_{t_{\text{up}}}\mathbf{I})$
$\mathbf{a}_{i+1} \leftarrow \mathbf{a}_{i+1} + \epsilon_{\text{up}}$
结束循环以求得
return $\mathbf{a}_N$