arXiv:2303.11373v1 [cs.LG] 20 Mar 2023

神经约束满足：物体重排中组合泛化的层次抽象

Michael Chang[*]、**Alyssa L. Dayan**[*]、**Franziska Meier**[†]、
Thomas L. Griffiths[‡]、**Sergey Levine**[*]、**Amy Zhang**[†]. ⋆

摘要

物体重排对具身智能体而言是一项挑战，因为解决这类任务需要在实体及其位置的组合数量极其庞大的配置之间进行泛化。更糟的是，这些实体的表征是未知的，必须从感官感知中推断出来。我们提出一种层次抽象方法，以揭示这些潜在实体，并从非结构化视觉输入中实现组合泛化。通过在从像素推断出的实体表征簇上构建因子化转移图，我们展示了如何学习一种对应关系，将智能体模型中对实体状态的干预与环境中对物体的操作联系起来。我们利用这种对应关系开发了一种控制方法，该方法能够泛化到不同数量和配置的物体；在模拟重排任务上的评估结果表明，其表现优于当前的离线深度强化学习方法。项目网站。

1 引言

抽象的价值取决于它对解决新问题是否有用。物体重排（Batra 等，2020）为研究如何学习可复用的抽象提供了一个直观的场景。解决新颖的重排问题，要求智能体不仅能够在没有监督的情况下推断物体表征，还要认识到，将物体从一个位置移到另一个位置的同一动作，可以用于不同情境下的不同物体。

我们在仿真中研究最简单的设置，使用一次移动一个物体的抓取并移动动作原语。即便是如此简单的设置也颇具挑战，因为物体配置空间呈组合式增长，从而形成了长时域组合任务空间。我们将重排表述为一个离线目标条件强化学习（强化学习）问题：智能体先在一个由感觉运动交互构成的经验缓冲区上进行预训练，然后评估其能否根据输入图像中指定的物体生成重排动作，以满足目标图像所描绘的约束。

不推断实体因子化表示的离线强化学习方法（Levine 等，2020）难以泛化到物体数量更多的问题。然而，采用能够推断实体的以物体为中心的方法（Veerapaneni 等，2020）进行规划也并非易事，因为使用学习得到的参数化模型进行长时域规划的难题（Janner 等，2019）在组合空间中会更加严重。

我们不采用参数化模型进行规划，而是从非参数化规划方法中汲取灵感。这些方法已成功将神经网络与图搜索相结合，以生成长时域规划（Yang 等人，2020年；Zhang 等人，2018年；Lippi 等人，2020年；Emmons 等人，2020年），并明确构建

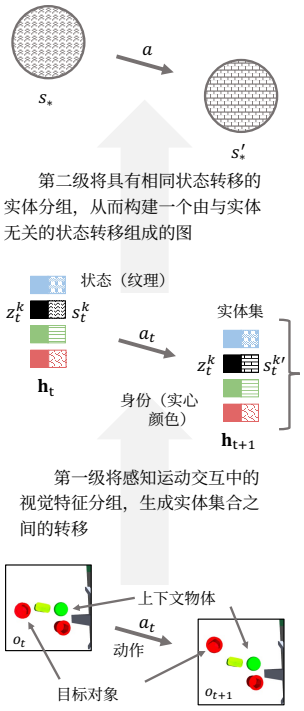


图 1：NCS 使用两层层次结构，将感觉运动交互抽象为由学习到的状态转移构成的图。受影响的实体以黑色显示。

[*]加州大学伯克利分校。 [†]Meta AI Research。 [‡]普林斯顿大学。 *德克萨斯大学奥斯汀分校。工作由 MC 在 Meta AI 实习期间完成。联系邮箱：mbchang@berkeley.edu、amyzhang@meta.com

一个来自经验缓冲区的转移图，并通过学习到的距离度量搜索图中记录的动作来进行规划。这类方法的优势在于，能够将离线数据中的不同路径片段拼接起来，以解决新问题。其缺点是，这类方法的非参数性质要求用于解决新问题的转移必须已记录在缓冲区中，因此，存储整体观测的传统方法并不适合组合泛化。图 2b 显示，同一状态转移可能出现在不同物体和不同情境中，但整体式非参数方法并未被约束去识别所有场景在抽象层面上都体现了相同的状态转移。这会导致图中的节点数量激增。为解决这一问题，我们设计了一种能够利用不同情境下状态转移之间相似性的方法。

我们的方法，**神经约束满足**（NCS）[1],融合了非参数规划与以对象为中心的表征各自的优势。我们的主要贡献在于证明：将传统上整体单一的实体表征因式分解为动作不变特征（其**类型**）和依赖动作的特征（其**状态**），使得规划与控制过程中能够在不同情境下为不同物体复用动作表征，从而应对物体重排中的核心组合挑战。为实现这种因子分解，NCS 构建了一个两层层次结构（图 1），将经验缓冲区抽象为一张图，其中包含各个实体的状态转移，并将其与其他上下文实体分离（图 3）。为解决新的重排问题，NCS 根据当前图像观测和目标图像观测推断可执行的状态转移，从图中重新组合状态转移序列，并将这些转移转换为动作。

在 §3 中，我们提出了一种问题形式化方法，通过明确建模潜在实体的独立性、对称性和因子分解，揭示物体重排任务的组合结构。这揭示了物体重排中的两个挑战，我们分别称之为**对应问题** 和**组合问题**。在 §4 中，我们介绍 NCS，这是一种控制智能体的方法：智能体基于涌现出的、经学习得到的实体表征进行规划并采取动作；该方法统一应对这两个挑战。在 §5 中，我们表明，在模拟的重排问题中，NCS 的表现优于最先进的离线强化学习方法和以物体为中心的基于射击法的规划方法。

2 相关工作

发现可重新组合的表征这一问题，通常由组合任务空间所引出。传统上，为了施加这种组合式归纳偏置，人们会利用人为定义的实体抽象所构成的马尔可夫决策过程，以紧凑地表征任务空间，例如因子化马尔可夫决策过程（Boutilier 等人，1995年；2000年；Guestrin 等人，2003a）、关系型马尔可夫决策过程（Wang 等人，2008年；Guestrin 等人，2003b；Gardiol 和 Kaelbling，2003年）以及面向对象的马尔可夫决策过程（Diuk 等人，2008年；Abel 等人，2015年）。基于此类符号抽象的方法（Chang 等人，2016年；Battaglia 等人，2018年；Zadaianchuk 等人，2022年；Bapst 等人，2019年；Zhang 等人，2018年）并未解决如何从原始数据中产生此类实体抽象的问题。我们的工作是首批直接从原始感觉运动数据中学习组合任务空间紧凑表征的研究之一。
近期以物体为中心的方法（Greff 等人，2017；Van Steenkiste 等人，2018年；Greff 等人，2019年；2020年；Locatello 等人，2020a；Kipf 等人，2021年；Zoran 等人，2021年；Singh 等人，2021年）确实会从感觉运动数据中学习实体表征及其变换（Goyal 等人，2021年；2020年），但仅将其用于图像和视频建模，而非执行动作。相反，我们研究实体表征在多大程度上可以重复用于解决任务。Kulkarni 等人（2019年）探讨了物体表征如何改善探索，但我们考虑的是需要零样本泛化的离线设置。Veerapaneni 等人（2020年）也研究控制任务，但其基于射击法的规划方法与其他学习得到的单步模型一样，会遭受累积误差的影响（Janner 等，2019），而我们的分层非参数方法则能够支持更长时域的规划。

非参数方法近年来在长时域规划中日益流行（Yang 等人，2020年；Zhang 等人，2018年；Lippi 等人，2020年；Emmons 等人，2020年；Zhang 等人，2021年），但这些方法的缺点在于，它们以整体方式表示整个场景，导致组合任务空间中的节点数量激增，因此无法应用于需要泛化到物体数量不同的新物体配置的重排任务。与我们的工作类似，Huang 等人（2019年）也通过在构建的潜在任务图上搜索来解决重排问题，但他们要求在部署时提供演示，

[1]关于此名称由来的说明，请参见附录 G。

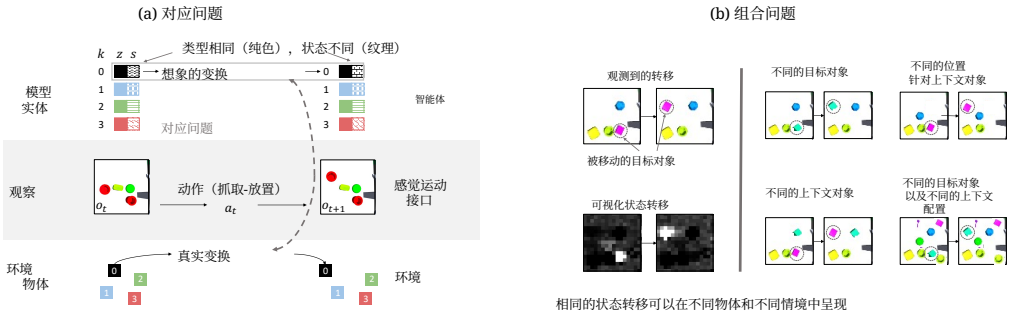


图 2：**解决物体重排需要应对两项挑战**。(a) **对应关系问题**是指将原始感觉运动信号抽象为实体表示，使智能体对实体的干预方式与其动作对环境中物体的影响之间建立对应关系的问题。 k 表示实体的索引， z 表示实体的类型（以纯色显示）， s 表示实体的状态（以纹理显示）。表示被移动物体的实体以黑色显示。(b) **组合问题**是指以一种能够让智能体将给定状态转移（以虚线圆圈标示）的知识迁移到不同情境的方式，表示组合任务空间的问题。

而 NCS 不需要，因为它会重用训练期间构建的上下文无关状态转移。Zhang 等人（2021年）直接在抽象子目标上进行非参数规划，而非以物体为中心的状态——尽管两者相似，但使用子目标而非抽象状态的缺点在于，这些子目标并不用于表示等价状态，因此无法在测试时泛化到新状态。我们的方法 NCS 既能捕捉已知状态之间的可达性，也能捕捉可映射到同一抽象状态的新状态（此前未见过的状态）。

3 基于目标的实体强化学习

本节介绍对标准目标条件部分可观测马尔可夫决策过程（POMDP）问题形式化的一系列修改，明确呈现以下类型物体重排任务的组合结构：“依次移动当前观察 o_1 中描绘的部分（或全部）物体，以满足目标图像 o_g中描绘的约束。”我们假设采用离线强化学习设置，智能体在转移缓冲区$\{(o_1,\ a_1,\ \ldots a_{T-1},\ o_T)\}_{n=1}^{N}$上进行训练，并在指定为 $(o_1,\ o_g)$ 的任务上接受评估。

标准 POMDP 问题形式化假设存在观测空间 $\mathcal{O}$、动作空间 $\mathcal{A}$、潜在空间 $\mathcal{H}$、目标空间 $\mathcal{G}$、观测函数 $E\colon \mathcal{H} \to \mathcal{O}$、转移函数 $P\colon \mathcal{H} \times \mathcal{A} \to \mathcal{H}$ 和奖励函数 $R\colon \mathcal{H} \times \mathcal{G} \to \mathbb{R}$。以这种方式对潜在空间进行整体建模，无法呈现不同场景之间的共性，例如包含位于相同位置的物体的场景，或包含同一类型多个实例的场景。这使我们无法设计能够利用这些共性来压缩组合任务空间的控制算法。

为解决这一问题，我们在标准表述中引入独立性、对称性和因子分解这几项结构性假设。独立性假设编码了这样一种直观属性：对某个物体施加动作不会影响其他物体。具体而言，将潜在空间分解为相互独立的子空间，如 $\mathcal{H} = \mathcal{H}^1 \times \ldots \times \mathcal{H}^K$，场景中的每个独立自由度（例如物体）对应一个子空间。对称性 假设编码了这样一种属性：所有物体都遵循相同的物理定律。具体而言，观察、转移和奖励函数在所有子空间之间共享，从而将 $\mathcal{H}^1 = \ldots = \mathcal{H}^K$视为相同对象。我们将此类子空间的成员定义为一个**实体**[2] $h \in \mathcal{H}^k$，并将能够解释某个观察的实体集合$\mathbf{h} = (h^1, \ldots, h^K)$定义为一个**实体集**，这与 Diuk 等人（2008年）和 Weld（2003年）的做法类似。最后，因子分解 假设表明，每个子空间都可以分解为 $\mathcal{H}^k = \mathcal{Z} \times \mathcal{S}$，其中$z \in \mathcal{Z}$ 表示实体的外观等动作不变特征，而 $s \in \mathcal{S}$ 表示位置等动作相关特征。我们将 z 称为**类型**，将 s 称为**状态**。

引入这些假设，解决了上述不同场景之间共性建模的问题。通过让不同场景中的实体共享同一状态 s，我们可以描述包含位于同一位置的物体的场景。通过让场景中的多个实体共享同一类型 z，我们可以描述包含多个同类型物体实例的场景。这种表述也使我们能够自然地将目标表示为一组约束

[2]我们用“物体”指代环境中的一个独立自由度，用“实体”指代智能体对该物体的表示。

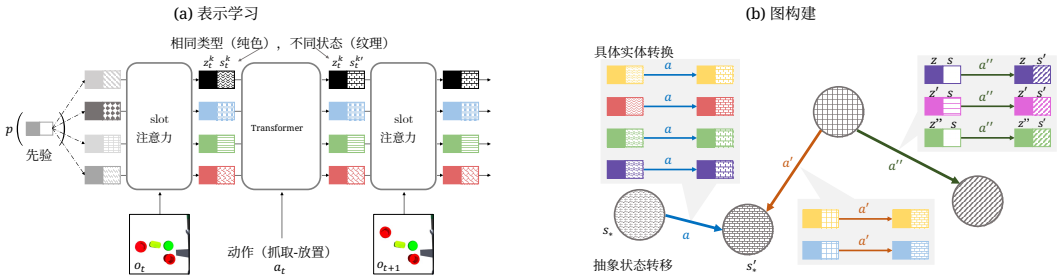


图 3：**建模**。NCS 构建了一个两级抽象层次结构，以对经验缓冲区中的转移进行建模。**(a) 第一级：** NCS 学习从包含抓取并移动动作的感觉运动转移中推断出一组实体，每次转移中有一个实体被移动。我们通过约束实体的类型 z（以纯色显示）在不同时间步之间保持不变来实现这一点。GPT 动力学模型学习以稀疏方式预测实体在下一个时间步的状态 s（以纹理显示）。这通过迫使网络经由实体瓶颈来预测和重建观测，从而解决了对应问题。**(b) 第 2 层：** NCS 将实体集合上的转移抽象为单个实体状态之间的转移，并构建一个图，其中状态是节点，状态之间的转移是边。具体做法是对具有相似初始状态和最终状态的实体转移进行聚类。这通过使状态转移能够复用于不同实体类型以及具有不同上下文实体的情形，解决了组合问题。

$\mathbf{h}_g = (h_g^1, \ldots\ldots, h_g^k)$。解决任务，就是采取动作，将初始观察 o_1 中类型由$\mathbf{z}_g$给定的实体子集转换为由$\mathbf{s}_g$指定的新状态。

在问题形式化中揭示这种结构，为我们提供了一套语言，用以设计能够以独立、对称且因子化的方式表征实体，并利用这三种属性压缩组合任务空间的方法。这些方法需要解决两个问题：学习以这种方式表征实体的**对应问题**，以及利用这些属性使规划变得可处理的**组合问题**。对应问题之所以困难，是因为它假设没有人工监督来说明实体是什么。它也超出了现有针对图像和视频的以物体为中心的方法所解决的问题，因为它涉及动作：它要求对实体进行表征，使智能体对其动作如何影响实体的建模，与这些动作在环境中实际如何影响物体相对应。组合问题则超出了用于求解面向对象的马尔可夫决策过程、关系型马尔可夫决策过程和因子化马尔可夫决策过程的方法所解决的问题，因为它要求智能体利用学得而非由人工定义的实体表征，识别先前观察到的状态转移是否以及如何能用于解决新问题。评估这两个问题的自然判据是，测试智能体在多大程度上能够通过零样本泛化，解决涉及新物体配置集合的重排任务；这些新配置与训练中观察到的配置互不重叠，并假设训练配置总体上已覆盖 $\mathcal{Z}$ 和 $\mathcal{S}$。§5中的实验正是对此进行测试。

简化假设 为了聚焦于重排的组合特性，我们不关注低层操作，因此将每个动作表示为$(w, \Delta w)$，其中 w 为笛卡尔坐标 $w = (x, y, z)$。我们假设动作稀疏地一次影响一个实体，并且动作对物体状态的影响不取决于物体的身份。我们不关注处理遮挡，因此假设物体被限制在 xy 平面或 xz 平面上，且摄像机可以直接看到它们。遵循先前的研究（Hansen-Estruch 等人，2022年；Castro 等人，2009年），我们作出一个双模拟 假设：状态空间可以划分为有限个等价类，并且每一对等价类之间都存在一个动作原语，可实现二者之间的转移。最后，我们假设物体可以独立移动。初步实验表明，NCS 可以扩展以支持堆叠方块等涉及物体间依赖关系的任务，但如何处理这些依赖关系值得在未来工作中单独探讨。

4 神经约束满足

在 §3 中，我们介绍了物体重排的结构化问题形式化，并将其归结为求解对应问题和组合问题。现在，我们介绍我们的方法——神经约束满足（NCS），这是一种控制智能体的方法；该智能体基于由学习得到的实体表征构建的状态转移图进行规划并采取动作。本节分为两部分：建模和控制。建模部分又分为两部分：表示学习和图构建。表示学习部分解决对应问题，而图构建和控制部分解决组合问题。

4.1 建模

NCS 的建模组件将经验缓冲区抽象为因子化状态转移图，使其能够在不同的重排问题中重复使用。下面我们将说明，如何先训练一个以对象为中心的世界模型，以推断出相互独立、对称且因子化的实体，然后通过聚类状态转移相似的实体来构建状态转移图。这两个步骤构成了针对原始感觉运动转移的两级抽象层次结构。

第一级：表示学习 第一级关注实体表征的无监督学习，这些表征可分解为动作不变特征（其**类型**）和动作相关特征（其**状态**）。具体而言，我们的目标是将视频转移 $o_t,\ a_t \rightarrow o_{t+1}$ 建模为实体集合上的转移 $\mathbf{h}_t,\ a_t \rightarrow \mathbf{h}_{t+1}$，其中每个实体 h^k 都分解为一对 $h^k = (z^k,\ s^k)$。在我们的设定中，环境里的一个动作每次只会移动一个物体，因此成功的表示学习意味着满足三个判据：(1) 世界模型能够正确识别与被移动物体对应的单个实体 h^k；(2) 只有该实体的状态 s^k 会改变，而其类型 z^k 应保持不变；(3) 其他实体表征 $h^{\neq k}$ 也应保持不变。判据 (1) 和 (3) 排除了用整体式表征表示整个场景的标准方法，因此我们需要以对象为中心的世界模型，而非整体式世界模型。但判据 (2) 排除了标准的以对象为中心的世界模型（例如 (Veerapaneni 等，2020；Elsayed 等，2022年；Singh 等人，2022年b)），因为这些模型不会将实体表征分解为动作不变特征和动作相关特征。

由于混合模型的参数在构造上彼此独立且具有对称性，我们提出将因子化的物体中心世界模型构建为一个等变序贯贝叶斯滤波器，以混合模型作为潜在状态，其中实体表征是混合分量的参数。回顾一下，滤波器由两个主要部分组成：潜变量估计和潜变量预测。我们基于混合分量与 SA 槽之间的联系（Chang 等人，2022年），采用最先进的槽注意力（SA）（Locatello 等人，2020b）实现潜变量估计。我们采用 Transformer 解码器（TFD）架构（Vaswani 等人，2017）实现潜变量预测，因为 TFD 关于其输入具有等变性。我们将 SA 槽表示为 $\boldsymbol{\lambda}$，将 SA 注意力掩码表示为 $\boldsymbol{\alpha}$。我们将每个槽位 $\lambda \in \mathbb{R}^n$ 拆分为两半 $\lambda^z \in \mathbb{R}^{\frac{n}{2}}$ 和 $\lambda^s \in \mathbb{R}^{\frac{n}{2}}$。给定观测 o 和动作 a，我们使用前馈网络将动作嵌入为 $\tilde{a}$，并将滤波器实现为：

$$\boldsymbol{\lambda}_1 \sim \text{高斯分布} \qquad \hat{\boldsymbol{\lambda}}^s{}_{t+1} = TFD\left(\text{查询 } = \boldsymbol{\lambda}^s_t,\ \text{键/值 } = [\boldsymbol{\lambda}^s, \tilde{a}_t]\right)$$

$$\boldsymbol{\lambda}_t, \boldsymbol{\alpha}_t = SA\left(\hat{\boldsymbol{\lambda}}_t, o_t\right) \qquad \hat{\boldsymbol{\lambda}}_{t+1} = \left[\boldsymbol{\lambda}^z_t, \hat{\boldsymbol{\lambda}}^s{}_{t+1}\right]$$

其中，$[\cdot,\cdot]$ 是拼接运算符，$\boldsymbol{\lambda}$ 是潜在预测步骤的输出，$\boldsymbol{\lambda}$ 是潜在估计步骤的输出。我们将此滤波器嵌入 SLATE 主干网络（Singh 等人，2022a）中，并将这一实现称为 **动态 SLATE**（dSLATE）。有关 SLATE 的背景介绍以及 dSLATE 的超参数，请参阅附录 A.1。

通过将 $\boldsymbol{\lambda}^z{}_{t+1}$ 构造为 $\boldsymbol{\lambda}^z_t$ 的副本，dSLATE 强制要求 $\boldsymbol{\lambda}^z$ 中包含的信息与动作无关，因此我们将 $\boldsymbol{\lambda}^z$ 视为 dSLATE 对实体类型的表示。至于实体的状态，可以使用槽位的动作相关部分 $\boldsymbol{\lambda}^s$，也可以使用注意力掩码 α。如果所有物体看起来都相似且不存在遮挡，使用 α 可能就足够了，而且更便于直观分析；而在其他情况下，$\boldsymbol{\lambda}^s$ 可能更合适。我们在实验中分别给出了一个示例。为简化后续的符号表示，并与 §3 中的符号保持一致，我们用 $\mathbf{h}$ 表示 $(\boldsymbol{\lambda},\ \boldsymbol{\alpha})$，用 $\boldsymbol{z}$ 表示 $\boldsymbol{\lambda}^z$，并用 $\boldsymbol{s}$ 表示 $\boldsymbol{\lambda}^s$ 或 $\boldsymbol{\alpha}$。因此，按照构造，dSLATE 满足判据 (2)。从经验上看，我们观察到它与 SLATE 一样满足判据 (1)，并且 TFD 学会了稀疏地编辑 $\boldsymbol{\lambda}^s_t$，从而满足判据 (3)。

第 2 层：图构建 在第一个层级生成了一个包含实体集转换的缓冲区 $\{\mathbf{h}_t,\ a_t \rightarrow \mathbf{h}_{t+1}\}_{n=1}^N$ 后，第二个层级的目标（图 3b）是利用该缓冲区构建一个因子化状态转移图。解决组合问题的关键，是构建该图的边，使其表示的不是整个实体集合的状态转移（即 $\mathbf{s}_t,\ a_t \rightarrow \mathbf{s}_{t+1}$），如先前工作所做的那样（Zhang 等人，2018），而是单个实体 的状态转移（即 $s^k_t,\ a_t \rightarrow s^k_{t+1}$）。围绕单个实体的转移而非实体集合构建边，可以使同一转移在存在不同上下文实体时重复使用。围绕状态转移而非实体转移构建边，则可以使同一转移在类型不同的实体之间重复使用。这样，智能体便能重新组合此前遇到的状态转移序列，以解决不同情境下涉及不同实体的重排问题。

新的重排问题。因此，除非另有说明，以下所说的“状态”特指单个实体的状态。

基于状态可划分为有限个组这一双模拟假设，我们构建图，使节点表示各个状态之间的等价类，边则表示将状态从一个等价类转换到另一个等价类的动作。为实现这一点，我们对缓冲区中各个实体的状态转移进行聚类，这相当于对各个实体在转移前后的状态进行聚类。我们将每个簇质心视为图中的一个节点，并用将一个节点的状态转换为另一个节点状态的单个动作标记节点间的边。构建图的算法如算法 1 所示，包含三个步骤：(1) 从实体集的状态转移中分离出单个实体的状态转移；(2) 根据状态簇创建图节点；(3) 用动作标记图边。

第一步是确定每次转移中移动了哪个物体，也就是确定 dSLATE 预测在转移中受到 a_t影响的实体 h^k（$\mathbf{h}_t$，a_t，$\mathbf{h}_{t+1}$）。我们通过实现函数 isolate 来完成此操作，该函数通过求解$k = \arg\max_{k' \in \{1,\dots,K\}} d(s_t^{k'}, s_{t+1}^{k'})$，找出转移过程中状态变化最大的实体的索引。其中，$d(\cdot, \cdot)$ 是距离函数，详见附录中的表 3。这样便将实体集合上的转移缓冲区 $\mathbf{h}_t$，$a_t \to \mathbf{h}_{t+1}$ 转换为单个实体上的转移缓冲区h_t^k，$a_t \to h_{t+1}^k$。

算法 1构建图

1: 输入模型、缓冲区 2: 对于缓冲区中的 $\{(o_t, a_t, o_{t+1})\}_n$，执行 3: # 从转移中推断实体 4: $\{(\mathbf{h}_t, a_t, \mathbf{h}_{t+1})\}_n \leftarrow$ 模型 $(\{o_t, a_t, o_{t+1}\}_n)$。 5: # 确定转移中发生变化的实体 6: $\{(h_t^k, a_t, h_{t+1}^k)\}_n \leftarrow$ isolate $(\{(\mathbf{h}_t, a_t, \mathbf{h}_{t+1})\}_n)$ 7: 结束循环 8: # 通过聚类实体对转移进行划分 9: $\{s_*\}_{m=1}^M \leftarrow$ 聚类 $(\{(s_t^k, a_t, s_{t+1}^k)\}_{n=1}^N)$10: # 簇之间的转移即为边 11: 使用节点初始化图 $s_*^{[m]}$，对于 $m \in [1: M]$12: 对每个 $\{(h_t^k, a_t, h_{t+1}^k)\}_n$ 执行 13: # 推断簇分配 14: $[i], [j] \leftarrow$ 绑定 (h_t^k)，绑定 (h_{t+1}^k)15: # 使用动作标记边 a_t16: graph.edges $[i, j] \leftarrow$ 创建边 $\left(s_*^{[i]} \overset{a_t}{\to} s_*^{[j]}\right)$17: 结束循环 18: 返回图

第二步是对每次转移前后的状态进行聚类。我们实现了一个名为 cluster 的函数，它使用 K 均值将这些状态簇的质心作为图节点返回 $\{s_*\}_{m=1}^M$。

第三步是用边连接这些节点，边记录缓冲区中实际执行的动作，这些动作将一个状态转变为下一个状态。我们实现了一个名为 bind 的函数，给定实体 h^k，该函数返回质心 s_* 的索引 $[i]$，此质心是与该实体的状态 s^k最近邻的质心。对于每次实体转移（h_t^k，a_t，h_{t+1}^k），我们将实体 h_t^k和 h_{t+1}^k 绑定到各自对应的节点 $s_*^{[i]}$和 $s_*^{[j]}$，并在 $s_*^{[i]}$和 $s_*^{[j]}$之间创建一条标记为动作 a的边；基于这样的假设覆盖之前的边：只要聚类恰当，每对节点之间就应该只有一个动作。

在我们的实验中，cluster 和 bind 使用相同的距离度量（见附录中的表 2），但也可以使用其他聚类算法和距离度量。我们的实验（图 11）还表明，每对节点可以对应多个动作原语，前提是这些动作都在绑定到同一对节点的状态之间映射。

4.2 控制

为了解决新的重排问题，我们从图中重新组合状态转移序列。具体而言，智能体将重排问题分解为一组针对各实体的子问题（例如，单个物体的初始位置和目标位置），在转移图中搜索能够将当前实体状态转变为其目标状态的转移，并在环境中执行与该转移关联的动作。由于我们构建图中的转移时不考虑类型和上下文，因此不同的重排问题可以共享相同子问题的解决方案，从而实现这种问题分解。决定组合哪些转移的核心挑战，在于确定哪些转移可能能够组合。也就是说，智能体必须判断图中的哪些节点对应给定的目标约束，哪些节点对应当前观测中的实体；但当前实体$\mathbf{h}_t$ 和目标约束$\mathbf{h}_g$ 本身也必须根据当前观测和目标观测 o_t 及 o_g推断得出，因此智能体需要仅凭感觉运动接口，同时推断该做什么以及如何去做。

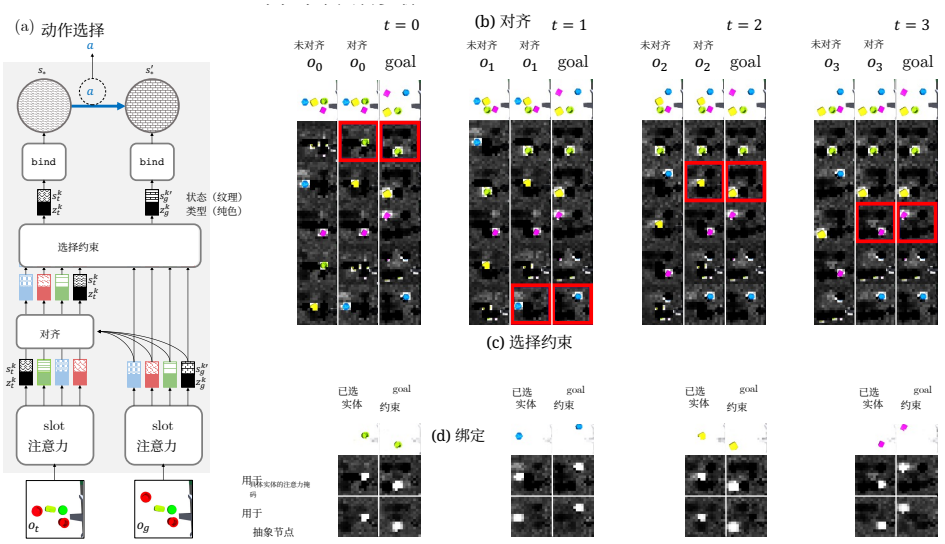


图 4：**规划与控制**。给定一个仅由当前观测和目标观测（o_0，o_g）指定的重排问题，NCS 会将该重排问题分解为每个实体对应的一个子问题（o_t，o_g）。(a) 展示了 NCS 用于求解每个子问题的计算过程，(b-d) 则结合具体情境展示了这些步骤。对于每个子问题（o_t，o_g），NCS 会根据当前观测和目标观测推断实体。目标实体的状态表示了对当前实体期望位置的约束。(b) NCS 将当前实体的索引与类型相对应的目标实体索引对齐。(c) 如红框所示，NCS 选择索引 k，以满足下一个目标约束 s_g^k。在 (a) 中，已选目标约束和当前实体也以黑色标出；注意，它们的类型相同，但状态不同：我们希望选择一个动作，将当前实体的状态转换为目标约束所要求的状态。(d) NCS 将已选目标约束及其对应的当前实体绑定到转移图中的节点 s_* 和 s'_*。最后，它会识别连接这两个节点的边，并在环境中执行与该边关联的动作。

我们的方法分为四个步骤，如算法 2 和图 4 所示，更多细节见附录 A.3。第一步，我们使用 dSLATE，从o_t 和 o_g （例如，初始图像和目标图像中所有物体的位置和类型）推断出$\mathbf{h}_t$ 和 $\mathbf{h}_g$。第二步（图 4b），由于实体之间存在排列对称性，我们找到一种二分匹配，将 h_g^j 中的每个实体与 h_t^k 中类型相同的对应实体匹配起来，并对$\mathbf{h}_t$ 的索引 k 进行排列，使其与$\mathbf{h}_g$的索引对齐。我们实现了一个 align 函数，使用匈牙利算法在（z_t^1，$...z_t^K$）和（z_g^1，$...z_g^K$）之间执行这种匹配，并以欧几里得距离作为匹配代价。第三步选择接下来要满足的目标约束 h_g^k（图 4c）。我们通过确定差异最大的约束 h_g^k 来实现这一选择约束流程

算法 2动作选择

1：给定模型、图 2：输入目标 o_g、观察 o_t3：# 推断目标约束和当前实体 4：$\mathbf{h_g}$、$\mathbf{h_t}$ ← 模型（o_g），模型（o_t） 5：将 $\mathbf{h_t}$ 的实体索引与 $\mathbf{h_g}$ 的索引对齐 6：$\pi \leftarrow$ 对齐（$\mathbf{h_t}$，$\mathbf{h_g}$） 7：根据 π置换 $\mathbf{h_t}$ 的索引 8：$\mathbf{h_t} \leftarrow$（$h_t^{\pi[1]}$，...，$h_t^{\pi[K]}$） 9：识别下一个要满足的第 k 个目标约束 10：$k \leftarrow$选择约束（$\mathbf{h_t}$，$\mathbf{h_g}$） 11：推断簇分配 12：$[i]$、$[j] \leftarrow$绑定 (h_t^k)、绑定 (h_g^k) 13：将节点 $[i]$ 转变为节点 $[j]$ 的动作 14：返回 graph.edges$[i, j]$.action

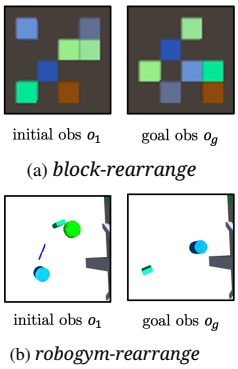
状态及其对应项 h_t^k，这归结为在 isolate 中使用相同的距离函数求解相同的 argmax问题。最后一步根据选定的目标约束 h_g^k 及其对应项 h_t^k，依据 h_t^k和 h_g^k 的状态分量将它们绑定到图中，并返回标记在它们各自节点之间的边上的动作（图4d）。如果推断出的节点之间不存在边，我们就随机采取一个动作。

5 实验

我们提出了 NCS，作为物体重排问题的一种解决方案，以应对两项挑战：NCS 通过使用 dSLATE 学习因子化的物体中心世界

模型来解决对应问题，并通过将实体表示抽象为可查询的状态转移图来解决组合问题。现在，我们来检验 NCS 解决这两个问题的效果。关键问题是，与最先进的离线强化学习算法相比，NCS 是否更善于从感知输入中对具有组合结构的任务空间进行泛化。如 §3 所述，要回答这一问题，关键在于评估所有方法解决新重排问题的能力，而这些问题所涉及的物体配置与训练期间遇到的配置互不重叠。要找出组合空间中互不重叠的子集，最直接的方法是使用不同数量的物体进行评估。我们在两个重排环境中，将 NCS 与若干离线强化学习基线及消融实验进行比较，发现我们的方法与次优方法之间存在显著的性能差距。

环境。 在*block-rearrange* （图 5a）中，所有物体的大小、形状和朝向都相同。 $\mathcal{S}$ 覆盖网格中的 16 个位置。 $\mathcal{Z}$ 是红绿蓝值从 0 到 1 的连续空间。*robogym-rearrange*（图 5b）改编自 OpenAI（2020）的 rearrange 环境，并移除了*block-rearrange* 中所有物体大小、形状和朝向均相同的假设。物体从包含 94 个网格模型的集合中均匀采样，该集合由 YCB 物体集（Calli 等人，2015年）和一组基本几何形状组成；颜色则从 13 种颜色中采样。尽管*robogym-rearrange*中的位置不像*block-rearrange*那样预先定义，但实际上，物体在桌面上的摆放方式受到限制，否则摄像机将无法看到它们，因此双模拟在此仍是一个合理的假设。对于*block-rearrange*，我们使用 SA注意力掩码 α 作为状态 s；对于*robogym-rearrange*，我们使用 SA 槽位 $\boldsymbol{\lambda}^s$ 中与动作相关的部分作为状态 s。更多环境细节见附录 B。



(a) *block-rearrange*

(b) *robogym-rearrange*

图 5：我们的环境包括 *block-rearrange* 和 *robogym-rearrange*。图 5a 展示了目标约束的完整规格；图 5b 展示了部分规范，仅指定两个物体的目标位置。

实验设置。 我们评估两种设定：完整 和部分。在完整 设定中，目标图像显示所有物体都位于新的位置。部分 设定的规格不完整：只有一部分物体具有相关的目标约束（图 5b）。在*block-rearrange*中，初始状态下所有约束均未满足。在*robogym-rearrange*中，初始状态下有四个约束未满足。我们的指标是分数成功率，即满足的约束数量的平均变化量除以初始时未满足的约束数量。

经验缓冲区由 5000 条展示 4 个物体的轨迹组成。我们在包含 4 至 7 个物体的任务上进行评估，每种设置运行 100 个回合，使用 10 个随机种子。即使假设能够完全访问底层状态空间，任务空间仍然极其庞大：有 $|S|$ 个物体位置和 k 个物体时，在 t 个时间步内，针对物体配置的可能轨迹数量为 $\binom{|S|}{k} \times (k \times (|S|-k))^t$。对于包含 $k=7$ 个物体的*block-rearrange* 完整规格设定，这意味着需要在超过 10^{16} 条可能轨迹中进行搜索（推导见附录 E）。而我们假设只能访问像素，这使问题更加困难。

基线方法。 本文的主张是，对于物体重排，（1）以物体为中心的方法优于结构单一的离线强化学习方法；（2）对于物体重排，非参数图搜索优于参数化规划；（3）基于各个实体状态转移的因子化图搜索，优于基于整个实体集合状态转移的非因子化图搜索。为检验（1），我们将方法与基于 Kostrikov（2021）实现的最先进的基于像素的行为克隆（BC）和隐式 Q 学习（IQL）进行比较。为检验（2），我们将方法与 Veerapaneni 等人（2020）的一个版本的以物体为中心的模型预测控制（MPC）进行比较；该方法使用交叉熵方法处理 dSLATE 展开轨迹。为检验（3），我们将方法与一项消融实验（简称 NF，即“非因子化”）进行比较，该实验基于实体集合的状态转移而非单个实体的状态转移构建图。最后一个基线方法仅执行随机动作（Rand）。基线方法的实现细节见附录 C。

5.1 结果

图 1 显示，NCS 的表现显著优于所有基线方法（提升约 5–10 倍），从而在我们的实验背景下驳斥了与本文主张相悖的其他解释。大多数基线方法的表现不如随机方法，或仅略胜于随机方法。我们发现，使用经过训练、用于预测单个未来时刻的以实体为中心的世界模型进行射击式规划确实很困难（Janner 等人，2019）：MPC 基线表现不佳，因为其展开轨迹

表现较差，而且运行所需的计算量显著更大（耗时 11 小时，而非 20 分钟）。我们还发现，NF 消融实验的表现不佳，这表明对非参数图搜索进行因式分解十分重要。更多结果见附录 D，局限性见附录 F

表 1：此表比较了 NCS 与多种基线方法在 *block-rearrange* 和 *robogym-rearrange*的完整和部分评估设置下的表现。这些方法在 4 个物体上进行训练，并在泛化到 4、5、6 和 7 个物体时进行评估。我们报告分数成功率，其标准误基于 10 个随机种子计算。

(a) *block-rearrange*，完整规格。

方法	4	5	6	7
NCS（我们的方法）	**0.94** ± 0.01	**0.93** ± 0.00	**0.93** ± 0.00	**0.89** ± 0.00
Rand	0.06 ± 0.02	0.07 ± 0.03	0.07 ± 0.03	0.08 ± 0.03
MPC	0.16 ± 0.06	0.12 ± 0.04	0.11 ± 0.04	0.10 ± 0.03
NF	0.07 ± 0.03	0.06 ± 0.02	0.07 ± 0.02	0.08 ± 0.03
IQL	0.07 ± 0.01	0.03 ± 0.00	0.02 ± 0.00	0.02 ± 0.00
BC	0.03 ± 0.00	0.02 ± 0.00	0.01 ± 0.00	0.01 ± 0.00

(b) *block-rearrange*，完整规格。

方法	4	5	6	7
NCS（我们的方法）	**0.89** ± 0.01	**0.86** ± 0.01	**0.78** ± 0.01	**0.70** ± 0.01
Rand	0.06 ± 0.02	0.08 ± 0.03	0.08 ± 0.03	0.08 ± 0.03
MPC	0.13 ± 0.05	0.11 ± 0.04	0.10 ± 0.04	0.08 ± 0.03
NF	0.06 ± 0.03	0.07 ± 0.03	0.08 ± 0.03	0.07 ± 0.03
IQL	0.01 ± 0.01	0.07 ± 0.01	0.05 ± 0.01	0.05 ± 0.00
BC	0.05 ± 0.01	0.04 ± 0.00	0.03 ± 0.00	0.03 ± 0.00

(c)*robogym-rearrange*，完整规格。

方法	4	5	6	7
NCS（我们的方法）	**0.64** ± 0.01	**0.47** ± 0.01	**0.49** ± 0.01	**0.41** ± 0.01
Rand	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
MPC	0.00 ± 0.00	0.00 ± 0.00	0。00 ± 0.00	0.00 ± 0.00
NF	0.01 ± 0.00	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
IQL	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
BC	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00

(d) *robogym-rearrange*，部分规范。

方法	4	5	6	7
NCS（我们的方法）	**0.47** ± 0.01	**0.33** ± 0.01	**0.27** ± 0.01	**0.22** ± 0.01
Rand	0.005 ± 0.001	0.001 ± 0.00	0.002 ± 0.001	0.001 ± 0.00
MPC	0.00 ± 0.00	0.001 ± 0.001	0.00 ± 0.00	0.00 ± 0.00
NF	0.005 ± 0.001	0.001 ± 0.00	0.002 ± 0.001	0.001 ± 0.00
IQL	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
BC	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00

5.2 分析

在定量展示了 NCS 从像素中实现组合泛化的相对优势后，我们现在考察两项关键设计选择如何影响 NCS 的行为与性能：（1）将实体表征因式分解为动作不变特征和动作相关特征；（2）查询由动作相关特征构建的状态转移图。在潜变量预测过程中像 dSLATE 那样复制实体类型，是否足以将物体的位置和外观分别解缠结为状态和类型？dSLATE 是否学会只稀疏地修改传感运动转移中与被移动物体对应的实体，从而使状态转移图的节点能够在不同实体间有意义地复用？这些能力并非轻而易举就能获得，因为 NCS 仅通过经验缓冲区进行自监督学习。

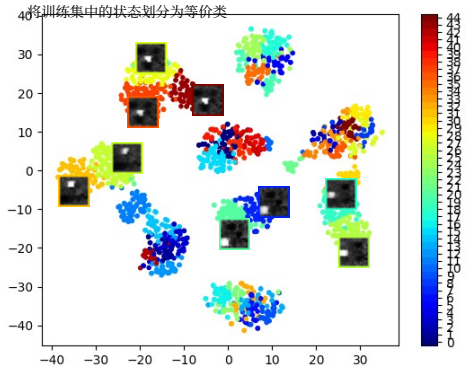


图 6：**状态上的等价类节点**。我们展示了为 *robogym-rearrange*推断出的状态聚类，其中每个簇质心都被视为转移图中的一个节点。我们用注意力掩码标记部分簇，该掩码通过对与该簇关联实体的槽位注意力掩码求平均得到。

图 4b 展示了 NCS 在*robogym-rearrange*上的对齐、选择约束和绑定函数，这表明，至少对于我们考虑的简化设置，这两个问题的答案都是肯定的。NCS 已学会将不同物体表示在不同槽位中，并构建一个节点能够捕捉位置信息的图。图 6 展示了一个 t-SNE（Van der Maaten 和 Hinton，2008年）图，将从 *Robogym* 环境中推断出的实体聚类。由于我们没有对状态应表示什么提供监督，因此观察到有多个簇索引映射到相似的点群。这表明，$\mathcal{S}$ 的多个不同区域似乎在建模相似的状态。我们还尝试合并冗余簇，但发现这并未提升定量性能。

6 讨论

物体重排为研究智能体如何从感觉运动经验中学习可复用抽象提供了一个直观的场景。本文通过引入 NCS，初步尝试将符号规划领域与人为定义的抽象，以及表示学习领域与深度网络联系起来。NCS 是一种控制智能体的方法：它利用从原始感觉运动转移中学习到的实体表征构建状态转移图，并在图上进行规划和执行动作，无需任何其他监督。我们证明，对实体表征进行因式分解

为动作不变特征和动作相关特征，对于解决使物体重排变得困难的对应关系问题和组合问题至关重要，并使 NCS 在物体重排的组合泛化方面显著优于现有方法。NCS 的实现为如何学习可复用抽象提供了概念验证，我们希望这能启发未来的研究，针对现实场景开发类似 NCS 的方法。

致谢

这项工作是在 MC 于 Meta AI 实习期间完成的。我们感谢 Leslie Kaelbling 提供宝贵的反馈，也感谢 Yash Sharma 和 Yilun Du 进行了富有价值的讨论。本研究部分得到了 Fannie and John Hertz Foundation 的支持，并获得了 ONR 项目资助（编号：#N00014- 18-1-2873）。

参考文献

David Abel、D. Ellis Hershkowitz、Gabriel Barth-Maron、Stephen Brawner、Kevin O'Farrell、James MacGlashan 和 Stefanie Tellex。基于目标的动作先验。发表于*ICAPS*，第 306–314 页。AAAI Press，2015年。

Victor Bapst、Alvaro Sanchez-Gonzalez、Carl Doersch、Kimberly Stachenfeld、Pushmeet Kohli、Peter Battaglia 和 Jessica Hamrick。用于物理构建的结构化智能体。发表于国际机器学习会议，第 464–474 页。PMLR，2019年。

Dhruv Batra、AngelXChang、Sonia Chernova、Andrew J Davison、Jia Deng、VladlenKoltun、Sergey Levine、Jitendra Malik、Igor Mordatch、Roozbeh Mottaghi 等。重排：具身人工智能面临的一项挑战。*arXiv* 预印本 *arXiv:2011.01975*，2020年。

Peter W Battaglia、Jessica B Hamrick、Victor Bapst、Alvaro Sanchez-Gonzalez、Vinicius Zambaldi、Mateusz Malinowski、Andrea Tacchetti、David Raposo、Adam Santoro、Ryan Faulkner 等。关系归纳偏置、深度学习与图网络。*arXiv* 预印本 *arXiv:1806.01261*，2018年。

Craig Boutilier、Richard Dearden 和 Moises Goldszmidt。利用策略构建中的结构。载于第*14*届国际人工智能联合会议论文集——第*2*卷，IJCAI'95，第1104–1111页，美国加利福尼亚州旧金山，1995年。Morgan Kaufmann 出版社。

Craig Boutilier、Richard Dearden 和 Mois´es Goldszmidt。基于因子化表示的随机动态规划。人工智能，121(1-2):49–107，2000年。

Berk Calli、Arjun Singh、Aaron Walsman、Siddhartha Srinivasa、Pieter Abbeel 和 Aaron M Dollar。YCB物体与模型集：迈向操作研究的通用基准测试。载于*2015*年国际先进机器人会议（*ICAR*），第510–517页。IEEE，2015年。

Pablo Samuel Castro、Prakash Panangaden 和 Doina Precup。完全可观测与部分可观测马尔可夫决策过程中的等价关系。载于第二十一届国际联合人工智能会议。Citeseer，2009年。

Chien-Yi Chang、De-An Huang、Danfei Xu、Ehsan Adeli、Li Fei-Fei 和 Juan Carlos Niebles。教学视频中的程序规划。载于欧洲计算机视觉会议，第334–350页。施普林格，2020年。

Michael Chang、Thomas L Griffiths 和 Sergey Levine。将物体表征视为不动点：通过隐式微分训练迭代细化算法。*arXiv* 预印本 *arXiv:2207.00787*，2022年。

Michael B Chang、Tomer Ullman、Antonio Torralba 和 Joshua B Tenenbaum。学习物理动力学的组合式物体方法。*arXiv*预印本 *arXiv:1612.00341*，2016年。

Erwin Coumans 和 Yunfei Bai。PyBullet：用于游戏、机器人学和机器学习物理仿真的 Python 模块。2016年。

Coline Devin、Payam Rowghanian、Chris Vigorito、Will Richards 和 Khashayar Rohanimanesh。自监督目标条件抓取与放置。*arXiv*预印本 *arXiv:2008.11466*，2020年。

Carlos Diuk、Andre Cohen 和 Michael L. Littman。用于高效强化学习的面向对象表示。载于 *ICML*，第 307 卷，*ACM*国际会议论文集系列，第 240–247 页。ACM，2008年。

Gamaleldin F Elsayed、Aravindh Mahendran、Sjoerd van Steenkiste、Klaus Greff、Michael C Mozer 和 Thomas Kipf。Savi++：迈向从真实世界视频进行端到端对象中心学习。*arXiv* 预印本 *arXiv:2206.07764*，2022年。Scott Emmons、Ajay Jain、Misha Laskin、Thanard Kurutach、Pieter Abbeel 和 Deepak Pathak。用于鲁棒规划的稀疏图记忆。神经信息处理系统进展，33：5251–5262，2020年。

Ben Eysenbach、Russ R Salakhutdinov 和 Sergey Levine。经验回放缓冲区上的搜索：连接规划与强化学习。载于 H. Wallach、H. Larochelle、A. Beygelzimer、F. d' Alch´e-Buc、E. Fox 和 R. Garnett（编），神经信息处理系统进展，第 32 卷。Curran Associates, Inc.，2019年。URL https://proceedings.neurips.cc/ paper/2019/file/5c48ff18e0a47baaf81d8b8ea51eec92-Paper.pdf。

Natalia Gardiol 和 Leslie Kaelbling。《关系型马尔可夫决策过程中的基于包络的规划》。载于 S. Thrun、L. Saul 和 B. Sch¨olkopf（编），神经信息处理系统进展，第 16 卷。MIT Press，2003年。URL https://proceedings.neurips.cc/paper/2003/file/ 4a06d868d044c50af0cf9bc82d2fc19f-Paper.pdf。

Tejas Gokhale、Shailaja Sampat、Zhiyuan Fang、Yezhou Yang 和 Chitta Baral。《用积木烹饪：基于图像对进行视觉推理的方案》。载于*IEEE/CVF*计算机视觉与模式识别会议研讨会论文集，第 5–8 页，2019年。

Anirudh Goyal、Alex Lamb、Phanideep Gampa、Philippe Beaudoin、Sergey Levine、Charles Blundell、Yoshua Bengio 和 Michael Mozer。《对象文件与图式：在动力系统中对陈述性知识和程序性知识进行因式分解》。*arXiv*预印本 *arXiv:2006.16225*，2020年。

Anirudh Goyal、Aniket Didolkar、Nan Rosemary Ke、Charles Blundell、Philippe Beaudoin、Nicolas Heess、Michael C Mozer 和 Yoshua Bengio。神经产生式系统。神经信息处理系统进展，34:25673–25687，2021年。

Klaus Greff、Sjoerd Van Steenkiste 和 J¨urgen Schmidhuber。神经期望最大化。*arXiv* 预印本 *arXiv:1708.03498*，2017。

Klaus Greff、Rapha¨el Lopez Kaufman、Rishabh Kabra、Nick Watters、Christopher Burgess、Daniel Zoran、Loic Matthey、Matthew Botvinick 和 Alexander Lerchner。利用迭代变分推断进行多对象表征学习。发表于 国际机器学习会议，第 2424–2433 页。PMLR，2019年。

Klaus Greff、Sjoerd van Steenkiste 和 J¨urgen Schmidhuber。关于人工神经网络中的绑定问题。*arXiv* 预印本 *arXiv:2012.05208*，2020年。C. Guestrin、D. Koller、R. Parr 和 S. Venkataraman。《因子化马尔可夫决策过程的高效求解算法》。《人工智能研究杂志》，19:399–468，2003年10月a。ISSN 1076-9757。DOI: 10.1613/jair.1000。URL https://jair.org/index.php/jair/article/view/ 10341。tex.ids: guestrin2003EfficientSolutionAlgorithms。Carlos Guestrin、Daphne Koller、Chris Gearhart 和 Neal Kanodia。在关系型马尔可夫决策过程中将规划泛化到新环境。在第*18*届国际人工智能联合会议论文集中，IJCAI'03，第1003–1010页，美国加利福尼亚州旧金山，2003年b。Morgan Kaufmann Publishers Inc. Philippe Hansen-Estruch、Amy Zhang、Ashvin Nair、Patrick Yin 和 Sergey Levine。双模拟使目标条件强化学习中的类比成为可能。*arXiv* 预印本 *arXiv:2204.13060*，2022年。De-An Huang、Suraj Nair、Danfei Xu、Yuke Zhu、Animesh Garg、Li Fei-Fei、Silvio Savarese 和 Juan Carlos Niebles。神经任务图：从单个视频演示中泛化到未见任务。在*IEEE/CVF*计算机视觉与模式识别会议论文集中，第8565–8574页，2019年。

Michael Janner、Justin Fu、Marvin Zhang 和 Sergey Levine。何时信任你的模型：基于模型的策略优化。神经信息处理系统进展，32，2019年。

Thomas Kipf、Gamaleldin F Elsayed、Aravindh Mahendran、Austin Stone、Sara Sabour、Georg Heigold、Rico Jonschkowski、Alexey Dosovitskiy 和 Klaus Greff。基于视频的条件对象中心学习。*arXiv* 预印本 *arXiv:2111.12594*，2021年。

Ilya Kostrikov。JAXRL：使用 JAX 实现强化学习算法，2021年10月。URL https://github.com/ikostrikov/jaxrl。

Tejas D Kulkarni、Ankush Gupta、Catalin Ionescu、Sebastian Borgeaud、Malcolm Reynolds、Andrew Zisserman 和 Volodymyr Mnih。用于感知和控制的物体关键点无监督学习。*NeurIPS*，2019年。

Sergey Levine、Aviral Kumar、George Tucker 和 Justin Fu。离线强化学习：教程、综述及开放问题展望。*arXiv* 预印本 *arXiv:2005.01643*，2020年。

Martina Lippi、Petra Poklukar、Michael C Welle、Anastasiia Varava、Hang Yin、Alessandro Marino 和 Danica Kragic。《用于可变形与刚性物体操作的视觉动作规划潜在空间路线图》。载于*2020* 年 *IEEE/RSJ*智能机器人与系统国际会议（*IROS*），第 5619–5626 页。IEEE，2020年。

Francesco Locatello、Dirk Weissenborn、Thomas Unterthiner、Aravindh Mahendran、Georg Heigold、Jakob Uszkoreit、Alexey Dosovitskiy 和 Thomas Kipf。《基于槽位注意力的对象中心学习》。*arXiv* 预印本 *arXiv:2006.15055*，2020a。

Francesco Locatello、Dirk Weissenborn、Thomas Unterthiner、Aravindh Mahendran、Georg Heigold、Jakob Uszkoreit、Alexey Dosovitskiy 和 Thomas Kipf。《基于槽位注意力的对象中心学习》。载于*NeurIPS*，2020 b。URL https://proceedings.neurips.cc/paper/2020/ hash/8511df98c02ab60aea1b2356c013bc0f-Abstract.html。

OpenAI。Robogym。https://github.com/openai/robogym，2020年。

Luis Pineda、Brandon Amos、Amy Zhang、Nathan O. Lambert 和 Roberto Calandra。Mbrl-lib：一个用于基于模型的强化学习的模块化库。*arXiv*，2021年。URL https://arxiv. org/abs/2104.10159。

Alec Radford、Karthik Narasimhan、Tim Salimans、Ilya Sutskever 等人。通过生成式预训练提升语言理解能力。2018年。

Aditya Ramesh、Mikhail Pavlov、Gabriel Goh、Scott Gray、Chelsea Voss、Alec Radford、Mark Chen 和 Ilya Sutskever。零样本文本到图像生成。*arXiv* 预印本 *arXiv:2102.12092*，2021年。

Gautam Singh、Fei Deng 和 Sungjin Ahn。文盲 DALL·E 学会了组合。*arXiv* 预印本 *arXiv:2110.11405*，2021年。

Gautam Singh、Fei Deng 和 Sungjin Ahn。《不识字的 DALL-E 学会组合》。发表于国际学习表征会议，2022a。URL https://openreview.net/ forum?id=h0OYV0We3oh。

Gautam Singh、Yi-Fu Wu 和 Sungjin Ahn。《面向复杂自然主义视频的简单无监督对象中心学习》。*arXiv*预印本 *arXiv:2205.14065*，2022年b。Emanuel Todorov、Tom Erez 和 Yuval Tassa。《MuJoCo：用于基于模型的控制的物理引擎》。载于*2012*年 *IEEE/RSJ*国际智能机器人与系统会议，第5026–5033页。IEEE，2012年。Laurens Van der Maaten 和 Geoffrey Hinton。《使用 t-SNE 可视化数据》。机器学习研究期刊，9(11)，2008年。Sjoerd Van Steenkiste、Michael Chang、Klaus Greff 和 J¨urgen Schmidhuber。《关系神经期望最大化：对象及其交互的无监督发现》。*arXiv*预印本 *arXiv:1802.10353*，2018年。

Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。《Attention is all you need》。神经信息处理系统进展，30，2017。

Rishi Veerapaneni、John D Co-Reyes、Michael Chang、Michael Janner、Chelsea Finn、Jiajun Wu、Joshua Tenenbaum 和 Sergey Levine。视觉模型化强化学习中的实体抽象。载于机器人学习大会，第 1439–1456 页。PMLR，2020年。

C. Wang、S. Joshi 和 R. Khardon。用于关系型马尔可夫决策过程的一阶决策图。人工智能研究期刊，31：431–472，2008年3月。ISSN 1076-9757。DOI：10.1613/jair.2489。URL：http://dx.doi.org/10.1613/jair.2489。

Daniel S Weld。使用一阶机器学习求解关系型马尔可夫决策过程。第 8 页，2003年。

Ge Yang、Amy Zhang、Ari S. Morcos、Joelle Pineau、Pieter Abbeel 和 Roberto Calandra。Plan2vec：通过潜在计划进行无监督表征学习。载于第 *2* 届动力学与控制学习年度会议论文集，第 120 卷，机器学习研究论文集，第 1–12 页，2020年。

Shuo Yang、WeiZhang、Ran Song、Jiyu Cheng 和 Yibin Li。学习用于自主目标条件抓取的多物体稠密描述符。*IEEE*机器人与自动化快报，6(2):4109– 4116，2021年。

Andrii Zadaianchuk、Georg Martius 和 Fanny Yang。利用可独立控制的子目标进行自监督强化学习。载于机器人学习大会，第384–394页。PMLR，2022年。

Amy Zhang、Adam Lerer、Sainbayar Sukhbaatar、Rob Fergus 和 Arthur Szlam。基于属性的可组合规划。载于第*35*届国际机器学习大会论文集，*ICML 2018*，瑞典斯德哥尔摩¨，斯德哥尔摩，瑞典，*2018*年*7*月*10*日至*15*日，第80卷，第5837– 5846页。JMLR.org，2018年。

Lunjun Zhang、Ge Yang 和 Bradly C Stadie。《作为图的世界模型：学习用于规划的潜在地标》。载于 Marina Meila 和 Tong Zhang（编），第*38*届国际机器学习大会论文集，机器学习研究论文集第139卷，第12611–12620页。PMLR，2021年7月18日至24日。URL https://proceedings.mlr.press/ v139/zhang21x.html。

Daniel Zoran、RishabhKabra、Alexander Lerchner 和 Danilo J Rezende。《Parts：利用槽位、注意力和独立性最大化进行无监督分割》。载于*IEEE/CVF* 国际计算机视觉会议论文集，第 10439–10447 页，2021年。

A 实现细节

本节详细介绍 NCS 各组件的实现设计决策。dSLATE 的超参数见表 2。

A.1 背景：SLATE 骨干网络

SLATE（Singh 等人，2022a）是一种自编码器架构，它使用槽注意力（SA）（Locatello 等人，2020b）作为瓶颈。它先通过离散变分自编码器（Ramesh 等人，2021）对图像进行预处理，将其转换为图像特征网格；再将这些特征编码为令牌网格，并通过 SA 从该令牌网格中推断槽位。SA 还会生成注意力掩码，指示每个槽位关注哪些特征。训练这些槽位时使用 Transformer 解码器（Vaswani 等人，2017；Radford 等人，2018），并以槽位作为键/值，自回归地重建令牌。

轮次数		200
每轮回合数		5K
回合长度		5
批量大小		32
峰值学习率		0.0002
学习率预热步数		30000
丢弃率		0.1
离散变分自编码器	词汇表大小	4096
	温度冷却	1.0 到 0.1
	温度冷却步数	30000
	学习率（无预热）	0.0003
	图像尺寸	64
	图像令牌	图像尺寸 / 4
Transformer 解码器	层数	4
	头数	4
	隐藏维度	192
槽注意力	槽位	5
	迭代次数	3
	槽头数	1
	槽维度（h）	192
	类型维度（λ^z）	96
	状态维度（λ^s）	96
Transformer 动力学	层数	4
	头数	4
	隐藏维度	96

表 2：**训练 dSLATE 的超参数**这些超参数与 Singh 等人（2022a，图 7）中的设置几乎完全相同，但由于 dSLATE 处理的是视频演示而非静态图像，我们调整了部分超参数以节省内存。我们将批量大小从 50 调整为 32，将 Transformer 层数和头数从 8 调整为 4，并将槽注意力迭代次数从 7 调整为 3，且未观察到性能显著变化。由于经验缓冲区中的每段视频都包含四个物体，我们按照 Van Steenkiste 等人（2018年）和 Veerapaneni 等人（2020年）采用的惯例，使用了五个槽位，比物体数量多一个。

A.2 通过聚类 ST 构建节点 ATES

对于*block-rearrange*，我们发现，使用 SA注意力掩码 α 作为状态 s时，得到的聚类效果更好。对于*robogym-rearrange*，我们发现，使用 SA 槽位 $\boldsymbol{\lambda}^s$ 的动作相关部分作为状态 s时，得到的聚类效果更好。我们还通过实证发现，用于 K-means 聚类和绑定（通过最近邻实现）的距离度量，其某些选择取决于所使用的状态表示；相关总结见表 3。K-means 的实现改编自 https://github.com/overshiki/ kmeans_pytorch。

将训练好的 dSLATE 应用于经验缓冲区以构建图时，我们发现，增加 SA 迭代次数可以改善实体表征。因此，尽管我们训练时使用了

三次槽注意力迭代的 dSLATE，但构建图时使用了七次迭代。最后，我们发现，K 均值所用的簇数是构建能够反映状态转移的图时最重要的超参数。我们将簇数从 16 扫描到 50，并在表4中报告了找到的最优簇数。

状态表示	α	λ^s
isolate 距离度量	余弦	余弦
簇距离度量	IoU	平方欧几里得
绑定距离度量	余弦	平方欧几里得

表 3：**使用 NCS 构建转移图的超参数**。此表展示了我们在 4.1 节所述的 isolate、cluster 和绑定函数中使用的距离度量。对于 *block-rearrange* ，我们使用 SA 注意力掩码 α 作为状态 s；对于 *robogym-rearrange* ，我们使用 SA 槽位 $\boldsymbol{\lambda}^s$ 的动作相关部分作为状态 s。

	block-rearrange	*robogym-rearrange*	堆叠方块
簇的数量	30	45	47

表4：**用于构建转移图节点的簇数量**。

A.3 动作选择

为实现对齐，我们使用匈牙利算法的 scipy.optimize.linear sum assignment 实现，并以各个 z^k之间的欧几里得距离作为匹配代价。
给定当前实体集合 $\mathbf{h}_t$ 和目标约束 $\mathbf{h}g$，选择约束会返回索引 k ，表示下一个要满足的目标约束。根据 NCS 的构造，绑定到 h_t^k 和 h_g^k 的节点之间的边表示：如果在环境中执行与该边关联的动作，就会执行相应的状态转移。如果 NCS 未能找到这两个节点之间的边，例如 h_t^k 和 h_g^k 被错误地绑定到图中，那么 NCS 就会直接采取随机动作。texttt选择约束包含两个步骤：(1) 状态转移排序；(2) 状态转移采样。

排序 排序步骤的目标是对 (h_g^1、.....、h_g^K) 中的索引进行排序，以选择实际要对哪个索引 k 采取动作。直观而言，我们应根据 s_t^k 和 s_g^k 之间的差异来排列索引 k，因为差异较大意味着约束 h_g^k 尚未满足，这表示我们需要采取动作来移动由 h_t^k 表示的相应物体。我们沿用用于 isolate 的距离度量 $d(\cdot,\ \cdot)$ 来实现这一排序。

采样根据我们的排序，采样步骤的目标是选取一个 $k \in \{1, ..., K\}$，并通过动作影响与其关联的实体。一种方法是像 isolate 中那样，直接选择 k ，使其为 $k = \arg\max_{k' \in \{1,...,\tilde{K}\}} d(s_t^{k'},\ s_{t+1}^{k'})$。但我们通过实验证实，从一个预归一化概率由 $d(s_t^{k'},\ s_{t+1}^{k'})$ 给出的分类分布中对 k进行采样，能获得更好的任务性能，因此采用了这种随机采样方法。使用 argmax 可能表现较差的一种解释是，这种方法依赖于距离度量 $d(\cdot,\ \cdot)$ 和状态表示 s，而它们必须能够使距离度量准确无误地为需要移动的实体 k 赋予较高值，并为无需移动的实体 k 赋予较低值。但是，由于状态空间 $\mathcal{S}$是在 dSLATE 训练过程中学习得到的，且没有针对空间几何结构的显式监督，因此本应比另一组点距离更远的一对点，未必能通过固定的距离度量 $d(\cdot,\ \cdot)$ 准确体现出来。未来的工作将探讨如何对 $\mathcal{S}$的几何结构施加显式监督。

B 环境详情

环境*Block-rearrange* 在 PyBullet（Coumans & Bai，2016年）中实现，而*robogym-rearrange* 则在 MuJoCo（Todorov 等，2012年）中实现。
Robogym-rearrange （见图 7 和图 8）改编自 OpenAI 的 Robogym 仿真框架中的 rearrange 环境（OpenAI，2020年），并移除了*block-rearrange*中的假设。

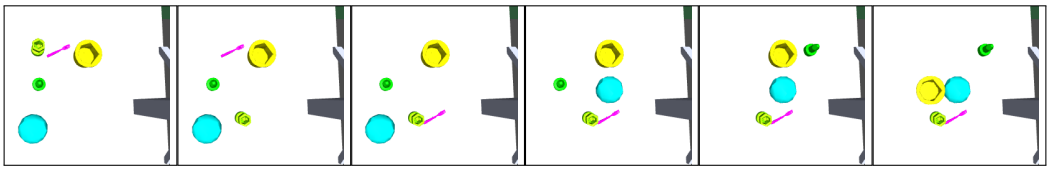


图 7：在本文使用的 robogym rearrange 环境中完成任务的示例。

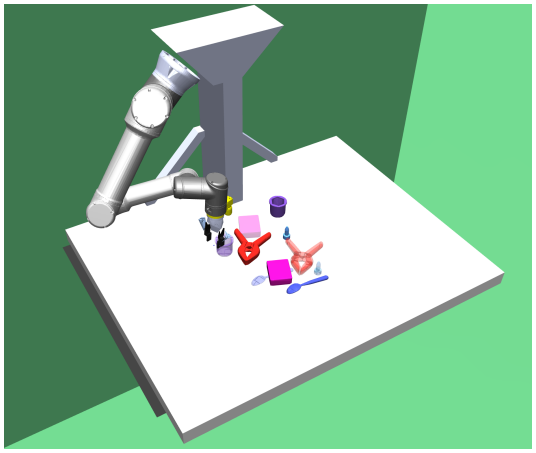


图 8：原始 Robogym rearrange 设置

即所有物体的尺寸、形状和朝向都相同，以及位置预先设定这一假设。此外，由于三维透视，物体在不同位置看起来可能略有不同。物体从由 94 个网格模型组成的集合中均匀采样，该集合包括 YCB 物体集 Calli 等人（2015年）以及一组基本几何形状；颜色则从 13 种颜色中采样。摄像机以俯视桌面的鸟瞰视角拍摄，每个物体的尺寸都按其最长维度进行归一化，因此又高又细的物体看起来会更小。物体的目标位置是随机采样的，确保它们彼此之间以及与所有初始位置均不重叠，并且目标朝向保持不变。由于位置取连续值，我们将初始抓取位置和目标放置位置的匹配阈值均设为不超过 0.05（桌面尺寸为 0.6 × 0.8）。

感觉运动接口 每个观察都是一个元组，包含一张显示当前观测的初始图像，以及一张显示待满足约束的目标图像——即物体的目标位置。每个动作都是一个元组（w，Δw），其中 w 是环境场地中的一个三维笛卡尔坐标（x，y，z）。物体会被初始化在随机且互不重叠的位置，并且这些位置也不会与其目标位置重叠。对于这些任务，z（高度）坐标始终固定。当 w 与物体位置之间的距离小于某个阈值时，该物体会被抓取。对于 *block-rearrange* 任务，物体位置是网格中的不动点，物体会被吸附到距离 $w + \Delta w$ 最近的网格位置。当物体被放置在目标位置的某个阈值范围内时，即认为约束已满足。

C 基线实现细节

随机（Rand）随机策略使用 env.action space.sample() 执行动作 _ .

行为克隆（BC）这种方法训练策略，使其直接输出所提供数据集中采取的动作。我们使用均方误差损失训练策略来模仿这些动作。

隐式 Q 学习（IQL）是一种简单的离线强化学习方法，它使用数据集中的动作进行时序差分（TD）学习，并训练行为策略价值函数。为了得到最优价值函数，IQL 使用期望分位数回归估计 Q 函数的最大值

采用非对称均方误差，优化目标如下：

$$L_V(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}[L_2^\tau(Q_{\hat{\theta}}(s,a) - V_\psi(s))] \text{ where } L_2^\tau(u) = |\tau - \mathbb{1}(u<0)|u^2 \quad (1)$$

$$L_Q(\theta) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}}[(r(s,a) + \gamma V_\psi(s') - Q_\theta(s,a))^2] \quad (2)$$

$$L_\pi(\phi) = \mathbb{E}_{(s,a)\sim\mathcal{D}}[\exp\left(\beta(Q_{\hat{\theta}}(s,a) - V_\psi(s))\right)\log\pi_\phi(a|s)]. \quad (3)$$

The $V(s)$估计值用于时序差分备份，并通过优势 -
wei加权行为克隆。

模型预测控制 (MPC) 该方法使用模型预测控制与交叉熵方法（交叉熵方法）来选择动作，并利用 dSLATE 的 Transformer动力学模型在潜在空间中进行展开轨迹。这与 OP3（Veerapaneni 等，2020）中使用的方法类似，不同之处在于我们采用了近期提出的架构组件（槽注意力（Locatello 等人，2020b）而非 IODINE（Greff 等人，2019），以及 Transformer 而非图网络（Battaglia 等人，2018；Van Steenkiste 等人，2018；Chang 等人，2016）），因此我们的 MPC 结果不能与 OP3 的结果直接比较。我们使用的 dSLATE 检查点与 NCS 所用的相同。
我们使用 Mbrl-lib 库（Pineda 等，2021年）实现此 MPC 基线，采用 10 次交叉熵方法迭代、0.05 的精英比例和 250 的种群规模。这是我们找到的最佳配置，可在 8 个物体和 100 个测试回合的设置下，将运行时间控制在两天以内。我们遍历了交叉熵方法迭代次数 $[5, 10, 20]$, 、精英比例 $[0.05, 0.1, 0.2]$ 和种群规模 $[250, 500, 1000]$,，并发现精英比例是最重要的超参数。
代价函数的计算方法是：首先使用附录 A.3 中相同的对齐流程，对齐预测的槽位 $\mathbf{h}_T$ 和目标约束$\mathbf{h}^g$，然后将槽位之间的平方欧几里得距离相加，得到 $cost = \sum_k (h_T^k - h_g^k)^2$。

非因子化图搜索（NF）此方法是对 NCS 的消融实验：它不针对各个实体的状态转移构建图，而是针对实体集合的状态转移构建图，也就是说，每次转移为 $(\mathbf{s}\ a, \mathbf{s}')$，而不是 $(s^k, a, s^{k\prime})$。与 MPC 一样，我们使用了与 NCS 相同的 dSLATE 检查点。
这项消融实验旨在阐明，与实体集合相比，将转移图按单个实体因式分解的优势。由于 NF 的转移图中的节点表示一组实体状态，而非单个实体状态，我们采用 Dijkstra算法（如 Eysenbach 等人（2019年）、Yang 等人（2020年）和 Zhang 等人（2018年）所用）来规划一条从初始观察所绑定的节点到目标观察所绑定的节点的无断路径。每个时间步，我们都使用 Dijkstra算法沿节点规划一条路径，然后返回该路径上第一条边所关联的动作。与 NCS 一样，NF 是一种非参数模型，这意味着要将一组实体绑定到图中的某个节点，该节点必须包含与这些实体状态相对应的确切实体状态集合。如果未能成功绑定到图，或者找不到当前节点与目标节点之间的路径，我们就像 NCS 一样采样一个随机动作。

D 补充结果

本节介绍 NCS 的补充结果与分析。

D.1 关键超参数分析

本节分析在构建图时，NCS 中使用的若干超参数对任务性能的影响：簇的数量、从经验缓冲区中取用的样本数量，以及槽注意力中使用的槽位数量。我们在 robogym 环境中进行评估，完整目标规格包含四个物体。如图 9 所示，性能取决于初始化簇的数量，以及用于构建图的训练集批次数量。簇数量过少时，簇的粒度过粗，无法区分位置差异显著的物体。簇数量过多时，数据被不必要地拆分为重复簇，导致性能下降。数据越多，性能越好，因为图的覆盖范围更广。尽管当槽位不足以表示环境中的所有物体时，NCS 的性能会变差，但槽位数量达到所需数量的两倍时，性能几乎不受影响。因此，即使环境中物体数量未知、但有上限，我们的方法仍然适用。

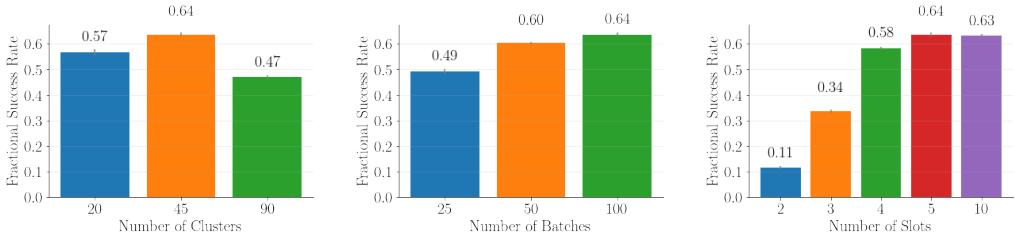
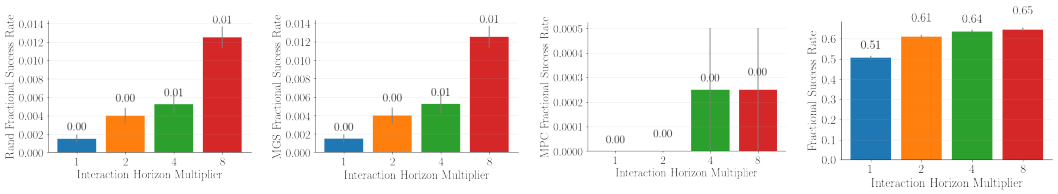


图 9：当用于构建图的初始化簇数量和训练集批次数量，以及槽位数量发生变化时，我们的方法的性能。

D.2 增加基于模型的基线方法的计算时间

我们测试了将基于模型的基线方法的计算时间加倍，是否能使其性能提升至与 NCS 相当。在论文主体的结果中，我们将回合长度限制为解决任务所需最少动作数的 4 倍。在图 10 中，我们将这一交互时域倍数从 1 倍调整到 8 倍。与基线方法相比，NCS 在交互时域较短时性能下降较少。我们发现，NF 的表现与随机基线相近。由于 NF 在无法将给定实体集合绑定到其图中时会采取随机动作，这一结果表明，实体子集的组合空间非常庞大，以至于 NF 大多数时候都无法成功将其绑定到图中。通过检查 NF 采取随机动作的情况，我们验证了这一点。在所有方法中，MPC 表现最差，甚至不如随机方法。我们测试发现，附录 C 中描述的代价函数会为符合目标约束的潜变量分配比随机采样潜变量更低的代价，这表明误差的主要来源是预测展开轨迹不准确。出现这种情况并不意外，因为学习得到的模型在展开时会产生累积误差（Janner 等，2019），而此前将 MPC 用于以物体为中心的方法仅进行非常短时域的展开（Veerapaneni 等，2020）。



(a) 随机　(b) NF　(c) MPC　(d) NCS

图 10：**改变交互时域**。随着我们改变交互时域（以完成任务所需的最少步数的倍数表示），NF（b）和 MPC（c）基线方法在*robogym-rearrange* 上的性能，与 NCS（d，转载自图 11）和随机基线方法（a）进行比较。请注意，y 轴的刻度并不相同。较长的时域能够提升性能，但当交互时域倍数为 1 时，NCS 的准确率仍至少比将基于模型的基线方法的交互时域倍数提高到 8 时所获得的性能高出 50 倍。

D.3 更具挑战性的设置

最后，我们在更具挑战性的设置下分析了 NCS，这些设置粗略模拟了现实世界机器人学中噪声干扰的特性。如图 11（左）所示，在每个时间步的动作中加入高斯噪声时，只要噪声方差不超过成功抓取和放置到目标位置所允许的最大距离，NCS 就比基线方法更加稳健。即使交互步数显著减少，其性能仍然很高（图 11，右）。然而，我们的成功率仍远未达到完美，这表明要将 NCS 扩展到现实世界，还有大量工作要做。

组合空间

本节详细说明 § 5 中所述任务空间组合规模的计算。初始状态中的物体配置数量为 $\binom{|S|}{k}$。在完整规格设定中，必须移动所有物体，因此 $t \geq k$。每一步都可以将 k 个已占用网格单元中的任意一个移动到 $\binom{|S|}{k}$ 个未占用网格单元中的任意一个，因此后继状态的数量为 $k \times (|S| - k)$。对于 $|S|$ 个物体位置和 k 个物体，经过 t 个时间步的物体配置轨迹数量为

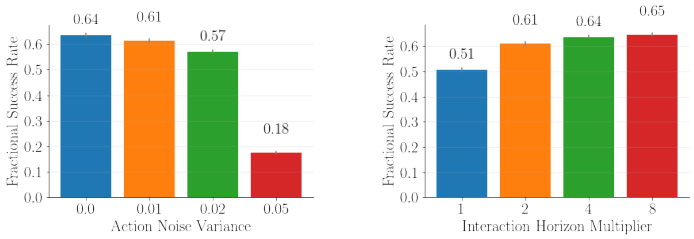


图 11：**NCS 压力测试** 此图展示了在不同动作噪声水平（左）以及不同交互时域（右）下 NCS 在 *robogym-rearrange* 上的性能。交互时域定义为完成任务所需最少步数的倍数。

为 $\binom{|S|}{k} \times (k \times (|S| - k))^t$。对于 *block-rearrange* $|S| = 16$，因此 $k = 7$ 下可能的轨迹数量为 $\geq 4.5 \times 10^{16}$。

F 局限性与未来工作。

NCS 采用非参数、非学习式控制方法，以凸显我们对组合任务空间的表示所具备的泛化能力，但这使 NCS 只能组合此前见过的实体之间已有的转移。沿状态转移压缩组合空间已带来显著收益，但无法适应测试时引入新物体的情况。目前，NCS 使用 SLATE 和 K 均值等工具实现，这些工具仍有很大的改进空间。我们预计，未来版本的 NCS 将以其后续改进版本取代 SLATE 和 K 均值，从而进一步提升我们的结果。

除了改进以物体为中心的模型、使其能够稳健地对真实像素进行建模这一挑战之外，要将我们的方法扩展到 Gokhale 等（2019）和 Chang 等（2020）所研究的真实世界环境，还需克服另一项挑战：将高层的抓取与移动动作原语转换为真实机器人的电机扭矩，同时处理不同物体的几何形状、质量和属性。鉴于许多机器人学习领域的研究（例如 Devin 等（2020）；Yang 等（2021））都在解决目标条件化的物体抓取与操作这一问题，将此类目标条件化策略训练为供 NCS 组合使用的抓取与移动原语，或许是将我们的方法扩展到真实世界环境的一种途径。

本文假设物体可以彼此独立地移动。初步实验表明，NCS 可以扩展以支持堆叠方块等涉及物体间依赖关系的任务，但如何处理这些依赖关系值得在未来工作中单独探讨。

G 为什么称为“神经约束满足”?

NCS 可以被视为通过物理方式求解一个具身约束满足问题，其中状态是变量，身份是变量值，而动作则执行变量赋值。与符号约束满足不同，这些变量、变量的取值域、赋值算子和约束都从感觉运动接口中学习而来，因此得名“神经约束满足”。