arXiv:2303.00323v2 [cs.RO] 9 Jan 2024

DeFNet：面向多步织物折叠任务的解构策略

Ningquan Gu[1],Ruhan He[1*] 和 Lianqing Yu[2]

[1]武汉纺织大学计算机科学与人工智能学院，中国[2]武汉纺织大学机械工程与自动化学院，中国

文章历史
编于2024年1月10日

摘要
机器人执行织物折叠任务复杂且具有挑战性，这是由织物的可变形性所致。基于解构策略，我们将复杂的多步织物折叠任务拆分为三个相对简单的子任务，并提出解构式织物折叠网络（DeFNet），其中包含三个相应模块来解决这些任务。(1) 我们采用基于潜在空间路线图的折叠规划模块（FPM），在潜在空间中推断从起始状态到目标状态的最短折叠中间状态。(2) 我们采用基于流的方法——折叠动作模块（FAM）——计算动作坐标并执行相应动作，以到达推断出的中间状态。(3) 对于多步织物折叠任务，我们引入迭代交互模块（IIM），使其在每次抓取放置动作后迭代执行 FPM 和 FAM，直至织物达到目标状态。通过实验，我们在仿真中将该方法应用于织物折叠任务，并与三个基线方法进行了比较。我们还将该方法应用于现有的机器人系统，并展示了其性能。

关键词
织物折叠；解构策略；潜在空间；基于流的方法；迭代；

1. 引言

织物操作在日常生活和工业中有许多应用。然而，机器人要实现从一堆衣物中挑出目标织物、展开毛巾以及折叠或抚平衣物等功能，过程复杂且具有挑战性 [1–4]。在抓取与放置过程中，可变形构型中的自遮挡 [5] 会导致只能部分观测 [6], ，这使得操作复杂度高于刚体。

早期的织物折叠或抚平方法主要依赖非学习方法和预设动作 [1,7]。然而，这些策略通常速度较慢，也缺乏针对任意初始状态与目标状态进行泛化的能力。近年来，研究人员探索了基于学习的织物操作方法 [4,8–10], ，其效果优于传统的非学习方法。尤其是，利用低维潜在空间表示进行规划的做法在织物操作任务中日益普及，因为这类表示能够应对状态空间的复杂动态，从而将原本难以处理的状态纳入考量。例如，Lippi 等人 [11,12] 提出了视觉动作规划（VAP）方法，该方法利用潜在空间路标图（LSR）按顺序生成一系列以视觉方式呈现计划的图像，并采用动作提议模块（APM）预测图像之间的动作，应用于箱子堆叠和T恤操作任务。

*联系人：Ruhan He。电子邮箱：heruhan@wtu.edu.cn

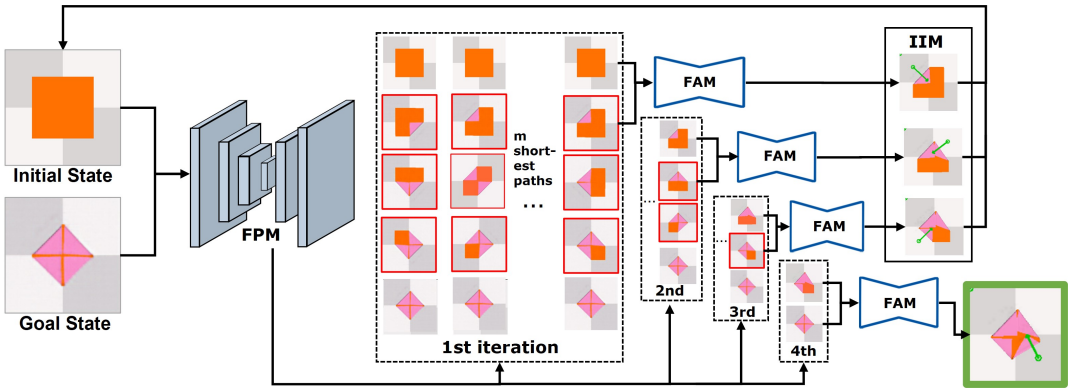


图1. 我们提出的织物折叠方法将任务拆解为三个模块。首先，折叠规划模块（FPM）根据初始状态和目标状态，对织物的中间折叠状态进行推理。FPM 会生成一系列带红色边框的图像，表示推断出的中间状态。对于多重折叠任务，通常存在多条最短折叠路径。其次，折叠动作模块（FAM）采用基于流的方法。我们随机选择一条最短路径，并将对应的图像输入 FAM，以获得中间折叠状态之间的抓取与放置动作。最后，迭代交互模块（IIM）执行每个抓取与放置动作（图像中以绿色箭头标示），并将织物的当前观测作为新的起始状态，再次输入 FPM。每次执行抓取与放置动作后，IIM 都会迭代执行前两个模块，直至到达目标状态（带绿色边框的图像）。通过运用这三个模块，我们的方法能够高效且有效地解决复杂的织物折叠任务。

并在箱子堆叠和T恤操作任务中，采用动作提议模块（APM）预测这些图像之间的动作。然而，他们的动作提议策略会同时直接输出抓取点和放置点坐标。由于放置点在很大程度上取决于抓取点，因此难以捕捉动作空间内在的复杂性和相互依赖关系 [13]。此外，Weng 等人[14] 将 FlowNet 应用于织物操作任务，使其能够推理织物的流动情况，并获得更精确的抓取与放置点。他们的方法表现显著优于先进的无模型和基于模型的织物操作策略。不过，该策略的一个局限是，它要求了解多步骤任务中的每个子目标，而在某些特定场景中，这一点可能难以做到。

为应对织物折叠任务的复杂性，我们提出了 Deconstructed FabricFoldingNetwork（DeFNet），采用一种分解策略，将任务拆分为三个相对简单的子任务。DeFNet 的框架如图1所示。

DeFNet 由三个模块组成。第一个模块是折叠规划模块（FPM），其在潜在空间中运行以确定织物折叠路径。FPM 以初始状态和目标状态为输入，生成一个直接的中间状态序列。我们在 FPM 中采用潜在空间路标图（LSR） [12] 作为规划策略。第二个模块是折叠动作模块（FAM），它采用光流策略计算每对中间织物状态之间的抓取与放置点。该模块能够精确确定折叠过程所需的动作。第三个模块是迭代交互模块（IIM），其在增强执行过程中发挥着关键作用。每次抓取放置动作后，IIM 都将当前织物观测作为新的起始状态，并将其反馈给 FPM。通过在每次动作后迭代执行第一和第二个模块，IIM 可有效减少执行中的潜在不确定性，

过程，并促进更可靠的感知与规划。在我们的实验中，我们的方法优于最先进的织物操作策略。

本文的贡献可总结如下：

- 基于解构策略，我们提出了一种新颖的框架 DeFNet，将复杂的折叠任务拆分为三个相对简单的子任务，从而能够提高长时序折叠任务中每个折叠步骤的准确率，并提升整体准确率。
- 我们为多步织物折叠任务设计了一个迭代交互模块，将 FPM 和 FAM 完美集成到框架中。同时，该迭代模块对于提高每个步骤的准确率非常有益。
- 我们通过仿真实验将 DeFNet 与三种当前最先进的方法进行比较，并开展了消融实验，结果表明 DeFNet 的性能最佳。

2. 相关工作

2.1. 织物操作学习

由于织物的高维空间，机器人操作织物是一项具有挑战性的任务，而且很难表示织物的状态。织物操作领域中仍有若干尚未解决或优化不足的任务，包括织物仿真 [15], 织物平滑 [16], 铺床 [4],和织物折叠 [17]。许多以往的研究使用传统计算机视觉算法描述织物，并通过预设轨迹对其进行操作。然而，这些方法在处理织物的复杂构型时存在局限，可能无法有效解决长时序的连续折叠任务。例如，传统计算机视觉算法，包括使用 Harris 角点检测和不连续性来获取峰值脊线和峰值角点 [18],、使用 RANSAC 定位织物的角 [1], ，甚至提起布料并将其抛起以获得新的目标形状 [19], ，在织物构型复杂时往往会失败。采用学习方法，尤其是深度学习网络或深度强化学习，为织物操作任务提供了更多选择。例如，Lee等人 [20] 通过收集真实机器人与织物交互的数据集，学习了一种用于折叠织物的无模型算法。然而，其性能受离散动作空间限制，而且不太擅长解决长时序的连续折叠任务。类似地，Hoque等人 [21] 学习了视觉动力学模型并使用交叉熵方法进行规划，可用于解决任意目标和长时序任务，但其方法在织物操作任务中需要大量抓取与放置动作，可能需要多次动作才能折叠一个角。相比之下，我们的框架能够处理长时序的连续折叠任务，并推断具有最短折叠路径的中间折叠状态。

2.2. 潜在空间路标图

对于系统状态无法通过解析方式确定的复杂场景，规划方法可根据规划执行的位置归纳为两种主要策略。第一种是在高维图像空间中进行规划。Finn等人使用长短期记忆模块预测帧与帧之间的随机像素流。Hoque等人于2021年提出了VisuoSpatial Foresight，这是一种预测织物运动的方法。该方法基于先前的研究，这些研究涉及

同时学习领域随机化RGB图像和深度信息的视觉动力学。具体而言，Nair等人利用五百小时的数据来学习绳索的逆动力学。另一种方法是在低维潜在状态空间中进行规划，从而降低输入图像空间的复杂度。2020年，Lippi 等人提出并实现了视觉动作规划（VAP），这是一种能够全局捕捉潜在系统动力学的基于图的结构。它通过VAP的潜在空间路线图（LSR）模块预测刚体和可变形物体的状态。2022年，Lippi 等人改进了LSR模块的性能，包括改进LSR构建算法的扩展方式，并加入关键超参数的动态调整。不过，他们的动作提议策略会同时直接输出抓取位置和放置位置。由于放置点很大程度上取决于抓取点，因此难以捕捉动作空间内在的复杂性和相互依赖关系。与以往的表示方法不同，我们采用基于流的方法；事实证明，该方法在基于目标的织物操作方面显著优于以往方法。

2.3. 用于策略学习的光流

估计两幅图像之间的逐像素对应关系，可以获得每个像素的光流。2015年，Dosovitskiy 等人 [24]将 FlowNet 引入视频领域，使其能够每秒预测十对图像的光流，在实时方法中取得了出色的准确度。2022年，Weng 等人 [14] 将 FlowNet 应用于织物操作，通过推理操作过程中织物的光流，获得了更精确的抓取与放置点；其性能显著优于最先进的织物操作策略。然而，Weng 等人的策略 FabricFlowNet（FFN）必须了解织物的每个中间折叠子状态。在多步折叠任务中，很多时候我们通常只有初始状态和目标状态。相比之下，我们的方法所需的输入更少，只需初始状态和目标状态，无需输入子目标。

3. 问题陈述

我们的目标是使机器人能够执行多步折叠任务。任务可以表述如下：给定织物的起始状态和目标状态，经过一系列动力学有效动作，即可得到最终状态。

问题形式化如下：输入是一个元组 $\langle I_{\text{start}}, I_{\text{goal}}\rangle$。 I_{start}是展平的织物。 I_{goal} 表示我们希望达到的折叠目标状态。

$a_i \in \mathcal{U}$ 表示机器人执行的动作。 $\mathcal{U}$ 表示动作空间。当执行动作 a_i时， $a_i \in \mathcal{A}$ 织物可以将当前状态I_{i}改变为下一个状态 $I_{\text{i+1}}$。经过一系列动作 $\mathcal{A}$： $\langle a_0, a_1, \cdots, a_n\rangle$，我们可以获得织物的最终状态 I_{final}。

$$(\mathcal{A}, I_{final}) = \pi(I_{\text{start}}, I_{\text{goal}}) \tag{1}$$

我们希望学习 π，以最小化动作数量以及 I_{goal} 与 I_{final}之间的差异。

4. 方法

4.1. 方法概述

我们提出了一种新颖的策略，即将折叠任务分解。如图1所示，我们可以将任务拆分为三个相对简单的子任务，从而提升性能：首先，分析起始状态与目标状态之间的联系，以获得织物最短的中间折叠状态。其次，计算初始状态与预测的中间折叠状态之间的抓取与放置动作点。第三，每执行一次折叠动作后，我们便将织物的当前观测作为新的起始状态，反复输入第一个模块。

我们将策略分解为：

$$\pi = \pi_{\mathrm{p}} \cdot \pi_{\mathrm{a}} \tag{2}$$

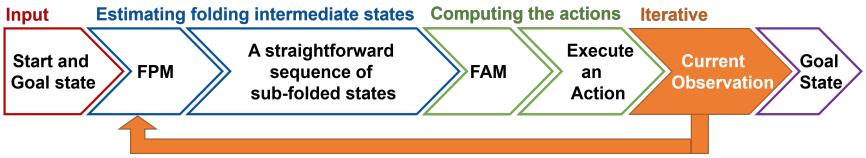


图2。 我们的方法流程

图2展示了我们方法的整体流程。在织物折叠任务中，我们的框架接收一对元组（红色箭头框），$\langle I_{\mathrm{start}},\ I_{\mathrm{goal}}\rangle$。FPM（蓝色箭头框）利用低维空间预测中间折叠状态，生成由子步骤$\mathcal{G}$组成的直接序列：$\langle I_0 = \hat{}\ \hat{}\ \hat{}I_{\mathrm{start}},\ I_1,\ I_2 \cdots I_{\mathrm{n\text{-}1}},\ I_{\mathrm{n}} = I_{\mathrm{goal}}\rangle$。从起点到目标的路径是最短的，这意味着如果我们想要得到目标形状，只需要执行n次折叠动作。可能存在多条最短折叠路径，我们可以随机选择其中一条，而不会影响结果。接下来，我们根据FPM的输出，使用FAM计算抓取与放置点。每次折叠动作后，织物都会进入一个新状态；随后，我们将当前观测迭代输入FPM，直到达到目标状态。

4.2. 折叠规划模块

折叠规划模块（FPM）学习一种策略π_{p}，用于推断从 I_{start} 到 I_{goal}的最短中间折叠状态 $\mathcal{G}$，目标是尽量减少所需的折叠步骤数。

$$\mathcal{G} = \pi_{\mathrm{p}}\left(I_{\mathrm{start}}, I_{\mathrm{goal}}\right) \tag{3}$$

我们参考 Lippi 等人的部分工作实现折叠规划模块，该模块由两个组件组成：变分自编码器（VAE） [25], 和潜在状态路线图（LSR） [12]。VAE 将以高维状态形式提供的观测映射到潜在空间中的结构化低维状态，从而提取相应状态；它也可以根据潜在状态生成观测。LSR 构建于 VAE 得到的潜在空间中 $\mathcal{Z}$。LSR 会构建并连接相似状态的簇，以确定潜在规划路径

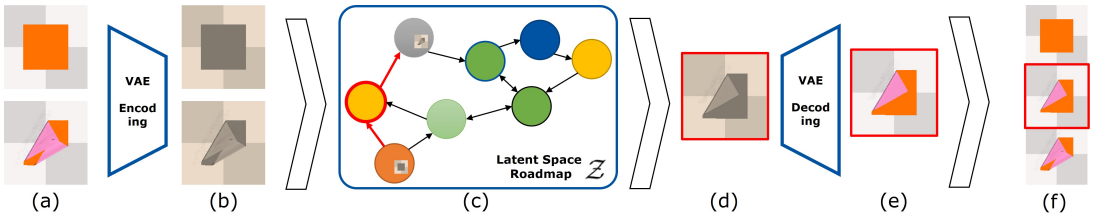


图 3。 折叠规划过程由多个步骤组成。首先，变分自编码器（VAE）将输入图像（a）转换为低维潜在表示（b）。在这一低维空间中，操作规划在构建的潜在空间路标图（LSR）（c）中执行，以确定到达目标状态的最优中间折叠状态序列。最后，利用 VAE 将预测结果（d）转换回高维图像空间（e）。将高维预测结果与初始状态和目标状态相结合，即可获得完整的最短折叠规划步骤。

这些路径位于起始状态与目标状态之间，而这两个状态均由 VAE 提取。这样便能够确定提取出的状态之间的潜在规划 z_{start} 和z_{goal}。最后，LSR 用于检索这两个已确定状态之间的所有最短路径，从而确定到达目标状态所需的最短中间折叠状态。

执行折叠规划模块时，我们以初始状态和目标状态、训练好的 VAE 以及构建好的 LSR 作为输入。如图 3 所示，初始状态（a）首先由 VAE 编码为低维潜在表示，如（b）中的棕色图像所示。编码后的信息随后映射到 LSR 中，以找到对应的节点。接下来，我们确定这些节点之间 LSR 中的所有最短路径，如（c）中的红线所示。在这些路径中，我们选取中间节点，该节点表示为带红色圆圈的橙色节点。中间节点的潜在信息如（d）所示。最后，VAE 对（d）中的潜在信息进行解码，生成（e）中的视觉图像。将规划得到的中间折叠状态与起始状态和目标状态相结合，即可获得完整的最短折叠规划步骤。为了通过抓取与放置确定如何完成中间折叠子步骤，我们使用基于流的策略计算抓取与放置点，具体如第 4.3 节所述。该策略通过确定织物在折叠过程每一步中的最优抓取与放置点，使机器人能够执行规划好的折叠子步骤。

折叠规划模块（FPM）的损失函数受两个因素影响：变分自编码器（VAE）损失 $\mathcal{L}_{\text{vae}}$ 和动作项 $\mathcal{L}_{\text{action}}$，如 [12] 所述。于是，FPM 的损失可定义为：

$$\mathcal{L} = \frac{1}{2}\left(\mathcal{L}_{\text{vae}}\left(I_1\right) + \mathcal{L}_{\text{vae}}\left(I_2\right)\right) + \alpha \cdot \mathcal{L}_{\text{action}}\left(I_1, I_2\right) \tag{4}$$

I_1 和 I_2 属于域空间 $\mathcal{T}_I$，参数 α 用于控制潜在编码之间的距离对潜在空间结构的影响。

4.3. 折叠动作模块

折叠动作模块（FAM）学习一种策略 π_{a}，用于计算每对连续织物状态之间的动作 a_i。每个动作 a_i 包含两个值：抓取坐标 $g = [x_g, y_g]$ 和放置坐标 $p = [x_p, y_p]$。给定最短的

从起始状态到目标状态的中间折叠步骤序列，表示为 $\mathcal{G}$:$\langle I_0 = I_{\text{start}},\ I_1,\ I_2 \cdots I_{\text{n-1}},\ I_{\text{n}} = I_{\text{goal}} \rangle$，FAM 可以生成这些状态之间的抓取与放置坐标。

例如，执行动作 a_i后，织物可以从状态 I_{i}变为 $I_{\text{i+1}}$。通过计算序列中每个动作的最优抓取与放置坐标，FAM 使机器人能够高效、准确地执行折叠计划。

$$a_i = \pi_{\text{a}} \left(I_{\text{i}}, I_{\text{i+1}} \right) \tag{5}$$

Weng 等人 [14] 已经证明，基于流策略计算抓取与放置点，可以显著提升织物折叠过程的性能。为此，FAM 由两个主要组件构成：FlowNet 和 PickNet，如图 4 所示。

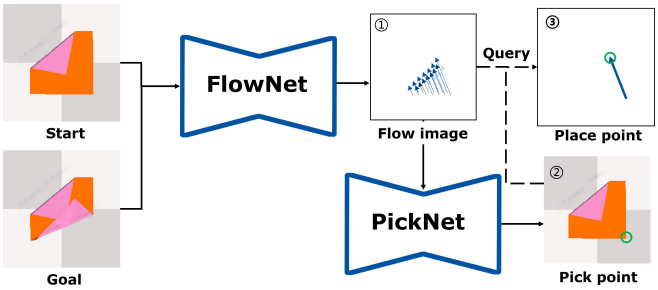


图 4. 折叠动作模块（FAM）利用流策略计算抓取与放置点，由 FlowNet 和 PickNet 组成。FlowNet 负责计算织物各粒子的光流，从而提供织物在折叠过程中如何移动和改变形状的信息。随后，PickNet 根据计算出的光流推理出最优抓取点。最后，通过查询抓取点的流动箭头来确定放置点；流动箭头提供了流动方向和距离的信息。

在 FlowNet 中，网络的训练损失为端点误差（EPE），即所有像素上的真实流 f 与预测流向量 f之间的平均欧几里得距离。因此，FlowNet 的损失函数为:

$$\mathcal{L}_{\text{Flow}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \hat{f} - f \right\|_2 \tag{6}$$

PickNet 是一种全卷积网络，以光流图像 f 作为输入，并生成单个热力图 H，用于预测机械臂的最优抓取点。抓取点通过 $g = arg\,max_g H(g)$ 计算得到。PickNet 使用二元交叉熵（BCE）损失进行训练，该损失用于衡量预测热力图 $\hat{H}$ 与真实热力图 H之间的差异。损失函数为:

$$\mathcal{L}_{\text{Pick}} = \text{BCE} \left(H, \hat{H} \right) \tag{7}$$

4.4. 迭代交互模块

迭代交互模块（IIM）用于在多步折叠任务中，每次抓取与放置动作后迭代执行折叠规划模块（FPM）和折叠动作模块（FAM）

直至达到目标状态。由于织物具有可变形性，且执行过程中可能存在不确定性，实际执行结果可能偏离目标状态。如图5所示，即使只执行一次动作，织物的目标状态与结果状态之间也可能存在细微差异。在多步折叠任务中，这些偏差可能不断累积并扩大，导致明显误差。

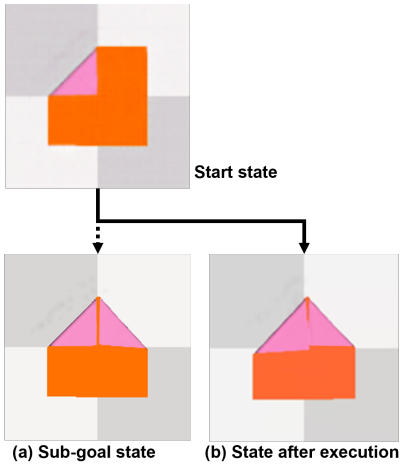


图5。 图(a)展示了FPM生成的推断中间折叠子目标，而图(b)展示了执行抓取放置动作后达到的实际状态。如图所示，两者之间存在轻微的可见差异，这表明推断的子目标与实际达到的状态之间可能存在偏差。

为解决这一问题，IIM被引入长时域折叠任务中。IIM能够减少每次动作执行后可能出现的不确定性，并促进更可靠的感知与规划，最终提升多步任务的性能。如图1所示，每次动作执行后，IIM都会将织物的新观测视为新的起始状态，并将其输入FPM，以生成新的折叠计划并推断新的中间状态序列。随后，这些中间状态会被输入FAM，由FAM计算并执行新的动作，从而生成新的织物状态。这一迭代过程会持续进行，直至达到最终目标状态。例如，在图示中，经过四次迭代后达到了目标状态。

通过这一迭代过程，IIM 能够减少执行偏差，提高多步折叠任务的效率和准确性。IIM 是应对偏离目标状态的潜在偏差、提升长时域折叠任务整体性能的关键组件。

5. 实验

5.1. 训练数据集

训练数据集 $\mathcal{T}_I$ 由以下形式的元组组成：$\langle I_0,\ I_1,\ (a,\ u)\rangle$。在每个元组中，$I_0 \subset \mathcal{I}$ 表示对应初始状态的图像，而 $I_1 \subset \mathcal{I}$ 表示对应后续状态的图像。ρ 表示发生在两幅图像之间的动作，其中 $a \in \{0, 1\}$ 表示是否发生了织物移动。u 包含夹爪的抓取与放置坐标点

，用于将 I_0 所表示的状态变为 I_1所表示的状态。因此，元组 $\langle I_0,\ I_1,\ (1,\ u)\rangle$ 是一个动作对，而元组 $\langle I_0,\ I_1,\ (0,\ u)\rangle$ 是一个无动作对。在无动作对中，两幅图像是对同一底层状态的不同观测，两幅无动作图像之间仅存在轻微扰动。

为了区分无动作对和动作对，我们根据织物粒子将动作定义为 0 或 1，其中 a $=1$ 表示织物粒子的最大运动超出了设定的运动范围。如图6右侧所示，在动作对中，两张图像呈现出明显变化，其在潜在空间中的低维表示也有所不同。相比之下，在无动作对的图像中，两张图像仅呈现出轻微的可见变化，且粒子的最大运动小于设定阈值。因此，它们在潜在空间中的低维表示相同。

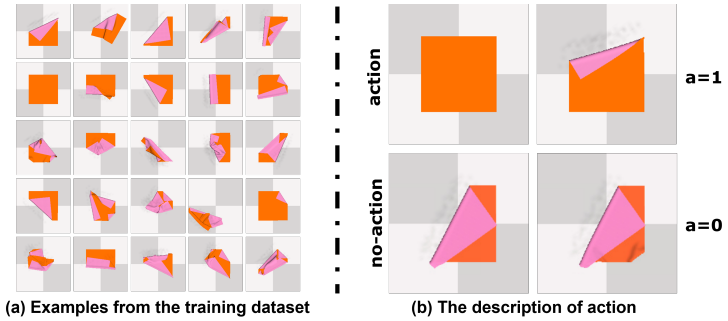


图6. （a）展示了训练数据集中的示例。对于每个样本，我们保存初始图像、抓取放置像素位置、执行动作后的下一张图像，以及两次观测中的织物粒子位置。（b）描述了折叠任务中动作对和无动作对的示例。

我们使用 SoftGym 收集研究所需的数据。SoftGym 是一个基于粒子的模拟器 Nvidia Flex 上构建的织物操作环境。我们通过对织物 [12,14]执行抓取与放置动作来生成数据集。图6（左）展示了数据集概况，其中记录了每次动作前后织物的信息。

我们将编码后的训练数据集 $\langle z_0$定义为 $z_1,\ \rho\rangle \in \mathcal{T}_z$ 它由将输入元组 $\langle I_0,\ I_1,\ \rho\rangle \in \mathcal{T}_I$ 编码到潜在空间 $\mathcal{Z}_{sys}$后得到的潜在元组集合构成。在这里，I_0 和 I_1 是输入图像，z_0 和 z_1 是图像在潜在空间 $\mathcal{Z}_{sys}$中的相应编码潜在表示。所获得的状态 $\mathcal{T}_z = \{z_0, z_1, \cdots, z_m\} \subset \mathcal{Z}_{sys}$ 称为覆盖状态。该方法具有较高的数据效率，因为我们采用的深度学习方法只要求数据集 $\mathcal{T}_I$ 包含系统所有可能动作对的一个子集，并且该子集能够充分覆盖系统动力学，而不需要包含全部动作对。

5.2. 实现细节

5.2.1. 网络实现细节

由 SoftGym [15], 收集的训练数据集包括 RGBD图像、织物的粒子信息、抓取放置动作的坐标以及图像中织物的掩码。在 FPM模块中，变分自编码器（VAE）的核心网络结构为 ResNet。FPM 的训练进行 800 个周期。训练期间，使用批量大小为 32，并将工作负载分配到 8

个工作进程。

在FAM模块中，FlowNet与 [24], 中的网络类似，但网络通道有所调整。此外，该模块采用了深度为六层的深度神经网络PickNet。两个模块均使用Adam优化器进行优化。我们将FAM模块的批量大小设为32，使用6个工作线程，训练轮数为300。

训练过程在运行Ubuntu 18.04的机器上进行。该机器配备了主频为3.50 GHz的Intel i9-9900X中央处理器、四块NVIDIA GTX 2080 Ti图形处理器和64 GB内存。整个训练过程约需57小时。

5.2.2. 物理实现细节

我们在一台配备DH-Robotics Adaptive Gripper PGE-50-26的六自由度UR5机器人上进行实验。实验工作空间的面积为1.7*1.2米。为保护夹爪，我们在工作空间内铺设了数块垫子。为获取RGB-D图像，我们在工作空间上方0.9米处安装了一台Microsoft Kinect V2.0传感器。实验场景已集成到机器人操作系统（ROS）中 [26] ，以便开展工作。图7展示了物理实验场景。

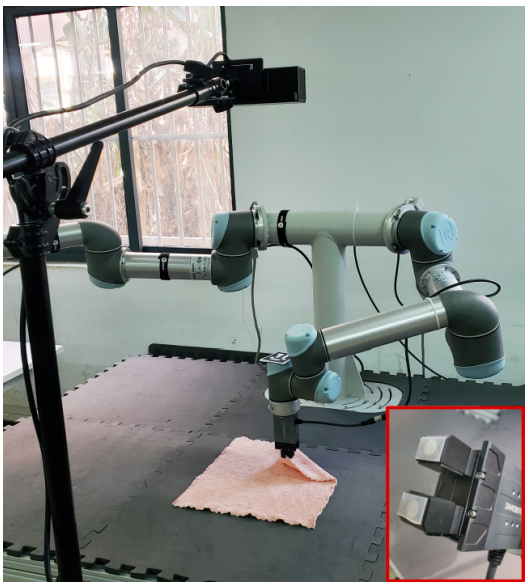


图 7。 在我们的物理实验场景中，我们在工作空间铺设了几块垫子，以保护夹爪。为了增加夹爪与织物之间的摩擦力，我们在夹爪末端粘贴了一块防滑硅胶垫，如红框所示。

5.3. 仿真实验

5.3.1. 实验设置

我们在 SoftGym 环境中评估了 DeFNet，并将其与三种基线方法进行了比较。我们假设初始织物处于展开状态，并已准备好被折叠为目标状态。所采用的评估指标是平均粒子距离误差，该误差针对达到的状态与目标织物构

型之间每个织物粒子的位置坐标计算。我们将我们的方法与 Lee 等人的工作进行了比较，[20],该方法是一种无模型方法。我们还与 Fabric-VSF 进行了比较，[21], 该方法通过交叉熵方法（CEM）学习视觉动力学模型。在本实验中，我们使用 RGB-D 数据集作为 Fabric-VSF 的输入。另一种基线方法是 LSR-V2 [12]。LSR-V2 与我们的方法之间的一个差异在于，LSR-V2 使用动作提议模块（APM）预测抓取与放置点，而我们使用基于流的策略。在这些实验中，我们仅输入织物的初始状态和目标状态，以比较所有算法最终的折叠结果。

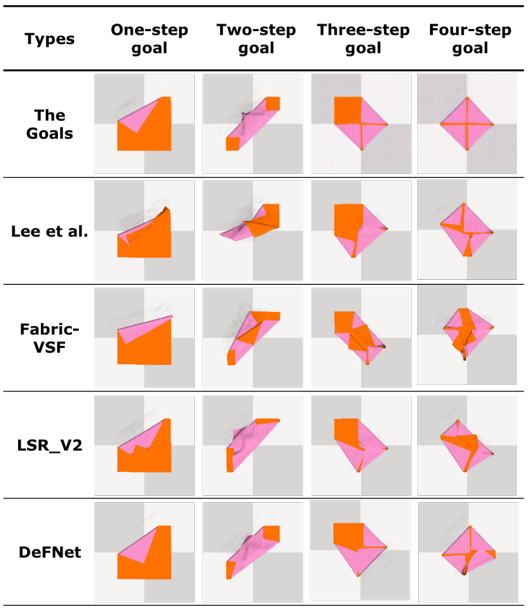


图8. 模拟器中的操作对比。我们的方法在折叠任务中表现最佳。

表1. 与基线方法的对比。织物折叠目标上的平均粒子距离误差（毫米）。

方法	一步	两步	三步	四步	All
Lee等人单臂 [20]	4.92±0.51	13.83±1.51	11.32±1.32	14.27±1.86	11.09±1.42
Fabric-VSF [21]	3.93±0.72	11.44±1.16	15.71±1.93	19.97±2.20	12.76±1.66
LSR-V2 [12]	4.27±0.43	7.08±1.11	12.47±1.70	15.90±1.33	9.81±1.23
DeFNet	**3.71±0.32**	**5.45±0.58**	**8.05±0.90**	**10.13±1.20**	**7.12±0.86**

5.3.2. 仿真实验结果

图8展示了所有方法的仿真结果，这些方法用于从相同的初始织物状态规划相同的目标。为评估这些方法的折叠性能，我们将目标构型分为四个等级，每个等级代表一种不同的目标形状，分别要求将织物至少折叠1、2、3或4次。我们进行了十次比较，并报告了织物的平均粒子距离误差（单位：毫米）。

平均粒子距离误差 根据表1中的四行数据可以看出，我们采用解构策略的方法在性能上优于基线方法。随着折叠难度增加，所有方法的平均粒子距离误差都随之增大

。不过，我们的 DeFNet 表现突出，在所有目标上的平均误差为 7.12，展现出卓越的性能。

表2. 四种任务中的抓取与放置动作数量。

方法	一步	两步	三步	四步
Lee等人 单臂 [20]	1.0	2.0	3.0	4.0
Fabric-VSF [21]	2.0	5.0	7.0	9.0
LSR-V2 [12]	1.0	2.0	3.0	4.0
DeFNet	1.0	2.0	3.0	4.0

抓取放置动作的数量 我们分析了各方法在折叠任务中执行的抓取放置动作数量。表2所示的结果表明，我们的方法以及 Lee 和 LSR-V2 基线方法在每项任务中所需的折叠动作数量都很少，且每个动作对应一个折叠步骤。相比之下，Hoque等人 [21]，报告的 Fabric-VSF 基线方法需要更多动作。这可能是因为折叠一个角时需要执行多个动作，正如他们的研究所述。

5.4. 真实世界实验

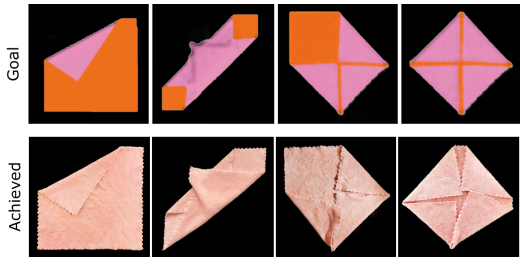


图 9. DeFNet 在真实世界实验中的结果。

表 3. 真实世界中所有织物折叠目标的平均交并比（MIoU）。

方法	一步	两步	三步	四步
DeFNet	0.952	0.887	0.921	0.826

图 9 展示了我们的系统在真实机器人上成功完成四项折叠任务的情况。由于机构精度和织物配置存在差异，模拟器中的结果与现实世界中的结果略有不同，但最终折叠出的毛巾与目标非常相似，肉眼几乎看不出差别，仅在细节上存在一些皱褶和轻微偏差。

表 3 展示了定量结果。在真实世界中，获取每个织物粒子的精确位置误差具有挑战性。因此，我们采用根据生成的织物掩码计算得到的平均交并比（MIoU）作为一种合理的指标。在一步折叠任务中，DeFNet 的 meanIoU 超过 0.952，而四步折叠任务的 mean IoU 达到 0.826。

6. 消融实验

我们对所提出的方法进行了消融实验，以评估该方法三个模块的重要性。消融实验旨在回答以下问题：

- 在织物任务中，使用流策略来确定折叠动作有何优势？在这项消融实验中，我们将 Lippi [12]的方法完整应用于折叠任务。Lippi 使用动作提议模块（APM）来确定抓取与放置点，这与传统方法类似，即同时推理抓取点和放置点。相比之下，我们的方法利用 Flownet 比较当前观测与子目标状态的每一对数据。随后，PickNet 利用流信息推理抓取点，并查询相应的放置点。

- 寻找折叠路径有何优势？在这项消融实验中，我们将初始状态和最终目标状态直接输入基于流的网络 FabricFlowNet（FFNet） [14], ，不进行路径规划，以评估在潜在空间中进行折叠规划的优势。

f

  在潜在空间中进行折叠规划。
- 迭代交互模块（IIM）有何优势？我们使用不含迭代模块的 DeFNet，这意味着所有折叠动作均完全根据首次推断出的中间折叠状态进行计算和执行。

表4. 消融实验中所有织物折叠目标的平均粒子距离误差（毫米）。

方法	一步	两步	三步	四步	All
LSR-V2 [12]	4.27±0.43	7.08±1.11	12.47±1.70	15.90±1.33	9.81±1.23
不使用子目标的 FFNet [14]	3.69±0.23	24.11±2.44	27.37±3.84	35.54±4.12	22.74±3.31
不含 IIM 的 DeFNet	3.71±0.32	6.95±0.64	9.33±0.98	11.98±1.23	7.98±0.92
DeFNet	3.71±0.32	5.45±0.58	8.05±0.90	10.13±1.20	7.12±0.86

为了量化消融实验的影响，我们在模拟器中进行了额外实验。首轮消融实验的结果见第5.3.2节，并直接用于我们的分析。消融实验的最终结果汇总于表4。

将 LSR-V2 与不含 IIM 的 DeFNet 进行比较后，我们发现，采用流策略来确定折叠动作时，性能提升了 18.7%。这表明，将流策略用作织物任务中动作的表示方式具有优势。

此外，使用潜在空间映射折叠路径显著提升了性能，距离从 22.74（不含子目标的 FFNet）降至 7.98（不含 IIM 的 DeFNet）。这得益于潜在空间能够捕捉中间折叠步骤。相比之下，由于缺少预测的中间折叠状态，不含子目标的 FFNet 会退化为单步折叠算法。

最后，比较 DeFNet 的距离 7.12 与不含 IIM 的 DeFNet 的距离 7.98 可知，将 IIM 应用于 DeFNet 后，性能提升了 10.8%。IIM 提供的迭代细化降低了对折叠过程进行预测时的不确定性，从而带来更准确的结果。

7. 结论与未来工作

在这项工作中，我们介绍了 DeFNet，这是一种用于织物折叠的新方法，由三个关键模块组成：折叠规划模块、折叠动作模块和迭代交互模块。为了评估该方法的有效性，我们在仿真中开展了多步织物折叠任务实验，并将其性能与三种基线方法进行比较。我们还进行了消融研究，以评估这三个模块各自的贡献。此外，我们将该方法应用于现有的机器人系统，并展示了其性能。未来，我们计划探索该方法在双臂折叠任务中的适用性。

参考文献

[1] Maitin-Shepard J、Cusumano-Towner M、Lei J 等。基于多视角几何线索的布料抓取点检测及其在机器人毛巾折叠中的应用。载于：2010年IEEE国际机器人与自动化会议；IEEE；2010年。第2308–2315页。[2] Miller S、Van Den Berg J、Fritz M 等。机器人洗衣折叠的几何方法。《国际机器人研究杂志》。2012年；31(2)：249–267。[3] Shibata S、Yoshimi T、Mizukawa M 等。面向家务机器人实现的布料物体折叠运动轨迹生成。载于：2012年第9届国际泛在机器人与环境智能会议（URAI）；IEEE；2012年。第126–131页。[4] Seita D、Jamali N、Laskey M 等。用于机器人铺床的织物抓取点深度迁移学习。载于：国际机器人研究研讨会；施普林格；2019年。第275–290页。[5] Huang Z、Lin X、Held D。结合遮挡推理的基于网格的布料操作动力学。arXiv预印本 arXiv:220602881。2020年。[6] Kaelbling LP、Lozano-P´erez T。信念空间中的集成任务与运动规划。《国际机器人研究杂志》。2013年；32(9-10)：1194–1227。[7] Bersch C、Pitzer B、Kammel S。用于洗衣折叠的双手机器人布料操作。载于：2011年IEEE/RSJ国际智能机器人与系统会议；IEEE；2011年。第1413–1419页。[8] Tsurumine Y、Cui Y、Uchibe E 等。采用平滑策略更新的深度强化学习：机器人布料操作中的应用。《机器人与自主系统》。2019年；112：72–83。[9] Ha H、Song S。Flingbot：动态操作在布料展开中的惊人有效性。载于：机器人学习会议；PMLR；2020年。第24–33页。[10] Canberk A、Chi C、Ha H 等。布料漏斗：面向多用途服装操作的规范化对齐。arXiv预印本 arXiv:221009347。2020年。[11] Lippi M、Poklukar P、Welle MC 等。可变形与刚性物体操作视觉动作规划的潜在空间路标图。载于：2020年IEEE/RSJ国际智能机器人与系统会议（IROS）；IEEE；2020年。第5619–5626页。[12] Lippi M、Poklukar P、Welle MC 等。通过潜在空间路标图实现物体操作的视觉动作规划。《IEEE机器人学汇刊》。2020年。[13] Wu Y、Yan W、Kurutach T 等。无需示范地学习操作可变形物体。arXiv预印本 arXiv:191013439。2019年。[14] Weng T、Bajracharya S M、Wang Y 等。FabricFlowNet：采用基于流的策略进行双手机器人布料操作。载于：机器人学习会议；PMLR；2020年。第192–202页。[15] Lin X、Wang Y、Olkin J 等。SoftGym：可变形物体操作深度强化学习的基准测试。载于：机器人学习会议；PMLR；2020年。第432–448页。[16] Seita D、Ganapathi A、Hoque R 等。通过算法监督器对连续织物平滑进行深度模仿学习。载于：2020年IEEE/RSJ国际智能机器人与系统会议（IROS）；IEEE；2020年。第9651–9658页。

telligent机器人与系统（IROS）；IEEE；2020年。第9651–9658页。[17] Ganapathi A、Sundaresan P、Thananjeyan B 等。在仿真中学习稠密视觉对应关系，以平滑并折叠真实织物。载于：2021年IEEE机器人与自动化国际会议（ICRA）；IEEE；2021年。第11515–11522页。[18] Willimon B、Birchfield S、Walker I。利用交互式感知展开衣物的模型。载于：2011年IEEE/RSJ国际智能机器人与系统会议；IEEE；2011年。第4871–4876页。[19] Triantafyllou D、Aspragathos NA。由单个机械臂在桌面上展开高度非刚性物体的视觉系统。载于：智能机器人与应用国际会议；施普林格；2011年。第509–519页。[20] Lee R、Ward D、Dasagi V 等。利用一小时真实机器人经验学习任意目标织物折叠。载于：机器人学习会议；PMLR；2020年。第2317–2327页。[21] Hoque R、Seita D、Balakrishna A 等。用于多步骤、多任务织物操作的VisuoSpatial Foresight。arXiv预印本 arXiv:200309044。2020年。[22] Finn C、Levine S。用于规划机器人运动的深度视觉前瞻。载于：2017年IEEE机器人与自动化国际会议（ICRA）；IEEE；2017年。第2786–2793页。[23] Nair A、Chen D、Agrawal P 等。结合自监督学习与模仿学习进行基于视觉的绳索操作。载于：2017年IEEE机器人与自动化国际会议（ICRA）；IEEE；2017年。第2146–2153页。[24] Dosovitskiy A、Fischer P、Ilg E 等。FlowNet：利用卷积网络学习光流。载于：IEEE计算机视觉国际会议论文集；2015年。第2758–2766页。[25] Kingma DP、Welling M。自编码变分贝叶斯。arXiv预印本 arXiv:13126114。2013年。[26] Mittler R。ROS很棒。《植物科学趋势》。2017年；22(1)：11–19。