学习多模态注意力以操作状态变化的可变形物体

Namiko Saito[1,2], Mayu Tatsumi[3], Ayuna Kubo[3], Kanata Suzuki[1,4], Hiroshi Ito[1,5], Shigeki Sugano[6] and Tetsuya Ogata[6,7]

Abstract— **为了支持人类的日常生活，机器人需要适应由于热量和力等外部因素而导致状态变化的对象，并根据情况采取适当的行动。日常环境中许多物体都表现出这种物理性质的动态连续变化。在这些情况下，来自多种模态的感觉输入通常既包含有价值的信息也包含噪声信息，而且随着对象状态的变化，每个传感器模态的重要性也会随时间发生变化。这使得实时感知和运动生成特别具有挑战性。我们提出了一种带有注意力机制的预测循环神经网络，该网络能够根据当前可靠性与相关性动态加权传感器模态，从而使机器人能够实现对处于状态变化中的物体的有效感知和自适应操作。为了证明所提方法的有效性，我们在一个物理类人机器人上进行了验证，并以烹饪炒鸡蛋的操作任务为例场景。我们的代码和数据集可以在这里找到：** https://github.com/namikosaito/CookingScrambledEgg

I. 介绍

对能够支持广泛日常任务的机器人需求日益增加。要在动态环境中运行，机器人必须感知物体状态的变化，并实时调整其行动。在许多日常生活场景中，物体并非静止不变——它们会受到热力和外力等外部因素的影响，从而导致其外在属性（形状、颜色、位置等）和内在属性（硬度、摩擦力、重量等）随时间变化 [1]–[3]。操控这些物体带来了独特的挑战，因为材料可能会随着时间软化、硬化、变形或改变状态，使得依赖预设的动作计划变得困难。识别这些变化需要融合多模态感官信息，包括视觉、触觉和力量。然而，这类信息往往包含有价值信号和噪声的混合，且每个传感器模态的重要性会根据物体当前的状态而波动。例如，当机器人未接触锅时，视觉信息很重要，而当视角被遮挡时，触觉和扭矩信息变得更加可靠，如图 2 所示。

Fig. 1: 烹饪炒蛋。

以往的研究主要只关注了外在属性 [4]–[7]。一些研究表明识别内在属性很重要，然而，它们要求预先定义探索性动作来识别内在属性然后再执行主任务 [8]–[10]，或者依赖于手动标注的标签 [11]。其他使用多模态感知的研究处理的是稳定或缓慢变化的目标 [12], [13]。我们假设关键限制是无法快速高效地处理感官信息，并在实时生成适应性动作以应对迅速变化的对象。为了解决这个问题，我们提出了一种具有注意力机制的预测深度学习模型，该模型根据当前可靠性和相关性动态加权感觉模态。虽然注意力机制已经应用于突出图像中的重要区域 [14], [15]，我们将这一概念扩展到连续操作任务中多模态传感器的权重分配。我们的方法积极决定关注哪些信息以及丢弃哪些信息，这些决策由对物体行为的预测引导。

我们在此展示了在一个炒蛋的情景中使用该方法，其中视觉和触觉/力反馈对于感知动态物体状态至关重要，机器人必须根据鸡蛋不断变化的属性调整其搅拌的方法和方向。由于加热和搅拌的作用，蛋液的性质会不断发生变化——变得越来越稠、更硬、结块，并且更加脆弱。机器人必须感知这种演变的状态并相应地调整其搅拌速度、轨迹和力度，否则会导致制作出质量不佳的食

*This work was supported by JST Moonshot R and D, Grant No. JPMJMS2031.

[1]Authors are with Future Robotics Organization, Waseda University, Tokyo, Japan. [2]Author is Microsoft Research Asia, Tokyo, Japan. (namikosaito@microsoft.com) [3]Authors are with Department of Modern Mechanical Engineering, Waseda University, Tokyo, Japan. [4]Author is with the Artificial Intelligence Laboratories, Fujitsu Limited, Kanagawa, Japan. [5]Author is with the Center for Technology Innovation - Controls and Robotics, Research & Development Group, Hitachi, Ltd., Ibaraki, Japan. [6]Authors are with Faculty of Science and Engineering, Waseda University, Tokyo, Japan. [7]Author is with the National Institute of Advanced Science and Technology, Tokyo, Japan.

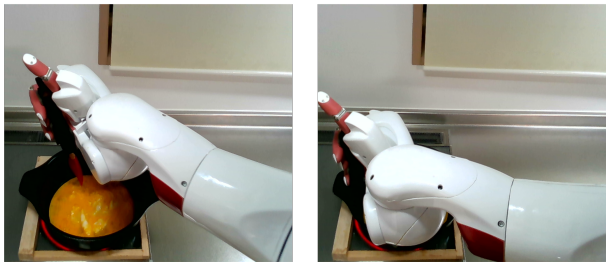

Fig. 2: 当机器人没有触碰到锅（左）时，视觉信息很重要。当视野被手臂遮挡（右）时，触觉和扭矩信息更加可靠。

物，如图 1 所示。仅依赖视觉可能会因机器手臂的遮挡而错过对质地变化或过度烹饪的检测，而单独使用触觉或力反馈则无法捕捉到燃烧的视觉提示。因此，炒蛋对于处理快速变化物体状态来说是一个具有挑战性的现实案例。

II. 方法

A. 演示学习

为了使机器人能够执行灵巧的日常操作，我们采用演示学习（LfD）[16], [17]，这是灵巧操作任务中常用的一种策略 [18]–[22]。我们通过遥操作收集专家示范，并使用这些数据训练机器人。

在我们的场景中，机器人需要在满足以下条件的情况下煎鸡蛋：（1）将蛋块分开，使每个块的长度不超过 15 厘米，（2）避免烧焦，通过颜色来判断，（3）确保没有生的部分留下。同时满足这些条件是非常具有挑战性的。如果没有适当的深度搅拌，只有底部会被煮熟，在顶部会留有生的部分。如果鸡蛋在锅里没有充分混合，生的部分会重新粘在一起形成更大的块并不均匀地烧焦。因此，机器人必须根据蛋块的分布、软度和厚度实时调整其动作。当简单地重复演示轨迹或随机移动时，机器人无法达到目标——留下生的部分，部分烧焦，或者未能分开大的块。即使是重播原来的遥操作轨迹也没有成功，因为在烹饪过程中鸡蛋的状态发生了动态变化。这突显了实时感知和运动适应的重要性，而不是静态的预先记录的动作。

B. 提议的学习模型

图 3 显示了所提出的深度学习模型，该模型由用于图像特征提取的卷积自编码器（CAE）[23] 和用于动作生成的多时间尺度递归神经网络（MTRNN）[24] 组成。在先前的工作 [8] 中，已经证明了 CAE 和 MTRNN 的结合对于动作生成是有效的。我们采用 MTRNN 主要有两个原因：首先，它提供了具有不同时间尺度的多个层次上下文节点，这允许解释任务执行中的长期和短期依赖关系。其次，其分层结构非常适合过滤多模态传感器输入，其中每种模式（例如视觉、触觉、扭矩）可以按其任务相关性加权。为了实现这一点，我们在 MTRNN 中集成了一个模态注意力机制，该机制动态调整感官运动输入的权重，并仅将相关信息传递到每个上下文节点。

TABLE I: CAE 的结构

	输入	输出	激活	处理
1	(128, 128, 3)	(64, 64, 8)	ReLU	convolution
2	(64, 64, 8)	(32, 32, 16)	ReLU	convolution
3	(32, 32, 16)	(16, 16, 32)	ReLU	convolution
4	8,192	1,000	ReLU	fully connected
5	1,000	MID	Sigmoid	fully connected
6	MID	1,000	ReLU	fully connected
7	1,000	8,192	ReLU	fully connected
8	(16, 16, 32)	(32, 32, 16)	ReLU	deconvolution
9	(32, 32, 16)	(64, 64, 8)	ReLU	deconvolution
10	(64, 64, 8)	(128, 128, 3)	ReLU	deconvolution

中间值 =20: 整体图像, 30: 裁剪后的图像

TABLE II: MTRNN 的结构

Nodes	Time constant	Number of nodes
C_s	32	7
C_f	5	30

1）计算机辅助工程：至于输入，我们使用两种不同尺寸的图像，一种显示整个手臂运动，另一种是裁剪后的图像，显示锅内情况。我们使用 CAE 将图像压缩为低维特征，以便 MTRNN 以一种良好平衡的方式学习所有感觉运动数据。CAE 被训练以输出重建数据（$y^{\mathrm{Img}}[t]$），与输入数据（$x^{\mathrm{Img}}[t]$）相同，其中 t 是时间步，通过最小化如下所示的均方误差（MSE），并使用自适应矩估计（Adam）算法的优化器。

$$E = \sum_{t} \left(y^{\mathrm{Img}}[t] - x^{\mathrm{Img}}[t] \right)^2 \tag{1}$$

我们使用中间层的数值作为图像特征（$x^{\mathrm{ImgFeature}}[t]$）。

表。I 展示了用于构建 CAE 的结构参数。我们将中间层神经元的数量设置为整个图像 20 个，修剪后的图像 30 个。我们测试了将中间层节点数量改变为 15, 20,..., 35，并设置这个数量使得输出（$y^{\mathrm{Img}}[t]$）能够通过检查重构原始图像（$x^{\mathrm{Img}}[t]$）的最小值。重构的图像不用于控制，而是供实验者验证 CAE 是否用整个图像表示了机器人手臂的位置，并且用修剪后的图像表示了个别的蛋块。包含 CAE 的模块训练了 1,500 个周期。

2）MTRNN：我们使用 MTRNN 作为主要模块来整合所有感觉运动数据，识别当前的蛋状态，并相应地生

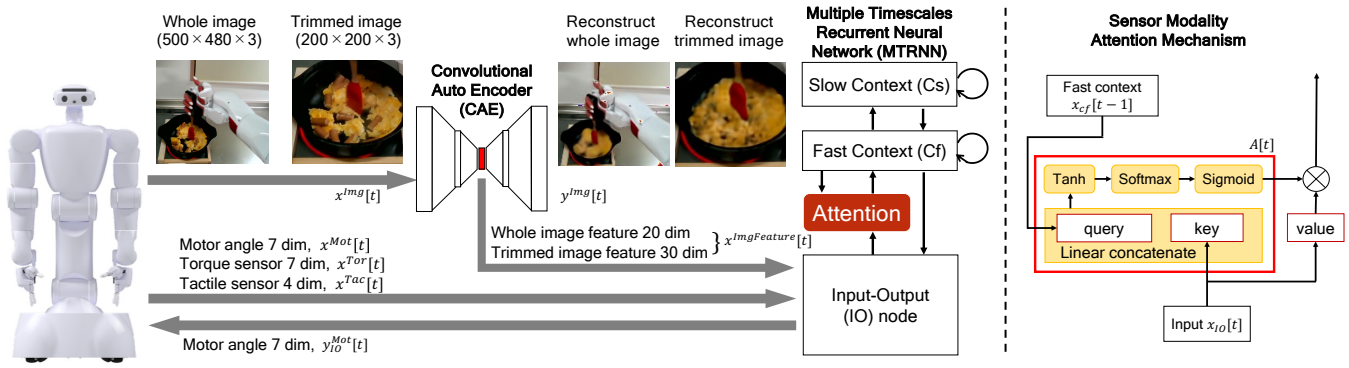

Fig. 3: 所提出的学习模型（左）由一个提取图像特征的 CAE 和一个带有传感器模态注意力机制的 MTRNN（右）组成，该 MTRNN 在考虑每个传感器模态的重要性时进行预测性学习。

成动作。MTRNN 是一个递归神经网络，它从当前和之前的输入预测下一步。它具有不同时间常数的多个节点，如表 II 所示。快速上下文（$\mathrm{C_f}$）节点用它们较小的时间常数学习短期原始数据，而慢速上下文（$\mathrm{C_s}$）节点学习顺序信息并表现得像具有较大时间常数的潜在空间。我们测试并选择了时间常数和节点数量，这些是能够最小化将在方程 11 中描述的误差的最佳组合。

在这项研究中，我们在 MTRNN 中实现了一个传感器模态注意力机制。我们将 IO 节点（$x_{\mathrm{IO}}[t]$）的值作为“键”和“值”，并将前一步骤中 $\mathrm{C_f}$ 节点的值（$x_{\mathrm{C_f}}[t-1]$）分别作为“查询”。使用键和查询，注意力机制（$A[t]$）按照以下方式计算并学习，并表示键的关注因子映射。

$$u_A[t] = w_{A,\mathrm{IO}}[x_{\mathrm{IO}}[t], x_{\mathrm{C_f}}[t-1]] \tag{2}$$

$$y_A[t] = \tanh(u_A[t]) \tag{3}$$

$$A[t] = \mathrm{sigmoid}\big(\mathrm{softmax}(y_A[t])\big), \tag{4}$$

其中，$w_{A,\mathrm{IO}}$ 是学习权重，与拼接后的键和查询相乘。在方程（4）中，仅使用 softmax 使得大多数值接近零，这导致上下文节点中只存储来自少数特定模态的信息。我们在 softmax 上应用 sigmoid 函数以平滑注意力图并监控所有模态。

前向计算过程如下所述。首先，输入数据 $\mathrm{T}[t]$，无论是模型训练期间的训练数据还是评估实验期间的实时数据。输入数据是来自 CAE（$x^{\mathrm{ImgFeature}}[t]$）的图像特征、扭矩传感器数据（$x^{\mathrm{Tor}}[t]$）、触觉传感器数据（$x^{\mathrm{Tac}}[t]$）和电机角度数据（$x^{\mathrm{Mot}}[t]$）。输入到 MTRNN（$x_{\mathrm{IO}}[t]$）的 IO 节点是 $\mathrm{T}[t]$ 与前一步预测的组合，其详细内容将在公式（10）中描述。

$$T[t] = [x^{\mathrm{ImgFeature}}[t], x^{\mathrm{Tor}}[t], x^{\mathrm{Tac}}[t], x^{\mathrm{Mot}}[t]] \tag{5}$$

神经元 i（$\in \mathrm{IO}, \mathrm{Cf}, \mathrm{Cs}$）在步长 t 处的内部值 u_i 的计算方式如下。为了计算节点 $\mathrm{C_f}$ 的内部值（$u_{\mathrm{C_f}}[t]$），我们利用与“值”（$x_{\mathrm{IO}}[t]$）相乘的注意力机制。

$$u_{\mathrm{IO}}[t] = w_{\mathrm{IO,Cf}} x_{\mathrm{Cf}}[t] \tag{6}$$

$$\begin{aligned} u_{\mathrm{Cf}}[t] = &\Big(1 - \frac{1}{\tau_{\mathrm{Cf}}}\Big) u_{\mathrm{Cf}}[t-1] + \frac{1}{\tau_{\mathrm{Cf}}} \big(w_{\mathrm{Cf,IO}} (A[t] x_{\mathrm{IO}}[t]) \\ &+ w_{\mathrm{Cf,Cs}} x_{\mathrm{Cs}}[t] + w_{\mathrm{Cf,Cf}} x_{\mathrm{Cf}}[t]\big) \end{aligned} \tag{7}$$

$$u_{\mathrm{Cs}}[t] = \Big(1 - \frac{1}{\tau_{\mathrm{Cs}}}\Big) u_{\mathrm{Cs}}[t-1] + \frac{1}{\tau_{\mathrm{Cs}}} \big(w_{\mathrm{Cs,Cf}} x_{\mathrm{Cf}}[t] + w_{\mathrm{Cs,Cs}} x_{\mathrm{Cs}}[t]\big) \tag{8}$$

其中，τ_i 是节点 i 的时间常数，$w_{i,j}$ 是从节点 j 到节点 i 的权重值，而 $x_j[t]$ 是输入值。随后，输出值计算为

$$y_i[t] = \tanh(u_i[t]). \tag{9}$$

该模块可以通过根据输出电机角度（$y_{\mathrm{IO}}^{\mathrm{Mot}}[t]$）控制机器人运动来生成机器人动作。随后，$y_i[t]$ 的值被用作下一个输入值，表示为

$$x_i[t+1] = \begin{cases} \alpha \times y_i[t] + (1-\alpha) \times \mathrm{T}[t+1] & i \in \mathrm{IO} \\ y_i[t] & i \in \mathrm{Cf}, \mathrm{Cs}. \end{cases} \tag{10}$$

接下来的输入值 $x_{\mathrm{IO}}[t]$ 通过将前一步骤的输出 $y_{\mathrm{IO}}[t-1]$ 和数据 $\mathrm{T}[t]$ 乘以反馈率 $\alpha (0 \le \alpha \le 1)$ 进行调整，这可以调整模型对其自身预测和实际输入之间的权重分配。如果反馈率较高，则该模型可以利用自身的预测历史稳定地预测下一步；而如果值较小，则该模型能够更灵活地适应实时情况，对实际数据更为敏感。我们将电机角度（$x_{\mathrm{IO}}^{\mathrm{Mot}}[t+1]$）设定为 0.8，其他值（$x_{\mathrm{IO}}^{\mathrm{ImgFeature}}[t+1], x_{\mathrm{IO}}^{\mathrm{Tor}}[t+1], x_{\mathrm{IO}}^{\mathrm{Tac}}[t+1]$）设为 0.6，这些是我们根据经验决定的。

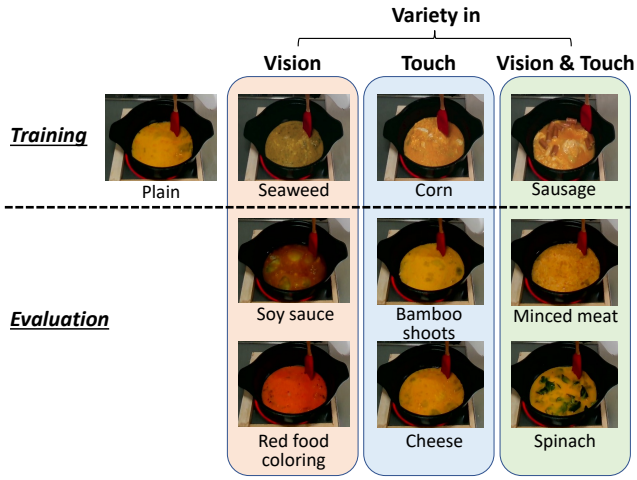

Fig. 4: 蛋液的变化。

为了在训练过程中实现反向计算，使用了通过时间的反向传播（BPTT）算法来最小化训练误差，公式如下：

$$E = \sum_t (y_{\mathrm{IO}}[t-1] - \mathrm{T}[t])^2. \tag{11}$$

权重更新为

$$w_{ij}^{n+1} = w_{ij}^{n} - \eta \frac{\partial E}{\partial w_{ij}^{n}}, \tag{12}$$

其中 $\eta(=0.001)$ 是学习率，$n(=20{,}000)$ 是使误差完全收敛的周期数。

C. 硬件和控制

我们使用 Dry-AIREC，它配备了 7 自由度的双臂和关节扭矩传感器。一个 RGB 摄像头（RealSense SR300）安装在它的头部，FSR406 触摸传感器则附着在其右手的手指和手掌上。机器人采用阻抗控制。

在一个厨房环境中，机器人左手拿着装有蛋液的量杯，右手拿着翻面铲。机器人通过预定义的动作倾倒蛋液，然后用右手按照提出的学习模型进行搅拌。

D. 任务设置和训练数据收集

我们通过改变加热功率，180 和 190° 摄氏度以及蛋液中混合的成分来进行实验（图 4）。训练使用了四种混合物：原味、添加 1 克海藻、100 克玉米和 10 根香肠——这些是为了隔离视觉、触觉或两者的效果。对于评估，我们准备了六种混合物：影响视觉的成分（酱油、食用色素）、触觉的成分（竹笋、奶酪）以及同时影响视觉和触觉的成分（绞肉、菠菜）。

我们通过指挥末端执行器的位置并求解逆运动学来决定臂的配置，从而收集演示数据集。我们采集了 32 个数据集：(4 种混合物) × (2 种温度) × (4 次试验)。我们在 2.5 Hz（每隔 0.4 秒）进行采样，平均烹饪时间是 180°C 时 620 步（248 秒），而 190°C 时为 455 步（182 秒）。右手的初始/最终姿态是固定的。输入到学习模型的数据包括：

- 图像：整体图像（500 × 480 × 3），和裁剪后的图像（从其左上角 (60, 270 像素) 处裁剪出的 200 × 200 × 3 部分），调整大小为 128 × 128 × 3。
- 触觉传感器：4 个点（食指、中指和无名指以及手掌）
- 扭矩传感器：7 个关节
- 动机角度：7 个关节

图像、触觉、扭矩和电机角度数据分别根据振动和噪声水平被归一化到 [0, 1]，[-0.85, 0.85]，[-0.9, 0.9]，以及 [-0.9, 0.9]。

TABLE III: 使用训练食材烹饪的成功率

heating power	plain	seaweed	corn	sausage	total
180°C	5/5	4/5 (push out)	4/5 (block)	4/5 (block)	17/20
190°C	5/5	4/5 (push out)	5/5	5/5	19/20
total	10/10	8/10	9/10	9/10	36/40

( ): 失败原因

III. 结果

A. 烹饪成功率

表 III 显示了使用训练过的蛋液混合物烹饪的成功率。我们进行了 40 次试验，总共 36（90.0%）次试验满足所有规则：（1）没有大于 15 厘米的蛋块，（2）没有烧焦的部分，和（3）没有生的部分。表格还显示了失败的原因。对于最简单的普通蛋液混合物，机器人在所有试验中都成功了。然而，在使用海带的情况下，机器人有两次将锅推出，这是因为海带的颜色是深蓝色的，CAE 难以区分海带与黑色的锅。在加热功率为 180°C 的试验中，分别用玉米和香肠进行一次试验时，一些大块仍然留在锅里而失败了。以 180°C 的加热功率烹饪比以 190°C 更难。在 180°C 下，蛋液需要更多时间凝固，并且即使机器人已经分离了蛋块，蛋液也容易再次粘附到其他蛋块上。

表 IV 显示了使用未经训练的食材进行鲁棒性评估的结果。我们进行了 60 次试验，总共成功了 47（78.3%）次。搅拌的方式和烹饪的时间因食材而异；然而，配备了我们学习模型的机器人识别出了这些属性并灵活地调整了它们。带有奶酪的鸡蛋试验尤其困难，因为奶酪粘稠且具有延展性，其性质与训练时使用的食材有显著不同；然而，该模型在 10 次试验中成功了 6 次。我们得出结论，该模型已经获得了良好的泛化能力。

TABLE IV: 使用未经训练的食材烹饪炒蛋的成功率

heating power	视野		触碰		视觉与触觉		total
	say source	red food coloring	bamboo shoots	cheese	minced meat	spinach	
180°C	5/5	4/5 (block)	3/5 (blocks)	4/5 (block)	3/5 (blocks)	4/5 (block)	23/30
190°C	4/5 (block)	5/5	4/5 (burn)	2/5 (push out, block, burn)	5/5	4/5 (block)	24/30
total	9/10	9/10	7/10	6/10	8/10	8/10	47/60
	18/20		13/20		16/20		

(): 失败原因

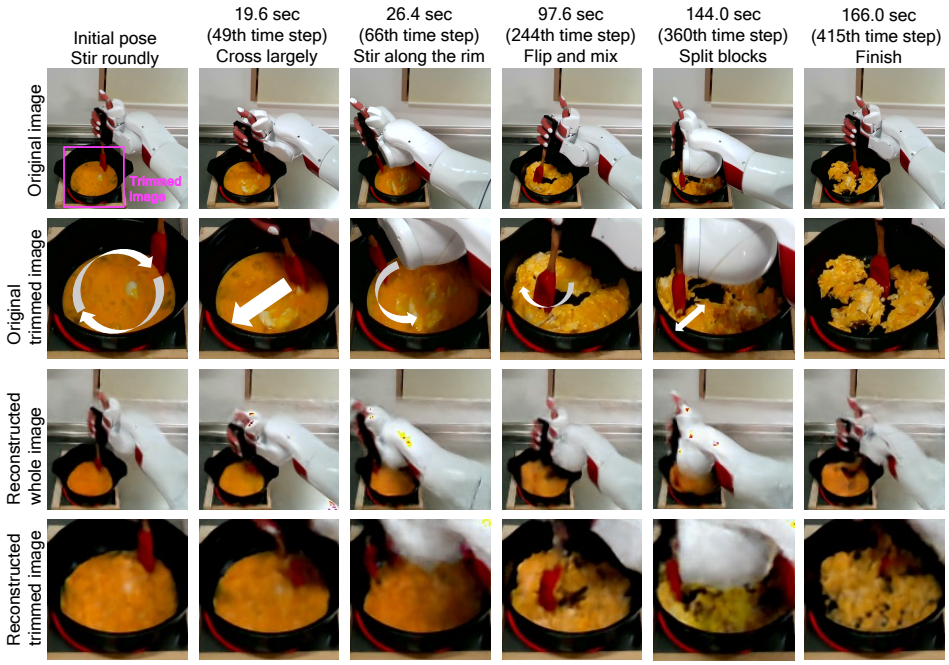

Fig. 5: 机器人使用我们的模型生成了动作，并成功用普通鸡蛋烹饪，加热温度为 190° 摄氏度。原始图像和重建的图像被展示出来。紫色方块区域被用作裁剪后的图像。根据情况，机器人执行了搅拌、翻面和分割操作。

B. 动作变化取决于蛋的状态

图 5 显示机器人在我们模型生成的动作下以 190°°C 的温度煮了普通的鸡蛋。首先，如图片所示，在开始时，机器人沿着锅边画圈搅拌，期望机器人识别鸡蛋的性质。其次，在第 49 步和第 66 步，当鸡蛋还生的时候，机器人在整个平底锅上大幅搅拌。随后，在鸡蛋开始受热后，机器人在第 244 步开始了翻动动作。接下来，在第 360 步几乎所有的部分都变硬时，机器人切开了大块的蛋并将其分开。最后，在第 415 步，机械臂回到初始姿势，完成了烹饪。这些动作并没有明确指示，但学习模型从演示中隐式地学会了它们。

我们也展示了由 CAE 重构的完整/修剪后的图像。完整的图像显示和代表了手臂的姿态，而修剪后的图像则显示鸡蛋的状态。特别是，修剪后的图像显示了鸡蛋的哪部分被聚集、分开或连接。因此，我们假设 CAE 识别了机器人的形态以及鸡蛋的状态。

C. 烹饪不同的菜肴

此外，我们还尝试烹饪完全不同的菜肴，如图 6 所示。机器人展示了炒饭、混合蔬菜和白汤的搅拌过程。尽管食材的状态变化与鸡蛋不同，但机器人展现了接近并分离聚集部分的动作。机器人决定动作结束的时间点，即当炒饭和蔬菜变软，以及白汤变得更为顺滑时。因此，我们的模型有可能识别出一般目标状态的变化。

IV. 讨论

A. 我们的模型的优势

所提出的学习模型提供了三个主要优势：实时响应能力、内存效率和对多种传感器模态的可解释性。首先，我们的模型提供实时响应能力，这对于需要即时反馈的动态任务至关重要。许多传统方法如动作分块与转换器

Fig. 6: 额外的实验使用不同的菜肴。

（ACT）模型 [21], [22] 使用时间窗口生成序列，这引入了延迟。相比之下，我们的预测模型在每个控制周期输出单一步长预测，使环境变化能够立即适应。

第二，我们的模型内存效率高。在这个任务中，它仅需 8.11GB 用于运动生成，而 ACT 在同一设置下需要 29.9GB。这使得我们的模型适用于物理机器人中的边缘计算，这对家庭辅助机器人尤其有益。

最后，我们的模型使用了多种感官模式；不仅包括许多传统研究中的视觉 [21], [22], [25]–[28] ，还包括触觉和力；这使得能够处理内在和外在属性。此外，通过特定于模态的注意力机制，我们的模型提供了模态可解释性，这是以前未曾解决的问题。这允许详细理解机器人如何与状态不断变化的对象进行交互，提高动态任务中的适应能力，并增强机器人实时做出明智决策的能力。

B. 臂轨迹分析与消融比较

为了验证我们模型的有效性，我们进行了一项消融研究。我们在移除传感器模式注意力机制的基础上，使用相同的参数训练了同样的 CAE 和 MTRNN 架构。没有注意力机制的情况下，我们将机器人加热至 190°°C 来煮普通鸡蛋，进行了五次试验，在这种条件下，所有试验在我们的模型下都成功了。结果，在四次试验中，机器人因为施加的力量过大而推到了锅壁并停止了运动，因为它超过了安全扭矩的限制。有一次试验中，机器人只搅拌了锅内可见位置，手臂没有遮挡到锅，并且鸡蛋是可见的。因此，没有注意力机制的情况下，机器人无法集中于扭矩和触觉传感器信息，难以在翻蛋器碰到锅壁后立刻回到正确的动作上。此外，没有注意力机制的模型过于依赖视觉，因而机器人只搅拌了可观察的位置。出于安全考虑，我们停止了实验，因为击打的动作会损坏机器人。

我们分析了搅拌轨迹，跟踪了翻转器的尖端，并比较了训练数据、我们的模型和没有注意力机制的模型，如图 7 所示。它们都是在机器人加热至 190° 摄氏度搅拌全蛋的情况下的情况。通过我们中间所示的图 7 的模型，轨迹覆盖了锅中的所有区域，就像左边的训练数据一样。此外，轨迹显示，一开始机器人以大动作圆周搅拌接近整个区域。然后逐渐将动作变为尖锐翻转或切分，针对特定区域。最后，机器人可以像训练数据那样烹饪出色炒蛋。另一方面，右边所示的没有注意力机制模型图显示，机器人只搅拌了锅的右侧区域。最终，左侧有一些大的块状物未被搅拌且烧焦。因此，我们可以认为模态注意力机制对于根据物体复杂的动态属性调整动作至关重要。

C. 传感器模态注意力分析

我们分析了注意力机制以确认每次关注的模态。在图 8 中，我们展示了注意力映射，这些是表示注意力权重的热图。这些图形中的数值已经标准化，颜色越浅表示该模态被高度聚焦，学习模型认为其重要且可靠。它们都是机器人搅拌普通鸡蛋并加热到 180°C 的相同情况。

图 8(a) 是整个维度输入数据的注意力图。我们观察到在第 148、186、242、300、347 和 425 步附近出现遮挡时有线条。换句话说，我们看到了集中感觉运动数据的转换。学习模型识别出遮挡并根据情况改变焦点。

图 8(b) 显示了每种模态平均后的注意力图。可以看到，特别是扭矩传感器数据的热色变得越来越浅，这意味着它在序列后半部分获得了关注。我们认为因为在后半部分，鸡蛋变得更硬；因此，学习模型必须专注于识别硬度；哪一部分是液态、半生还是硬的。相比之下，在最后一部分中，对整个图像数据的关注度减少。一开始，机器人需要大幅度移动以搅拌整个锅，但后来，动作变得越来越小，并涉及翻转或分割特定区域，这主要需要手腕转换。因此，整个图像的意义逐渐降低。

图 8(c) 显示了仅关注电机角度数据的注意力图。我们可以看到，位于肩部和肘部的第 2、4 和 5 个关节减少了注意力，主要用来在整个锅中搅拌。相比之下，在手腕上的第 6 和 7 个关节增加了注意力，这些关节主要用于打蛋。我们可以说，注意力机制根据鸡蛋的状态和必要的动作类型关注重要的关节。

D. 蛋状态检测分析

我们在 MTRNN 中进行评估实验时，对 Cs（$u_{\mathrm{Cs}}[t]$）的顺序内部值进行了主成分分析（PCA）。Cs 节点中有七个节点，但通过 PCA，我们可以用两个维度来分析 Cs 所表示的信息。图 9 展示了每个加热功率和配料的两个样本；图 (a) 显示了直到第 520 步的序列，这是仅当以 190°°C 加热完成烹饪的时间点，而 (b) 则显示到第 640 步，这是大多数试验中以 180°°C 加热也完成烹饪的时间点。在 (a) 中，使用 190°°C 时，所有流都已经达到了正值；然而，在使用 180°°C 的试验中，所有流都没有从负值移动。另一方面，在 (b) 中，几乎所有流都到达了负值。因此，PC1 轴上的从负到正的过渡解释了鸡蛋的状

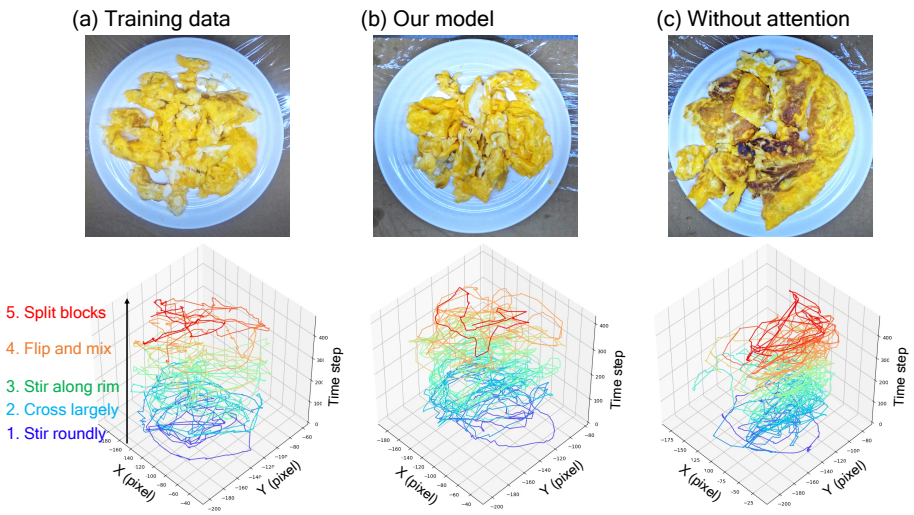

Fig. 7: 烹饪过程中使用普通鸡蛋时的臂部轨迹，温度为 190°C。训练数据、我们的模型和没有注意力机制的模型的搅拌路径及最终菜肴如图所示。我们的模型广泛覆盖了锅体，并随着时间从广泛的搅拌转变为集中且有针对性的动作，这与训练数据相似。而没有注意力机制的模型更倾向于右侧以保持锅体可见。

态，从生到熟。总之，Cs 节点代表了鸡蛋的状态，并提供了完全搅拌的时间，使机器人能够完成烹饪任务。

V. 限制与未来工作

未来工作有一些领域需要探索。首先，我们希望将这种方法推广到多个连续任务中。在这项研究中，我们仅关注单一操作任务。然而，许多实际任务需要结合多种动作并适应动态变化的对象状态。最近使用任务规划框架 [29], [30] 和基础模型 [27], [28] 的研究表明可以解决长期的操作问题，并能够提供有价值的见解。

其次，在物理环境中使用真实机器人进行所有实验存在实际限制。为了提高可扩展性和灵活性，我们将利用仿真环境安全高效地收集数据，并建立一个从仿真到现实的迁移管道，以确保在真实环境中的可靠部署。

第三，未来的工作包括对性能如何随附加传感器模态或更高分辨率输入扩展进行更详细的研究。此外，我们将评估模型对于一个或多个模态中的传感器噪声或故障的鲁棒性。

VI. 结论

我们解决了处理动态变化属性的对象的挑战。我们提出了一种实时感知对象外在和内在属性的学习模型，该模型使用包括视觉、触觉和扭矩信息在内的多模态感官数据。为了实现这一目标，我们引入了一个具有模式注意力机制的预测性递归神经网络，该机制根据输入的可靠性和重要性自适应地加权感官输入，使机器人能够根据任务环境灵活改变关注点。通过涉及动态操作任务的实验，我们证明了所提出的系统即使在未经训练的情况下也能实现稳健和泛化的性能。结果表明，选择性整合多模态信息对于处理状态持续变化的对象至关重要，并且我们的方法可以提高此类任务所需的实际感知和自适应控制能力。这项工作代表了向更自主、多功能的机器人系统的一步发展，这些系统能够在复杂、动态和不确定的真实世界环境中提供帮助。

References

[1] S. J. Lederman and R. L. Klatzky, “Hand movements: A window into haptic object recognition,” Cognitive Psychology, vol. 19, no. 3, pp. 342–368, 1987.

[2] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, “Tactile sensing—from humans to humanoids,” IEEE Transactions on Robotics, vol. 26, no. 1, pp. 1–20, 2010.

[3] T. Bhattacharjee, G. Lee, H. Song, and S. S. Srinivasa, “Towards robotic feeding: Role of haptics in fork-based food manipulation,” IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 1485–1492, 2019.

[4] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar, “Food recognition using statistics of pairwise local features,” in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 2249–2256.

[5] A. Petit, V. Lippiello, G. A. Fontanelli, and B. Siciliano, “Tracking elastic deformable objects with an rgb-d sensor for a pizza chef robot,” Robotics and Autonomous Systems, vol. 88, pp. 187–201, 2017.

[6] X. Lin, C. Qi, Y. Zhang, Z. Huang, K. Fragkiadaki, Y. Li, C. Gan, and D. Held, “Planning with spatial-temporal abstraction from point clouds for deformable object manipulation,” in 6th Annual Conference on Robot Learning, 2022.

[7] M. S. Salekin, A. B. Jelodar, and R. Kushol, “Cooking state recognition from images using inception architecture,” 2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST), pp. 163–168, 2019.

[8] N. Saito, T. Ogata, S. Funabashi, H. Mori, and S. Sugano, "How to select and use tools? : Active perception of target objects using multimodal deep learning," IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 2517–2524, 2021.

[9] N. Saito, N. B. Dai, T. Ogata, H. Mori, and S. Sugano, "Real-time liquid pouring motion generation: End-to-end sensorimotor coordination for unknown liquid dynamics trained with deep neural networks," in IEEE International Conference on Robotics and Biomimetics (ROBIO), 2019.

[10] T. Lopez-Guevara, R. Pucci, N. K. Taylor, M. U. Gutmann, S. Ramamoorthy, and K. Subr, "Stir to pour: Efficient calibration of liquid properties for pouring actions," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020.

[11] J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh, "Physically grounded vision-language models for robotic manipulation," in arXiv preprint arXiv:2309.02561, 2023.

[12] M. C. Gemici and A. Saxena, "Learning haptic representation for manipulating deformable food objects," in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2014.

[13] P. Sundaresan, S. Belkhale, and D. Sadigh, "Learning visuo-haptic skewering strategies for robot-assisted feeding," in 6th Annual Conference on Robot Learning, 2022.

[14] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Attention branch network: Learning of attention mechanism for visual explanation," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 10 697–10 706.

[15] H. Ichiwara, H. Ito, K. Yamamoto, H. Mori, and T. Ogata, "Contact-rich manipulation of a flexible object based on deep predictive learning using vision and tactility," in 2022 International Conference on Robotics and Automation (ICRA), 2022, pp. 5375–5381.

[16] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: a survey of learning methods." ACM computing surveys, vol. 50, 2017.

[17] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," Annual Review of Control, Robotics, and Autonomous Systems, vol. 3, no. 1, pp. 297–330, 2020.

[18] J. Liu, Y. Chen, Z. Dong, S. Wang, S. Calinon, M. Li, and F. Chen, "Robot cooking with stir-fry: Bimanual non-prehensile manipulation of semi-fluid objects," IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 5159–5166, 2022.

[19] H. Kim, Y. Ohmura, and Y. Kuniyoshi, "Robot peels banana with goal-conditioned dual-action deep imitation learning," ArXiv, vol. abs/2203.09749, 2022.

[20] Y. Saigusa, S. Sakaino, and T. Tsuji, "Imitation learning for nonprehensile manipulation through self-supervised learning considering motion speed," IEEE Access, vol. 10, pp. 68 291–68 306, 2022.

[21] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," Robotics: Science and Systems (RSS), 2023.

[22] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," in Conference on Robot Learning (CoRL), 2024.

[23] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in Artificial Neural Networks and Machine Learning – ICANN 2011. Springer Berlin Heidelberg, 2011, pp. 52–59.

[24] Y. Yamashita and J. Tani, "Emergence of functional hierarchy in a multiple timescales recurrent neural network model: A humanoid robot experiment," PLoS Computational Biology, vol. 4, no. 11, 2008.

[25] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in Proceedings of Robotics: Science and Systems (RSS), 2023.

[26] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," The International Journal of Robotics Research, 2024.

[27] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo, T. Kreiman, Y. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine, "Octo: An open-source generalist robot policy," in Proceedings of Robotics: Science and Systems, Delft, Netherlands, 2024.

[28] Q. Li, Y. Liang, Z. Wang, L. Luo, X. Chen, M. Liao, F. Wei, Y. Deng, S. Xu, Y. Zhang et al., "Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation," arXiv preprint arXiv:2411.19650, 2024.

[29] J. Yi, T. A. Luong, H. Chae, M. S. Ahn, D. Noh, H. N. Tran, M. Doh, E. Auh, N. Pico, F. Yumbla, D. Hong, and H. Moon, "An online task-planning framework using mixed integer programming for multiple cooking tasks using a dual-arm robot," Applied Sciences, vol. 12, no. 8, 2022.

[30] Z. Wang, C. R. Garrett, L. P. Kaelbling, and T. Lozano-Pérez, "Learning compositional models of robot skills for task and motion planning," in The International Journal of Robotics Research, vol. 40, 2021, pp. 866–894.

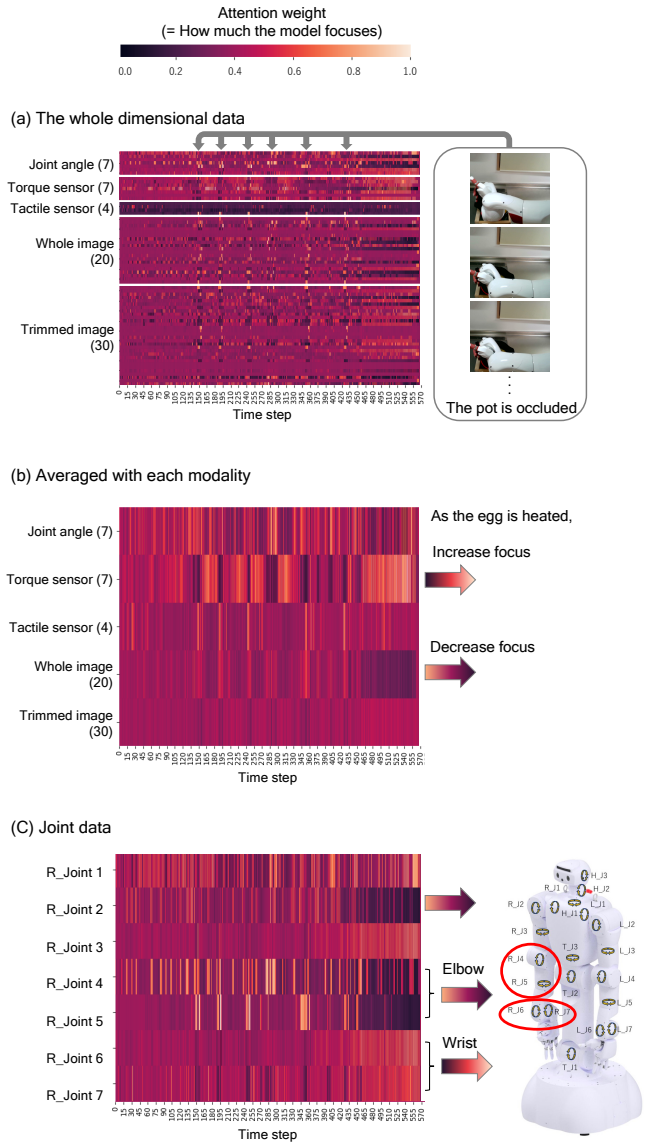

Fig. 8: 烹饪过程中使用普通鸡蛋在 180°C 时的注意力图。在 (A) 中，随着遮挡事件的发生，模态之间的注意力转移由垂直波线表示。在 (B) 中，在后半部分扭矩传感器成为焦点，而整个图像失去了焦点。在 (C) 中，对关节 2、4 和 5 的关注度降低（用于广泛的搅拌），而对关节 6 和 7 的关注度增加（用于精细分割）。

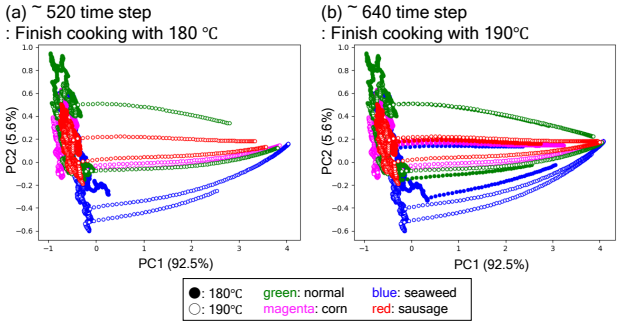

Fig. 9: 烹饪过程中顺序 Cs 节点值的主成分分析。每个加热功率和原料展示了两个样本。上：序列直到 520 步；下：直到 640 步。PC1 捕捉了鸡蛋从生到熟的变化，其值沿着 PC1 轴变化。Cs 节点根据烹饪状态编码搅拌完成时间。