

【神经符号 AI 讲座】神经符号 AI 赋能机器人直觉——机器人位置猜测网络的原理与实现

编者按：上海交通大学神经符号人工智能暑期学校开学典礼暨社区研讨会，于 2024 年 6 月 24-25 日在上海交通大学召开。上海交通大学彭颜龙作了题为《机器人位置猜测网络的原理与实现》的社区共享项目研究进展报告。“机器人位置猜测网络”项目采用神经符号人工智能方法，使机器人初步具备类似人类的直觉，从而提升其在复杂环境中的自主操作能力。报告深入探讨了神经符号人工智能方法下的直觉实现细节，以及对未来发展和通用人工智能领域贡献的展望，凸显了该技术对未来机器人智能化发展的积极影响。

一、机器人是否可以像人一样拥有直觉？

图 1 是生长在印度尼西亚四王岛咸水红树林的射水鱼。这是一种奇特的鱼类，它不捕食水中的生物，而是通过在水中发射高速射流，精准地将树上的昆虫击落，然后这些昆虫就会落入水中。



图 1 印度尼西亚四王岛咸水红树林里特有的射水鱼，喷水打落空中飞虫捕食。

在此，我想请大家思考一个问题：在执行捕食任务时，鱼类是否经过深思熟虑，选择合适的速度、力量，以及考虑水的折射率去喷射水流？显然，它们并没有。鱼类通过长期的捕食，逐渐掌握了捕猎技巧。换句话说，是否可以认为鱼类在捕猎方面拥有自己的直觉？它们不需要通过大量的计算来考虑水的

折射率，或是今天阳光和水的状况是否会影响捕获虫子，而是将这些技能内化为本能反应，无需判断就能执行，这便是所谓的生物直觉。人类同样拥有生物直觉，例如篮球运动员经过大量的练习，能够精准投篮，在出手的一刹那的“手感”其实也是直觉的一部分。

生物的直觉可以定义为生物通过进化学习和感知发展出的快速的、无意识的反应和决策能力。那么我们就要思考，生物是这样，我们机器人是否也能拥有直觉呢？近年来，许多机器人的操作让人觉得它们似乎具备了知觉。例如，丰田的投篮机器人在早期能够达到 100%的投篮准确率，室内导航机器人通过 SLAM 实现感知规划，波士顿动力的机器人可以完成炫酷的跑酷动作，如图 2。

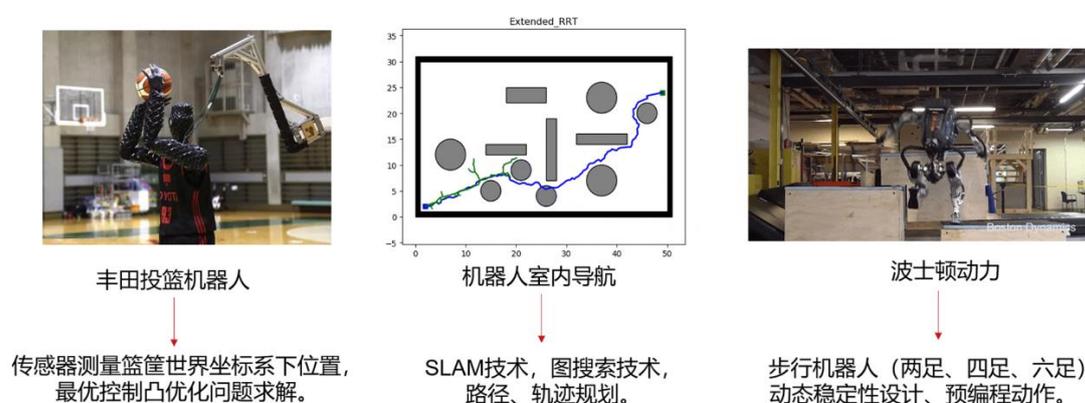


图 2 相关的诸多研究，并不是直觉。

那么我们考虑这是不是机器人的直觉呢？但如果我们深入探究其技术实现，就会发现投篮机器人是通过传感器测量篮筐和自身位置，计算抛物线，将投掷问题转化为凸优化问题来解决，背后是有优化模型的。室内导航技术融合了 SLAM 技术、图搜索技术和 RRT、RRT*等路径规划技。波士顿动力公司研发的双足机器人，当然也包括四足和六足机器人的动态稳定性设计，同时它有一些预编程的动作。尽管如此，它们与人类的投篮和射水鱼的捕食动作在本质上仍有区别，似乎并非基于直觉。

就在不久前，DeepMind 发布了一项最新研究，该研究利用深度强化学习技术训练了一个足球场上的射门机器人。大家会觉得，似乎这个机器人通过深度强化学习训练，能够完成射门动作，还能实现过人，似乎是它在短时间内做出的决策，而且似乎没有经过大量的计算和思考。因此，深度强化学习可能是实现机器人直觉的一个途径。然而，我们需要认识到，深度强化学习的解释性不足，并且仅在特定场景下有效。由于它是在特定场景下进行训练的，因此在不

同的足球场或足球上，它可能无法识别。此外，训练过程繁琐，主要源于它需要对环境进行建模，包括定义状态、动作、奖励策略等函数，这与我们在工业应用中寻求一个通用的万能算法来解决问题存在一定的出入。

在深入探讨如何利用神经符号人工智能指导机器人的直觉之前，我们首先应考虑直觉研究的意义。尽管直觉不能为机器人的动作带来绝对确定性，但它能在复杂且动态的环境中增强机器人的自主操作能力。特别是在涉及人机交互的场景，如家庭机器人和救援机器人等，如果机器人具备了直觉，便无需通过大量的计算、仿真和模拟，就能更好地理解人的行为，从而提高安全性。从经济和社会影响的角度来看，这种计算资源效率的提高将带来了显著的效益。

二、神经符号人工智能框架下的机器人直觉

2.1 神经符号人工智能框架

接下来，我们将探讨在神经符号人工智能指导下是否能够实现直觉。首先，让我们回顾一下先前的研究工作。我们基于 PDDL 语言建立了一个神经符号人工智能的任务与运动规划（NSAI-TAMP）框架，如图 3。



图 3 “知行合一、训推一体”的神经符号人工智能框架

这是我们已有的一个成果。神经符号人工智能系统融合了感知、规划、执行、学习、验证等功能，实现自主、可学习、可解释、可扩展的机器人具身控制架构。如图 3，当开始执行任务时，通过多传感器获得的 RGB 图像、深度图

像、力-力矩等信息通过神经网络，也就是我们所谓的**神经谓词**，让系统获得当前的机器人状态和当下的环境状态。同时我们会告诉系统，目标状态是什么。比如，我们希望把这颗螺栓拆掉，这是目标状态。通过神经谓词的感知，就能在符号空间中表示出机器人当前状态，也能在符号空间中表示出目标状态。

接下来是规划。通过深度学习或者先入先出（FIFO）的规划器，机器人能在符号空间中寻找一条从当前状态到目标状态的最短路径，这条路径由众多**动作原语**组成。这意味着机器人能够自主选择动作，根据当前环境实时判断接下来应执行的操作。例如，它可能需要**清除障碍**才能继续**拆解**，或者需要调整位姿才能**对准**螺栓。规划完成后，系统会发送指令给机器人硬件去执行。同时，神经谓词会实时验证传感器获取的数据所对应的状态。也就是说，当机器人进行了拆解操作，关键是需要确认是否成功拆下了这颗螺栓。这一点至关重要。如果未能成功拆解，意味着之前的判断和规划是错误的。因此，需要通过学习过程来修正系统之前对世界认知的偏差。

因此，目前我们的框架可以总结为“知行合一、训推一体”，其中“知”代表感知、验证、学习，“行”代表规划和执行。感知与执行之间通过神经谓词实时更新状态。“训推一体”则是一种学习模式，即神经谓词的网络训练和推断都是实时、自主进行的，不包含任何人工预编程。因此，在此前的研究工作，我们已经成功开发了“退役动力电池机器人智能拆解工作站系统”（NSBD Station），通过引入神经谓词，将视觉、力觉信息和机器人位姿等连续空间状态映射到符号状态，自主选择并执行动作原语，自主完成拆解任务。

- 协作机器人有效载荷：10kg；工作半径1300mm；6个旋转关节自由度；可重复性： $\pm 0.1\text{mm}$
- 末端执行机构：包含5套专为拆解场景设计的末端执行器及旋具自动更换系统，该系统支持20种以上旋具的灵活更换，并具备接杆加长功能。此外，还能实现3种不同拆卸扭矩的紧固件拆卸，以及2种不同深度的切割操作。
- 多维动态视觉、力觉传感系统
- 基于神经符号AI框架设计的软件控制系统，支持15种神经谓词、12种动作原语



图 4 退役动力电池机器人智能拆解工作站系统

2.2 我们为何要研究机器人的直觉？

我们为何要研究机器人的直觉？因为在较大的工作空间内，特别是在工业环境中，如面对多种类、复杂类型的动力电池拆解任务时，固定工位式的工作

站系统已无法满足需求。在需要任务调度或执行大范围作业任务的情况下，机器人必须具备移动能力。因此，我们选择了自主移动操作机器人（AMMR）来解决全自主拆解任务，如图 5 所示，这需要高效的底盘运动规划方案。



图 5 电池拆卸自主复合型机器人系统 BEAM-1 号

我们研究机器人直觉，目的是指导机器人在任务和运动规划中明确行动方向。例如，“机器人现在前往工作台的动力电池处拆卸一个螺栓”，这就要求我们预先掌握机器人当前的位置及目标运动位置。只有这样，我们才能进行路径规划，即决定机器人如何到达拆螺栓的位置，并进行运动学仿真。当然，这一过程可以通过 SLAM（同步定位与地图绘制）技术实现，但稍后我将解释为何无需依赖 SLAM。

在面对一个固定的拆解位置时，我们的运动规划算法框架，如 PDDL Stream，会通过随机采样来决定行动。例如，在拆解过程中，当需要移动到工作台的另一个位置时，它会通过随机采样来决定移动路径。具体来说，系统可能会随机决定移动到工作台的位置，然后利用运动学仿真来判断这个路径是否可行，是否会碰撞到机器人本体或者障碍物。我们知道，六自由度的机器人有八组解，但相比人手有二十多个自由度，其自由度其实是较低的。因此，在规划过程中，我们不得不仔细考虑随机采样的点是否可行，是否存在碰撞风险。

在传统的规划方法中，实际上采用的是随机采样技术。例如，之前通过使用 SLAM 进行实时规划，其实也是依赖于算法的随机采样机制。我们可以观察到，尽管过程中可能会采样出许多无用的点，但其目标导向性是明确的，这正是通过大量随机采样和仿真来实现的。然而，这种方法在工业环境中会占用大量资源，尤其是在资源受限的情况下。

我们是否能够实现机器人直觉呢？这里有几个问题值得我们深思。

我们最根本要解决的问题，是随机采样和模拟的低效率，以及模拟结果与实际情况之间的差距，这正是阻碍机器人迈向敏捷的关键瓶颈。在决策阶段，运动控制更多地依赖于既定的逻辑来解决目标参数问题（预编程），而无法迅速适应动态变化的环境。现有的机器人移动到哪儿呢？实际上过于依赖我们所谓的既定逻辑。例如，机器人移动到螺栓旁边 5 厘米或 50 厘米的距离，这些数值实际上是靠人去预编程的。包括它的运动方式和拆解过程，很多都是预编程问题，这正是我们想解决的主要问题。

我们能否依靠直觉来解决这个问题呢？最简单的例子是，当我们需要去拿远处桌子上的水时，我们实际上会下意识地走过去。我们并不需要考虑如果走到门的那个地方，我们就够不到这个水了，或者说我如果走到这个桌子就撞到了。我们并没有下意识去思考这些问题，我们反而会下意识地走到桌子旁边，在一个合适的距离去抓到这瓶水。这证明了直觉能够帮助我们解决问题。

那么，为什么不采用 SLAM 与视觉识别引导相结合的方法呢？我们考虑到，SLAM 只是对环境的感知，但此类方法到达目标的实质性运动规划还是通过采样的方法去实现的，运动学仿真求解是不可避免的过程。这样即使建立了高精度地图，运动规划的效率并没有被提高，这是具有探索性质的方式。可以观察到，许多室内导航技术被应用于家庭服务机器人中。这些机器人能够绘制家庭地图，并将其划分为不同的区域，如厨房和客厅，然后进行物体识别和抓取，以执行一系列家庭服务场景。然而，在工业精度，尤其是拆解电池包等任务中，无法达到与之相当的效率。

因此，我们需要明确我们的目标。如图 6 所示，一个可移动的机器人面对着一个远距离的电池包，或者说是需要大范围拆解的电池包，我们想要解决的问题是机器人移动到达的最佳位置。如果机器人过于接近，将会与电池包发生碰撞。如果距离过远，则会超出机械臂的工作范围，使得机器人无法触及螺栓。最佳解决方案是能够直接移动到一个既不导致机器人本体与电池包碰撞，又能确保机械臂能够到达并操作螺栓的位置。如果采样过程能通过神经网络的快速正向传播来实现，将极大地提高效率并减少计算资源的消耗。因此，我们希望所选的采样点既满足运动学的要求，又位于合适的工作空间内。

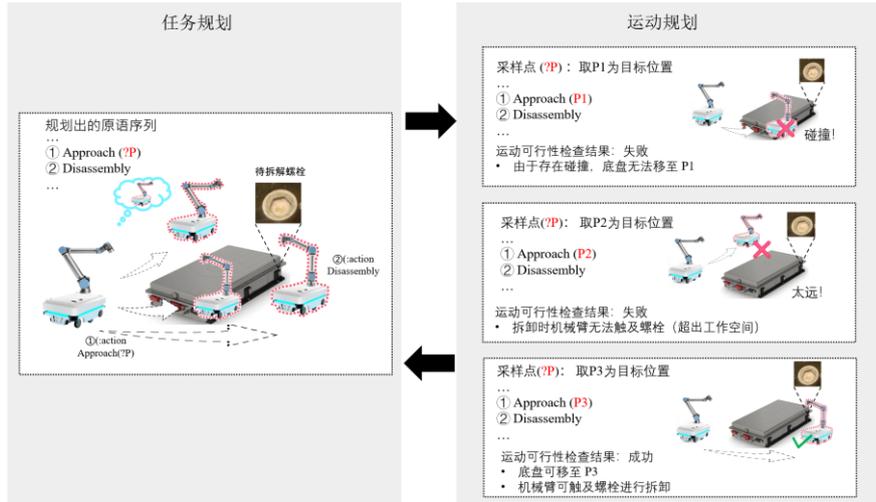


图 6 基于神经符号人工智能的任务和运动规划 TAMP

2.3 机器人位置猜测网络

我们提出了一种“机器人位置猜测网络”（RPSN）运动规划框架，如图 7 所示。

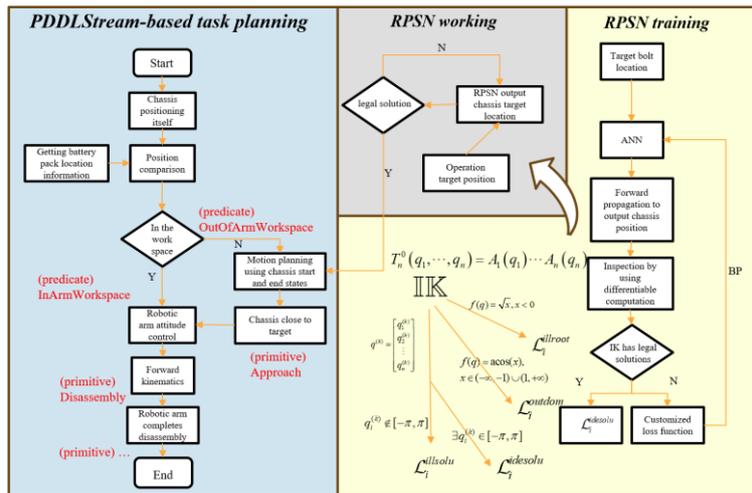


图 7 RPSN 运动规划框架：训练与工作流程

RPSN 是一种基于神经符号人工智能的移动机器人运动规划框架。借助 RPSN，机器人运动规划问题可以摆脱随机采样目标点模式下所耗费的大量时间和仿真运算成本。当我们向 RPSN 提供待拆解零件的目标位置时，它能够直接输出可信的用于任务规划的底盘目标位置，仿佛机器人真正理解了自己的任务和目标位置。相较于随机采样过程，RPSN 具备可微分、可解释、可持续学习和可扩展的特性，能够出色地完成动态场景下非结构化拆解工作的运动规划指导。

神经符号人工智能思想下的 RPSN 为何能为机器人赋予直觉？这是因为神经符号人工智能融合了知识驱动的逻辑系统与符号推理（类似于人类的推理能

力) 和数据驱动的神经网络概率学习 (类似于人类的直觉)。通过这种双重机制, RPSN 能够为机器人提供一种近似人类直觉的决策能力。

类似于人类的直觉, 机器人能够内在化其目标和行动路径。我们的研究框架是可微分、可解释和可扩展的, 目的是实现可持续学习。首先, 我们需要思考为何其他研究未能实现机器人直觉, 而神经符号人工智能却能为直觉的实现提供启示。如图 8 所示, (a) 上方是传统随机采样模式, 我们通常将待操作对象的位姿作为采样点。通过随机采样, 我们得到的是待操作对象, 比如一颗螺栓的位姿。这时, 通常需要运动仿真引擎进行仿真, 以确认该位置无碰撞且能成功拆解螺栓, 进而进行 SLAM 和规划。如果不满足条件, 比如发生碰撞, 就需要重新进行随机采样; 若满足条件, 则就输出采样结果交给机器人执行。

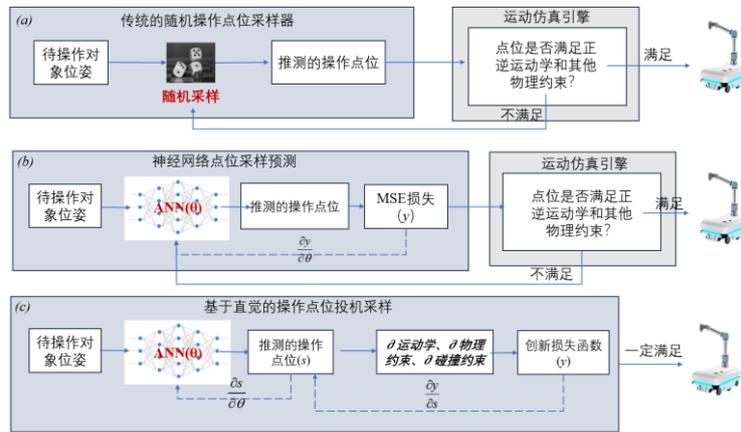


图 8 RPSN 与传统方法比较

当前流行的神经网络和深度学习方法同样适用于此类场景。如图 8 (b) 所示, 例如, 我们可以利用大量的已知数据关系, 构建一个数据集来表示拆解关系。神经网络通过学习, 能够根据网络的预测值与真实值之间的损失进行差值计算, 并将误差反向传播回网络, 从而实现位姿采样。然而, 由于神经网络的不可解释性, 我们无法预先确定某个特定的采样点是否可行, 因此这些采样点仍需通过运动学仿真引擎进行验证, 以确保神经网络预测的点位是切实可行的。

在神经符号人工智能框架下, 我们构建的直觉点位网络, 即 RPSN, 与传统神经网络的主要区别在于, 它将机器人的运动学和碰撞等物理信息有机地融入了网络结构中。这一融合是通过微分形式的运动学实现的, 唯有以微分形式表达, 这些信息才能被纳入网络, 并参与损失函数的反向传播过程, 从而使网络能够学习并掌握这些物理规律。因此, 在这种直觉启发式采样网络的预测下,

无需任何仿真验证。只要通过我们的框架输出结果，即可确保在运动学方面一定存在可行解。

RPSN 主要由两个核心模块构成。如图 8 (c) 所示，一个是以传统神经网络为基础，构建的位置预测网络，沿用了深度学习的网络结构；另一个是可微分的运动学计算引擎。我们将这两个模块紧密结合。当该系统输出待拆解螺栓的位置时，它能直接提供一个使关节具有合法解的底盘位置。这种类似于人类下意识的、无需仿真和繁琐计算、无需深思熟虑的行为，可以极大地提升运动规划的效率。

我们需强调的是，我们所提出的基于神经符号人工智能思维下的 RPSN 与传统神经网络的根本区别在于，我们无需提供大量依赖人工标注的数据。换言之，在训练过程中，我们并未依赖真实的标签。众所周知，训练一个深度学习网络通常需要准备大量带标签的数据集，例如，若要训练一个能识别小猫的网络，就需要大量的猫的图像数据进行标注。这正是人们常说的“有多少人工就有多少智能”的原因，因为数据标注需要投入大量的成本。然而，在我们的方法中，无需提供任何预选择的数据，网络的优化方向是由微分编程后的人类先验知识定义的，这带来了显著的数据优势。

机器人运动学算法通过一种可微的训练方式被有机地融入到网络中，并参与反向传播过程。至于具体的技术细节，将在后续进行详细讨论。我们究竟做了什么？通俗地说，在机器人认知世界的过程中，它真正地学习了物理约束，例如运动学是否有解的问题，并以直觉的形式展现出来。这与仅仅通过预测和与真实值的比较来学习特定数据集的映射截然不同。

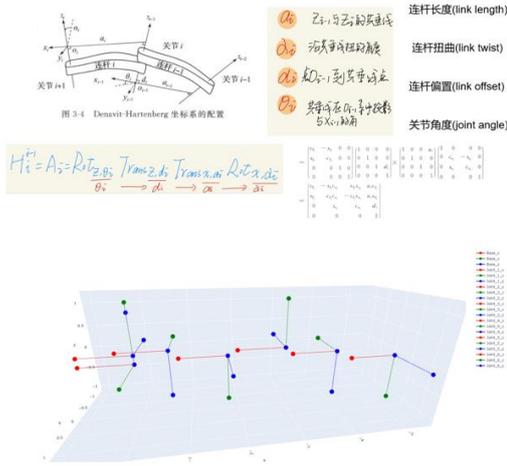
三、基于可微过程的 RPSN 训练

3.1 机器人运动学

机器人正运动学是已知每个关节的角度，通过计算得出机器人末端执行器位姿，即人类手臂的指尖的位姿。而逆运动学则是已知指尖位置，求解每个关节的角度。我们采用的是经典的六自由度 DH (Denavit-Hartenberg) 运动学，如图 9。这意味着机器人任何两个连杆之间的坐标系关系是通过连杆长度、连杆扭曲、连杆偏置、关节角度这四个值的矩阵，即称为齐次变换矩阵 (Homogeneous transformation matrix) 的 4 乘 4 的矩阵来表示的。通过这种方式，

可以在六自由度机器人的关节坐标系中，表示每个关节的独立坐标系。由此，末端或首端坐标系与基座坐标系的关系可以通过连续的关节矩阵来表达。正运动学问题因此可以通过这四个参数的矩阵乘积来解决。对于六自由度的机器人来说，逆运动学问题则涉及到解这个矩阵。

快速回顾6-DOF-DH运动学



正运动学: 根据给定的机器人关节变量的取值来确定末端执行器的位置和姿态。

逆运动学: 通过末端执行器的位置和姿态来求解对应的关节变量。

UR10e						
Kinematics	theta [rad]	a [m]	d [m]	alpha [rad]	Dynamics	Center of Mass [m]
Joint 1	0	0	0.1807	$\pi/2$	Link 1	7.369 [0.021, 0.000, 0.027]
Joint 2	0	-0.6127	0	0	Link 2	13.051 [0.38, 0.000, 0.158]
Joint 3	0	-0.57155	0	0	Link 3	3.989 [0.24, 0.000, 0.068]
Joint 4	0	0	0.17415	$\pi/2$	Link 4	2.1 [0.000, 0.007, 0.018]
Joint 5	0	0	0.11985	$-\pi/2$	Link 5	1.98 [0.000, 0.007, 0.018]
Joint 6	0	0	0.11655	0	Link 6	0.615 [0, 0, -0.026]

$$H_i^{i-1} = A_i = Rot_{z, \theta_i} Trans_{z, d_i} Trans_{x, a_i} Rot_{x, \alpha_i}$$

$$H_{total} = \begin{pmatrix} R & d \\ 0 & 1 \end{pmatrix}, \quad H_{total} = \prod_{i=0}^5 H_i^{i+1}$$

$R \in SO(3), \quad d \in \mathbb{R}^3$

$$T_n^0(q_1, \dots, q_n) = H_{total}$$

$$T_n^0(q_1, \dots, q_n) = A_1(q_1) \cdots A_n(q_n)$$

图 9 机器人 6 自由度 DH 运动学

3.2 RPSN 的实现方法

RPSN 的训练和 workflows 如图 7 所示。左侧的浅蓝色区域代表任务规划阶段。当输入为目标螺栓的位置时，系统判断该位置是否在机器人的工作空间内或外部。如果在工作空间内，则执行后续的动作原语；若在工作空间外，则调用 RPSN 的前向预测流程。值得注意的是，RPSN 的部署极其简单，只需要一次正向传播即可实现其功能。

RPSN 的训练过程尤其值得关注，因为它涉及到如何以可视化的方式将成熟的算法融入神经网络的训练中。我们的目标是使网络真正学会物理约束，而不仅仅是从特定数据集中学习到该数据集的物理约束。这涉及两层关系的处理，并且该方法需要具备扩展性和普适性。

接下来，我们将详细探讨 RPSN 的训练过程。首先解释一下什么是可微。这里的可微与运动学中的“可微”有很大区别。运动学中的可微是指通过运动学空间几何关系，例如在平面直角坐标系中表示出 XY，然后两边求全微分，从而计算雅可比矩阵。雅可比矩阵定义为关节空间速度向末端操作空间速度传递的映射矩阵。通过雅可比矩阵，可以将关节力矩映射到末端力和力矩，进而求解

运动机器人的静力学。进一步通过雅可比矩阵再进行一次全微分，可得动力学方程。这就是通常所说的求偏导数和全微分，也被称为方程的可微。

然而，我们所讨论的“可微”，是指在编程形式上的可微，即可微分编程（Differentiable Programming）。这意味着我们需要用代码的形式将一个不连续的函数，或者说把物理的逻辑转化为代码可微分的形式。因为只有代码的可微分才能被用于神经网络的训练。

让我们看一下当前深度学习中常用到的神经网络是如何工作的。这是一个基础的多层感知机（MLP），具有三层结构，包括输入层、隐藏层和输出层。图 10 是其计算图，它表示了神经网络中每个节点在正向传播过程中所经历的运算。

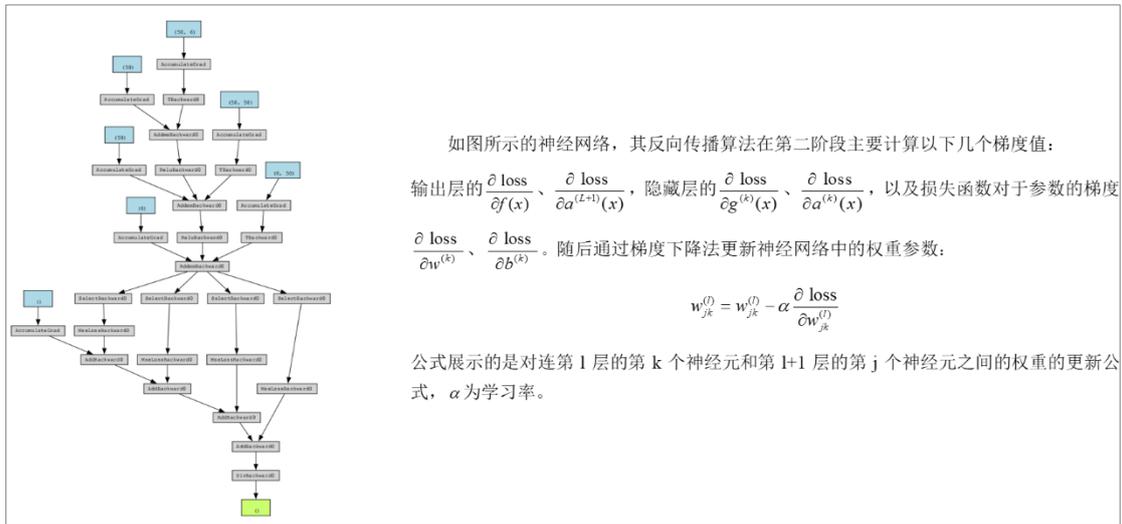


图 10 一个人工神经网络的计算图：三层 MLP。

如果细看可以发现，如 AddmmBackward、ReLUBackward，其中的 ReLU 就是神经元的一个典型激活函数。因此，神经网络通过正向传播和反向传播来优化梯度，实现模型训练。实际上，当前 99% 的深度神经网络都在采用这种基本原理，离不开这些核心公式，即反向传播算法。

在神经网络的训练过程中，通过计算损失函数对模型参数（如权重 W 和偏置 B ）的偏导数，来调整这些参数，从而实现模型的最佳拟合。这一基于梯度的优化过程对于包括 MSE（均方误差）损失在内的各种损失函数都至关重要，是确保所有类型神经网络性能优化的关键步骤。幸运的是，这一过程在深度学习框架如 PyTorch 和 TensorFlow 中已有现成的梯度求解器，我们仅需一行代码即可完成反向传播过程。

然而，与传统方法不同，我们关注的是让机器人真正理解和掌握物理规律，

包括碰撞和运动学约束，这需要对神经网络的正向传播过程进行修改。如图 11 所示，这是 RPSN 正向传播的局部图。上半部分是 MLP 的正向传播过程。这意味着，我们需要以一种可微分编程的方式，将物理学的约束添加到神经网络的训练过程中，通过神经网络前向传播得到最下面的损失，然后再通过反向传播去优化参数。

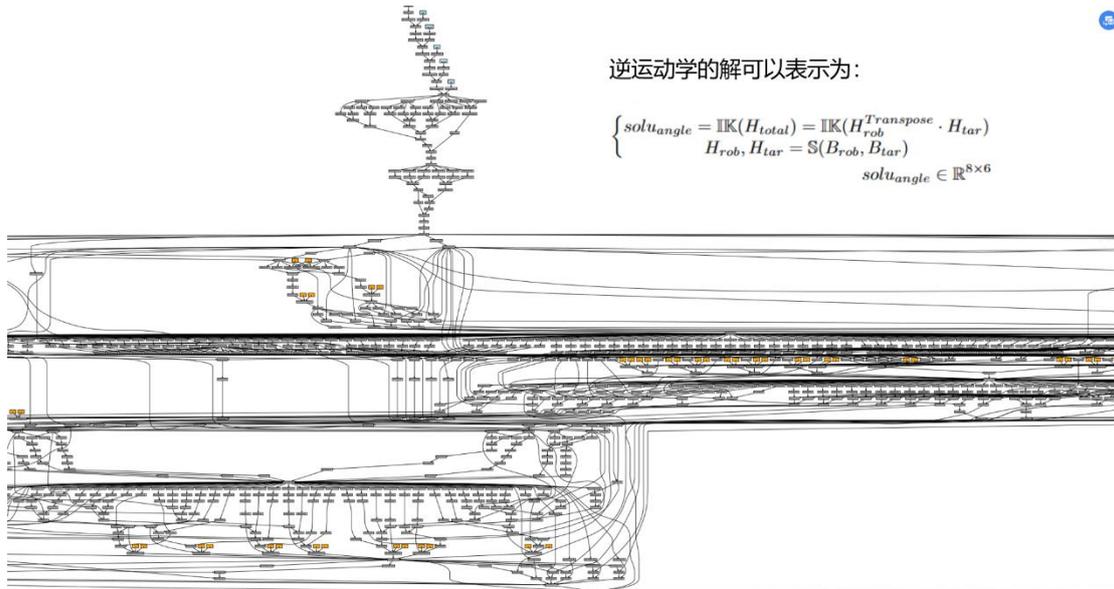


图 11 加入了可微运动学的 RPSN 正向传播计算图（局部）

在编程过程中，为了使求解过程具备可微性，我们对传统的矩阵式逆运动学求解方法进行了重新定义，引入了可微分的计算流程。具体而言，我们采用了张量运算来离散化角度求解，并定义了相应的 IK 算子和 S 算子。这一改进使得机器人的逆运动学解决方案能够无缝集成到计算图中，从而支持端到端的优化和训练。计算图中的每个节点代表一个可微的数学运算。只有当这些小框内的数学运算都是可微的，我们才能保证整个计算图是连通的，从而能够进行反向传播以完成训练。

综上所述，在可微分的计算流程中，我们解决了三个主要的技术难题：

- 首先，我们探讨了如何构建计算图，即如何构建一个可微分的运动学模型，将这一物理约束以可微的方式融入神经网络中，使得机器人能够真正理解这一物理约束，而不是仅依赖于我们人为编程的、穷举出来的物理约束。
- 其次，RPSN 作为一个没有明确手动标注标签数据参与训练的模型，我

们需要定义一个损失函数来指导模型训练。

- 第三，我们如何保证网络的随机性。

不连续函数或超出定义域的函数如何微分并参与计算图？

我们以计算机器人逆运动学时需要用到的 atan2 函数为例。该函数并非标准函数，其自变量并非遵循一对一或一对多的关系。它是一个离散且不连续的定义形式，显然不具备可微性。如果将此函数加入神经网络的计算图中，它将无法参与正向和反向传播过程。我们采用的方法是在编程阶段，利用可微编程重新计算正向传播，重新定义了 atan2 函数的正向和反向传播过程。解决此类问题的实质源于神经网络中的链式法则，即，当前节点在反向传播过程中的梯度，是通过当前节点对当前函数的偏导数，乘以先前传播的偏导数计算得出的。我们通过将其表达为可微的形式，并将其表示在计算图中。

$$\begin{cases} \nabla_y = \nabla_{\text{output}} \times \frac{\partial}{\partial y} \text{atan2}(y, x) = \nabla_{\text{output}} \times \frac{x}{x^2+y^2} \\ \nabla_x = \nabla_{\text{output}} \times \frac{\partial}{\partial x} \text{atan2}(y, x) = \nabla_{\text{output}} \times -\frac{y}{x^2+y^2} \end{cases}$$

如何定义损失函数才能为训练提供有效的引导？

解决这个问题涉及到自定义损失函数。定义一个损失函数似乎并不是一件难事，在深度学习和强化学习中，经常需要进行模型构建和函数定义，无论是奖励函数还是互动函数。然而，我们在这里定义损失函数的目的，是因为我们不需要人工逐张对图像进行增值标注。没有人工标注的数据集，网络是无监督的。那如何让它从无监督变成类似自监督学习呢？我们可以通过自定义的损失函数来对传播过程进行限制。以 RPSN 的训练过程为例，通过图 12 可以看到，它的前面部分是一个位置预测网络，这是一个传统的神经网络；后一部分，已经将运动学转换为可微分的形式，现在该计算损失函数了。损失函数是对整个可微的运动学的一个约束，目的是让机器人系统真正掌握物理规律。

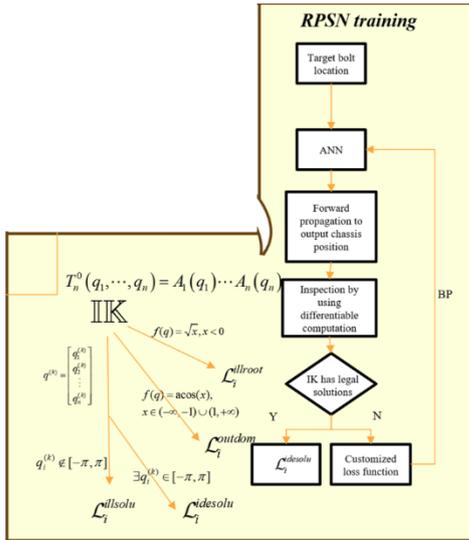


图 12 自定义损失函数训练 RPSN

在损失函数的定义部分，我们设定了多种损失函数。实际上，这与我们之前讨论的逆运动学过程中的情况相关。在逆运动学过程中，我们希望确保每组解中的六个关节都有合法的标准值。如果出现不合法的解，例如奇异位置，我们认为网络没有成功学习到物理约束，因此会对不合法解进行惩罚。因此，在此过程中，我们定义了许多这样的损失函数。最终的总损失是通过赋予不同权重来调整这些损失函数得到的。权重 δ 值，即损失的比例，决定了哪些子损失应当优先受到惩罚。U 是在定义 RPSN 系统时，我们希望引入一种针对特定任务的损失，同时也能推广到其他任务中的普适性损失。因此，U 的值能够控制带来两种 RPSN 系统的行为，具体细节在此不做赘述。

如何保证 RPSN 网络的随机性？

在没有使用 dropout 或其他随机化层的情况下，神经网络的预测是一次正向传播。在参数权重固定的情况下，相同的输入将产生相同的输出。为了防止在极小概率下出现特定输出的问题，我们需要让网络具有微小的随机性，同时，也能确保它确实学到了逆运动学的规律。我们的解决方案是在训练过程和预测过程中均引入了 dropout 层。正常情况下，dropout 层的使用是为了降低过拟合问题，这一方法在网络训练过程中尤为常用。通过调整 dropout 的超参数，我们能够确保这个随机性不会成为一个固定的点，同时也能满足我们前面损失函数所希望学习到的一些规律。

理想解 For $k = 1, 2, \dots, 8$ and $n = 1, 2, \dots, 6$

$$q^{(k)} = [q_n^{(k)} \ q_n^{(k)} \ \dots \ q_n^{(k)}]^T, \quad \text{where } q_n^{(k)} \in [-\pi, \pi]$$

损失函数

$$\mathcal{L}^{total} = \frac{1}{N} \sum_{i=1}^N (U \cdot \delta_{illroot} \mathcal{L}_i^{illroot} + U \cdot \delta_{outdom} \mathcal{L}_i^{outdom} + \delta_{illsolu} \mathcal{L}_i^{illsolu} + \delta_{idesolu} \mathcal{L}_i^{idesolu})$$

$$L_i^{illroot} = \begin{cases} 0 & \text{if } q^{pre} \geq 0 \\ -q^{pre} & \text{if } q^{pre} < 0 \end{cases}$$

$$L_i^{outdom} = \begin{cases} 0 & \text{if } q^{pre} \in [-1, 1] \\ |q^{pre}| - 1 & \text{if } q^{pre} \notin [-1, 1] \end{cases}$$

$$L_i^{illsolu} = \begin{cases} 0 & \text{if } q_n^{(k)} \in [-\pi, \pi] \\ |q_n^{(k)}| - \pi & \text{if } q_n^{(k)} \notin [-\pi, \pi] \end{cases}$$

$$L_i^{idesolu} = \begin{cases} 0 & \text{if } \exists n \in [1, 6], \forall k \in [1, 8], q_n^{(k)} \in [-\pi, \pi] \\ L_i^{illsolu} & \text{otherwise} \end{cases}$$

3.3 实验结果

解决了上述三个问题后，我们来看训练结果，即我们提出的框架是否有效。图 13 展示的是关于整个 RPSN 训练性能的实验。我们与传统的神经网络，即 MLP 感知机以及 RPSN 的两种方法进行了对比。

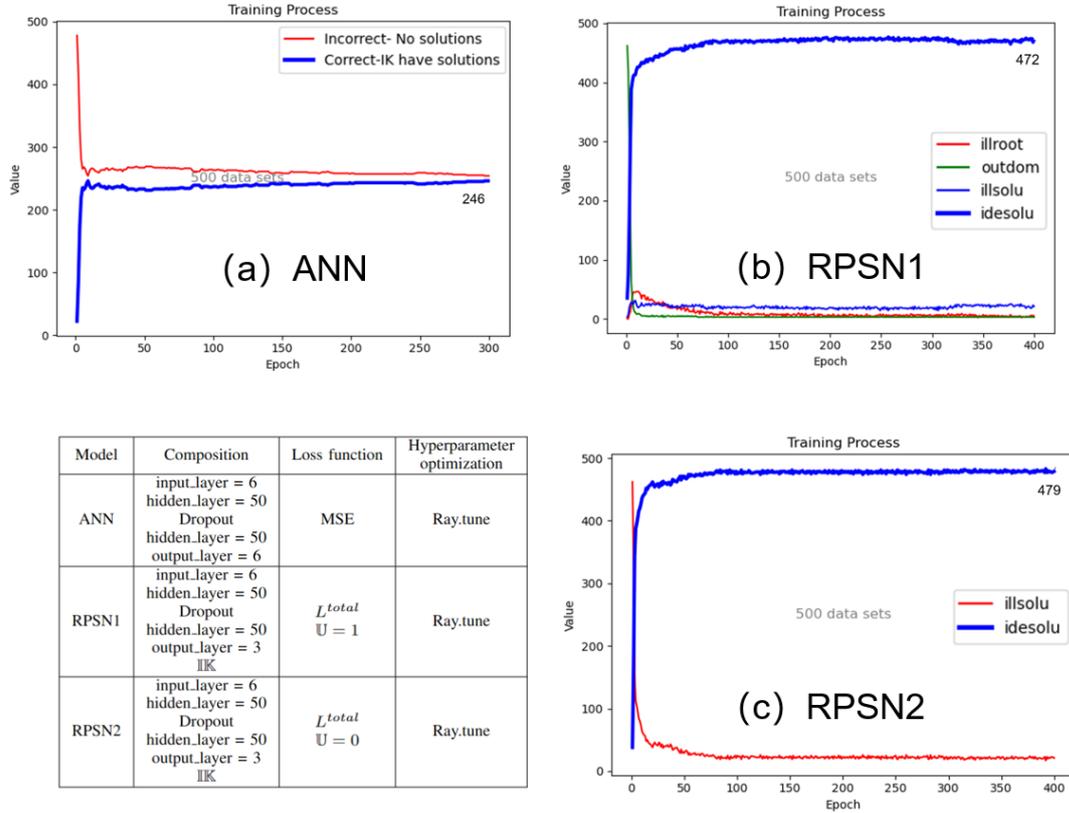


图 13 (a)ANN 的训练精度收敛在 53%，而 RPSN 在(b)(c)两种情况下都高于 94%。

在数据集比较小的情况下，当数据量较少时，传统的神经网络无法达到较高的预测准确率，通常在 53%左右就会停止下降。这可以理解为，我们所提供的 500 组数据集，网络可能已经学到了数据集里的映射关系，即它已经拟合到了这种规律，但并没有真正学习到逆运动学或碰撞的物理逻辑关系。而对于 RPSN，在采用两种不同的损失函数情况下，RPSN 都能保持超过 94%的准确率。

我们选取了从 300 组到 3200 组不同数据量去训练神经网络，如图 14 所示，传统的神经网络在最初阶段无法学到任何规律，因为数据量过小，它甚至无法学习到基础的映射关系。随着数据集的增加，其表现有所改善，但仍未能超过 50%的准确率。RPSN 模型，在少量数据下就能超过 80%，最终甚至超过了 95%，显示出其在数据方面的明显优势。

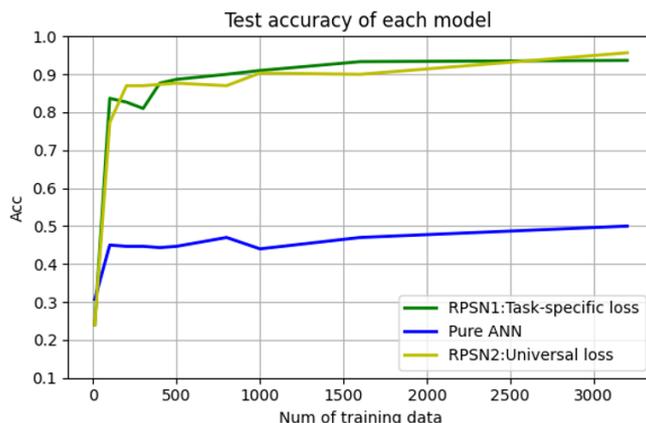


图 14 不同数据量下训练后的测试精度曲线，RPSN 模型的测试准确率高达 **95%**。

我们也针对之前提到的随机化问题进行了相关的性能测试。如图 15 所示。

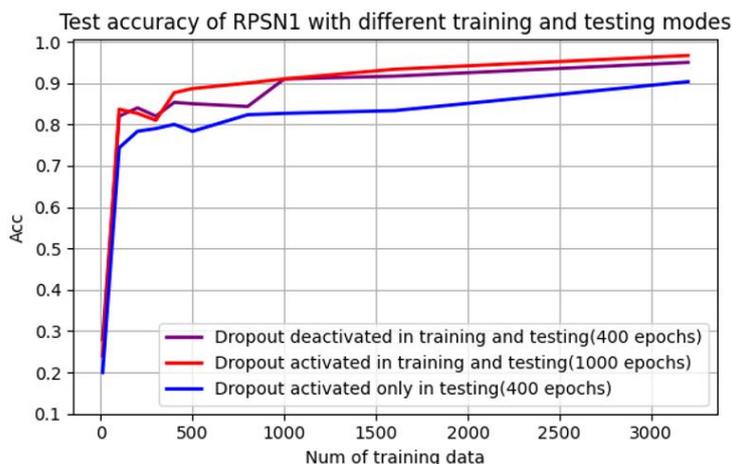


图 15 引入 Dropout 随机化后，RPSN 的单个预测精度提高到 **96%**。

我们选取了 RPSN 模型的 RPSN1 版本，在不同阶段引入了 dropout。一共有三组实验：第一组是在训练和测试过程中都不引入随机化，网络是固定的，相同的输入得到相同的输出。第二组在测试和训练阶段都引入随机化，而第三组只有在测试的时候引入随机化，在训练的时候不引入随机化。可以看到，引入随机化后，RPSN 的单个预测精度达到了 **96.67%**，但需要更长的时间和训练次数才能达到收敛。

我们关注的是所提出方法在数据效率上的优势。RPSN 平均只需要 1.28 次就能实现理想位置的 100%成功生成，远低于随机采样的 31.04 次，速度最高提升了 24.24 倍。我们还在真机硬件平台上进行了实际部署，设计了四个不同的真机实验场景：（1）从任何位置以一定距离接近目标螺栓将其拆卸、（2）在当前

位置的安全工作空间内拆卸螺栓后，移动到适当的位置继续拆卸下一个螺栓、
(3) 系统自主运行，执行大范围持续的自主拆卸任务、(4) 模拟工厂环境下的多次调度、长距离连续拆卸。总计超过 80 次真机实验，RPSN 能够在实际拆卸场景中准确采样位置，成功率高达 100%。

3.4 项目总结

我们利用神经符号人工智能技术，赋予了机器人在移动领域中形成初步直觉的能力。

移动机器人的运动规划通常需要经过多次随机抽样和运动学模拟迭代，才能精确定位目标底盘的位置。为解决这一问题，我们基于神经符号人工智能原理设计了机器人位置猜测网络 RPSN。RPSN 由一个位置预测网络和一个可微分运动学计算引擎组成，可以根据工作对象的姿态直接输出高度可信的底盘位置，并保证机器人运动学在此位置下有合法解。我们在退役动力电池拆解应用场景和其他工业模拟场景中部署了 RPSN。实验表明，RPSN 在初步确定目标底盘位置时的准确率高达 96%，迭代采样可持续提供 100% 的有效位置，平均采样次数为 1.28 次。

在这项研究中，我们首次提出了一种方法，通过可微分编程将机器人的运动学模型整合到神经网络的计算图中。这一创新不仅重新定义了传统神经网络的训练范式，还显著降低了训练过程中对大量数据的依赖。尤为重要的是，这种方法能够将知识约束有效融入神经网络，适用于所有松散耦合的移动机器人运动规划场景。这为我们未来的研究开辟了新的方向，即通过结合可微分的经典算法与神经网络，利用现有的成熟知识快速引导和优化神经网络的学习过程，从而实现更可靠、高效的推测与决策。

四、机器人直觉的未来研究

未来的研究大致分为四个部分。

首先，更多物理约束的微分编程。我们已经将运动学可优化的方法融入到网络训练过程中，让系统学习到运动学的约束。这种知识仅能提供在特定场景下完成特定动作的能力。因此，我们后续会考虑将碰撞检测和其他动力学约束，通过适当的方法整合到我们的系统中。

其次，损失定义的问题。我们未来的目标是将更多微分编程算法融入这个

框架内，损失函数将更加复杂，将从凸优化问题转变为非凸优化问题。非凸优化问题的解决并没有一个明确的解决范式。非凸优化情形下如何设计损失逃离鞍点？当前的方法可能包括在不同的优化器中添加动量或其他方法，或者通过设计损失函数来优化神经网络反向传播过程。未来，我们也将考虑如何把当前的软约束转化为一种类似硬监督的机制，如同数据集一般严格，但同时又能学习到基础规律的这种强监督网络。

第三，注重于不同场景下的迁移学习。例如，在路径规划中，即机械臂移动时，需要考虑其全部的本体和关节如何进行轨迹规划。通过特定的思想，使它具备相应的直觉。神经符号人工智能的一个显著优势在于，一旦某个任务被成功实现，其原理就可以迅速应用于其他领域。这意味着，不仅限于拆卸螺栓这一任务，在任何领域，一旦方法被证明有效，就可以快速扩展应用。

第四，基于神经符号 AI 的机器人可微任务与运动规划框架。我们需要思考，如何利用可微分的先验知识，包括运动学、动力学、逻辑和稳固性模型，通过融合历史数据（经验）与知识，从而实现对任务完成方式的高准确性猜测，实现更强的类人直觉。。

在最后，我想分享我们对智能系统的几点思考。判断一个系统，无论是人工构建的还是生物体，是自动化系统还是智能系统，我们认为最根本的标准在于它是否具备智能系统的特性。

首先，智能系统能够理解和感知物理世界，即通过传感器获取信息并理解周围环境。

其次，智能系统具有记忆和回忆的功能，这意味着它们能够学习。例如，如果今天执行某个动作效果不佳，智能系统应能记住这一错误，并从中学习如何更有效地执行该动作。

第三，智能系统具备分析、猜测和推理的能力。分析是指系统能够理解任务与运动规划之间的关系；猜测则涉及直觉层面的快速判断，是推理规划的一部分。

第四，智能系统能够采取行动，即执行具体的任务或操作。

最后，智能系统能够反思其行为的结果，并在此基础上进行持续学习，这是一个验证和优化的过程。这些特性共同构成了智能系统的核心，使其能够在

复杂多变的环境中表现出卓越的适应性和智能。

当前，大语言模型因其生成式特性而广受关注。这些模型不仅为运动任务和规划提供了新的思路，还引发了值得深思的问题。例如，自回归生成式 LLM 是不是只能局限于语言直觉而不能实现 AGI？机器人直觉是否能为世界模型的构建贡献一些力量？这些问题留给大家。

神经符号AI，赋能绿色制造的人工智能引擎
<https://www.nsaihome.org.cn>



NSAiHome
神经符号人工智能社区

（责任编辑 曹晓舟，审核 刘永光）