graspit-py：在位姿和摩擦不确定性下进行抓取规划的 GraspIt! Python 接口

Clinton Enwerem[a,*]、John S. Baras[a]、Calin Belta[a]

[a]马里兰大学电气与计算机工程系及系统研究所，美国马里兰州大学公园市 *Paint Branch Drive 8223* 号，邮编 *20742*

arXiv:2611.29011v1 [cs.RO] 21 Nov 2026

摘要

在多指抓取仿真中，位姿误差可能改变手部闭合时形成的接触点，而摩擦不确定性则会改变这些接触点所能承受的载荷。现有的 C++API 和 ROS 1 服务提供了以编程方式访问 GraspIt! 的能力，但在 Python 中使用时需要原生绑定或服务传输层，并且需要针对具体应用，将接触几何与不确定性采样进行集成。用于此类研究的 Python 工作流必须将原生闭合和接触点与调用方定义的力旋量评分、不确定性样本及命令重建相衔接。我们提出了 `graspit-py`，一种面向 GraspIt! 原生操作的 Python 接口，可提供这一工作流。该接口提供原生场景访问、接触力旋量分析，以及在位姿和摩擦不确定性下可配置的接近规划。抓取记录将相对于物体的手掌位姿和闭合后的独立坐标，与标称接触点及在采样的位姿和摩擦条件下评估的质量评分关联起来。我们的 MuJoCo 导出器可推导串联铰链链和仿射关节耦合（包括非零参考角），并保留各个碰撞组件。在五个手部模型的 75 种构型中，源顶点集与导出后的掌部坐标系顶点集之间的最大差异约为 1.3 µm，低于 100 µm 的几何容差。我们通过受控消融实验分离了参考角和碰撞组件的影响，并使用独立的笛卡尔力线性规划验证了 1,920 项方向性任务力旋量容量计算。在一项常见的 GraspIt! 工作负载中，通过直接 Python 实现的重置–闭合–评分–状态循环中位延迟，比通过持久化 ROS 1 服务低 10.7%。可执行的导出–规划–重建工作流，以及针对六种手部模型的抓取示例，展示了接触、评分和关节表示形式如何支持数值分析与命令重建。

关键词： GraspIt!、MuJoCo、抓取规划、接触力学、模型导出

源代码： https://github.com/coenwerem/graspit-py

1. 引言

GraspIt! 通过为机器人手部设计和抓取合成提供统一环境，支持相关研究，包括关节式手部建模、物体抓取获取和接触力旋量分析 [1]。对于抓取生成器或手部设计的比较，仿真器提供一致的力学模型，同时由研究人员指定提案和接触目标函数。例如，学习型提案生成器可以提供掌部位姿（手掌的位置和方向）以及用于闭合的手指构型，随后即可使用自定义的力旋量目标函数评估所产生的接触点。场景包含加载的手和物体模型、它们的放置位置以及手部构型。在此类研究中重复使用 GraspIt!，需要以编程方式控制其场景和闭合操作，并访问接触几何形状。这些功能使其抓取力学能够与数值优化和数据分析结合，用于自动化实验。

位姿和摩擦不确定性需要对抓取执行不同的操作：接近位姿中的误差可能导致接触点移动或消失，因此需要重新执行闭合；而摩擦的变化则会改变允许的接触力旋量

*通讯作者：Clinton Enwerem。ORCID：0000-0002-4239-2941。技术报告。电子邮箱：enwerem@umd.edu （Clinton Enwerem），baras@umd.edu （John S. Baras），calin@umd.edu （Calin Belta）

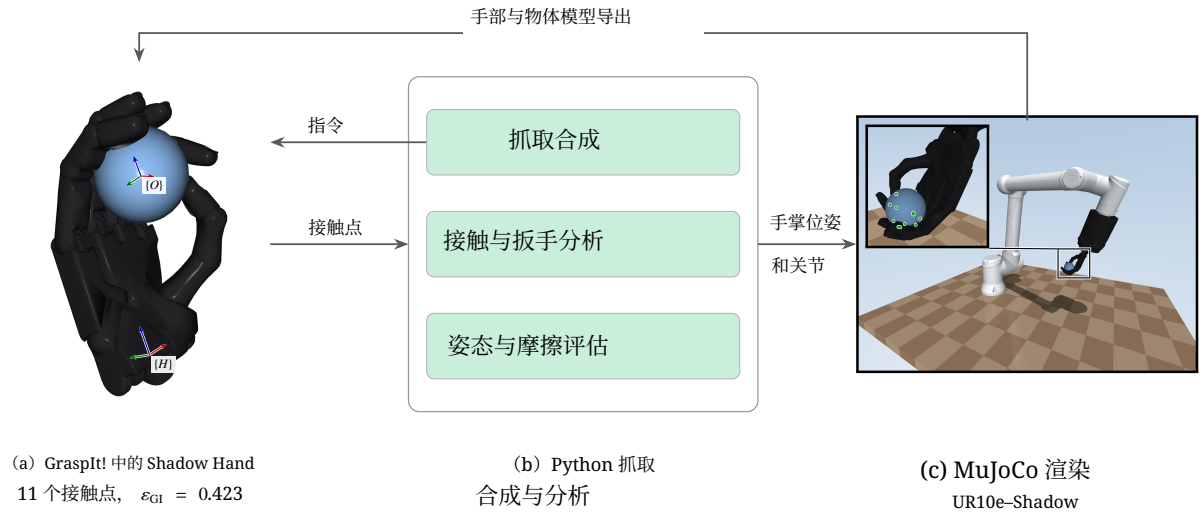


图 1：**Python 抓取工作流中的 GraspIt!**。(a) Shadow Hand 抓取一个直径为 60 毫米的球体，通过基于碰撞的闭合获得 11 个接触点，原生 Ferrari–Canny质量为 ε_{GI} = 0.423。视觉网格摆放在闭合关节构型下，掌部坐标系为 $\{H\}$，物体坐标系为 $\{O\}$。(b) 用于抓取合成、接触力旋量分析以及位姿/摩擦评估的 Python 操作。(c) 在 MuJoCo 中渲染的 UR10e–Shadow 场景，采用 (a) 中相对物体的手部位姿和关节构型。插图展示了抓取的掌侧视图。绿色空心圆标记投影后的 GraspIt! 接触位置，包括被遮挡的位置。箭头标出用于模型导出、抓取指令、闭合接触查询以及重建手掌和关节构型的接口操作。

对于每个由此产生的接触集。使接触位置和法向量独立于评分程序，研究人员便可将自己的力旋量模型和质量指标应用于这些闭合抓取。点云库（PCL）在点云处理方面采用了类似的分离方式，在提供可复用数据结构的同时，还提供模块化的滤波器和估计器 [2]。ROS 1 服务接口及其 Python `graspit_commander` 客户端向外部程序公开模拟器操作 [3], , 而 C++API 则提供场景访问和特征抓取规划 [4]。我们推出了 `graspit-py`，可在 Python 实验中直接访问原生 GraspIt! 的闭合和接触分析功能。其绑定将原生引擎连接到 Python 数值程序，无需 ROS 1 服务层或特定于应用的原生扩展。该接口将这种访问能力与数值力学和不确定性工具相结合。我们的规划器会在位姿扰动下重新执行闭合，利用数值方法评估摩擦样本，并保留得分样本，使研究人员能够分析抓取敏感性，并选择适合其研究的统计排序方法。

在任务仿真中使用规划抓取的研究，也需要将用于抓取分析的手与用于任务的机器人模型关联起来。两者在关节结构或碰撞几何体上的差异，即使两个模型表示的是同一只实体手，也可能改变可用于支撑物体的接触点。建立这种对应关系，需要构建关节坐标映射，并比较附加到对应连杆上的碰撞组件。我们通过从编译后的 MuJoCo 模型导出手部和刚体几何体 [5] ，并保留重建源模型中闭合构型所需的关节对应关系，来支持这种关联。图1 展示了由此形成的研究工作流：GraspIt! 提供闭合接触点，供 Python 进行抓取合成和数值评估；相对物体的手掌位姿和关节映射则支持在 MuJoCo 中重建。研究人员由此可以利用 GraspIt! 的力学模型研究不同的候选方案和接触目标函数，并将选定的构型带入任务仿真器进行后续分析。

1.1. 相关工作

抓取生成和抓取评估对模拟器接口提出了不同的要求。特征抓取规划会在 GraspIt! 中搜索手指姿态的低维子空间。 [6], 而 FRoGGeR 则使用可微的质量度量来优化指尖位置。 [7]学习方法会根据物体几何和部分视图生成多指配置或抓取位姿分布。 [8, 9, 10, 11]抓取分类体系会对手部姿态进行组织 [12]; GraspLDM、CenterGrasp 和 REGNet 等图像条件方法与点云方法则提供

几何提议 [13, 14, 15]。数据驱动的抓取合成进一步阐明了生成提议与评估提议之间的区别。 [16, 17]我们的接口接受任意掌部位姿和独立坐标。所提供的规划器使用原生接触仿真，评估接近方向、滚转、预成形以及启用闭合的坐标的各种组合。因此，提议生成可由调用方提供，同时仍可对由此产生的接触点进行数值评估。

接触力旋量分析用于确定哪些外部力旋量（即力-力矩对）能够由接触点处的力平衡。解析抓取度量区分了力闭合（抵抗所有方向力旋量的能力）、可行接触力，以及平衡任务载荷的能力 [18, 19, 20, 21, 22]。不确定的摩擦和偏移接触点会改变力旋量凸包及其力闭合特性 [23]。基于姿态的重排序用于评估抓取对物体姿态误差的敏感性 [24], PONG 则界定表面法线不确定时力闭合的概率 [25], , 而内在鲁棒性结果将摩擦锥不确定性与 Ferrari–Canny 质量及最小权重质量关联起来 [26]。对于给定力预算和力矩缩放的力闭合抓取，Ferrari–Canny 质量是归一化力旋量凸包内包含的最大、以原点为中心的欧几里得球的半径。Grasp’D 可通过仿真接触进行微分[27], , 而弹性指尖稳定性可通过势能评估 [28]。这些量描述了不同的物理特性。我们将原生接触几何与数值力旋量计算分开呈现，以便明确指定评分所用的摩擦模型、力预算和任务载荷。

不确定性感知规划还取决于不确定量的表示与更新方式。部分可观测决策过程区分状态不确定性与规划器可用的观测 [29]。编码器反馈和粒子滤波可以降低平面抓取序列中的位姿不确定性 [30], , 而基于假设的信念规划则在接近抓取轨迹中使用接触观测，以应对非高斯姿态不确定性 [31]。变分信念模型表示接触参数的不确定性 [32], , 而风险敏感增量采样则将风险准则应用于随机最短路径代价 [33]。FIRMGrasp 根据分布的不利尾部计算摩擦系数，并据此评估力闭合裕度 [34]。在当前规划器中，决策是一个接近指令，采样量则是闭合后的力旋量质量。位姿扰动会触发新的闭合抓取，并针对由此产生的每个接触集评估摩擦值，随后计算下尾统计量。

要传递此类指令，规划与执行模型之间必须保持一致性。机械臂运动学根据关节坐标确定连杆位姿 [35]; 即使坐标值一致，铰链轴或参考坐标系发生变化，也会改变指尖位置。基于 ROS 的 urdf2graspit 转换器可将 URDF 运动学和网格映射为 GraspIt! 描述 [36]。我们的导出器以编译后的 MuJoCo 模型为基础，并保留仿射关节对应关系以及所选几何表示中使用的各个碰撞组件。由此生成的关节与连杆映射使得在指定构型下可以量化对应关系，并支持在 MuJoCo 中重建规划得到的相对于物体的抓取。

接触几何也不同于支配抓取获取与保持的反馈和动力学。基于闭环深度的抓取方法和学习到的动作价值会利用观测或试验结果更新指令 [37, 38]。触觉反馈可引导抓取获取和重新抓取 [39, 40], 调节作用力或稳定单个手指[41, 42], , 并为转移提供稳定性预测器 [43]。动态抓取评估会考虑位姿不确定性下重复抓取获取和提举过程中发生的物体运动与接触变化 [44]; 柔顺抓取合成还会对形状不确定性下的手指阻抗进行建模 [45]。这些方法说明，闭合运动和接触响应与名义力旋量评分同样重要。在装配中，柔顺富接触运动 [46]和冗余机械臂运动—力控制 [47] 作用于被抓取物体，而 Orbit 等仿真系统则提供机器人学习环境 [48]。该包装器会向这些外部规划、仿真和控制组件提供接触几何与重建指令。

1.2. 贡献

我们提供了一个 Python 接口，用于获取原生 GraspIt! 闭合抓取，并在位姿和摩擦不确定性下评估其接触力旋量。场景、手、物体、接触点和连杆查询功能，辅以数值运动学和力旋量空间工具（第 2 节）。对于具有串联转动手指链、且铰链通过固定比率和偏移耦合的 MuJoCo 手模型，我们推导出仿射关节映射（包括参考角偏移），以及保留各个碰撞组件所需的刚性坐标系变换（第 2.5 节）。抓取记录将接受的掌部姿态和闭合后的独立坐标与标称接触点及位姿—摩擦评分数组关联起来，从而明确列出评分输入和重建数据（第 2.6 节和第 2.7 节）。我们通过互补的几何、数值和运行时比较评估了该接口：逐连杆对应关系、受控导出器

消融实验、与独立笛卡尔力公式的一致性，以及多手闭合和不确定性采样下的运行时间与常驻内存扩展性（第 4 节）。

2. 软件架构

我们将原生场景操作与基于数组的分析分离，使研究人员能够将同一闭合模型用于不同的候选生成器和数值目标。原生 GraspIt! 提供手部运动、碰撞查询和接触几何形状。规划模块将这些操作与位姿扰动和依赖摩擦的评分相结合，而模型转换则提供重建所需的几何形状和源关节对应关系。图 2展示了这些数据依赖关系，表 1 列出了公共模块。以下接口明确说明了坐标、接触和评分表示形式。

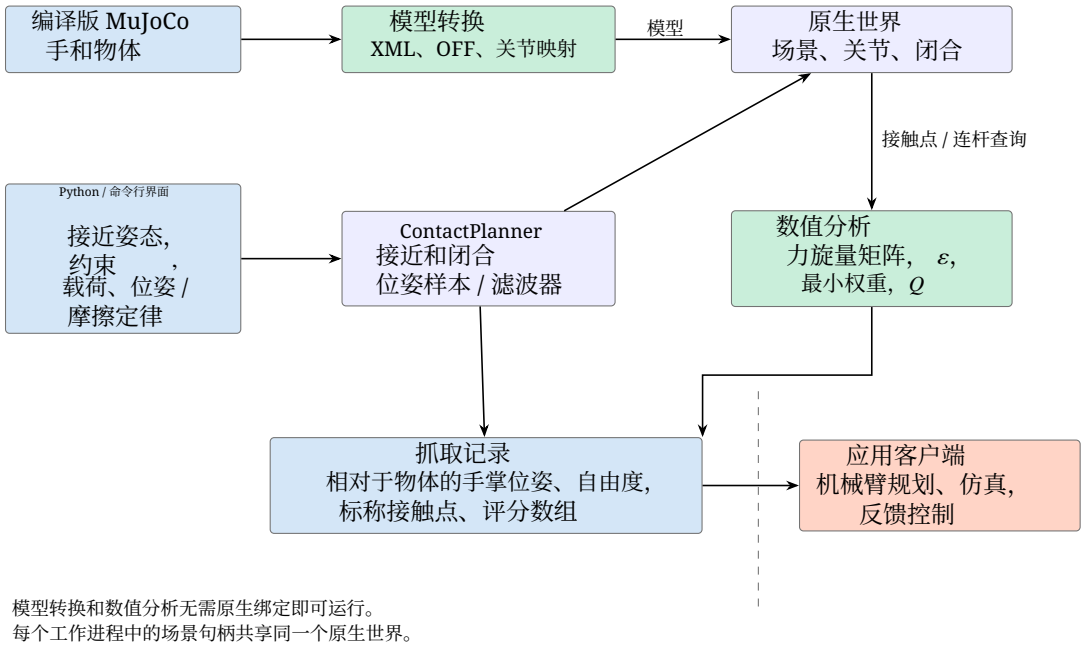


图 2：**包装器架构**。 模型转换将手部和物体几何形状以及关节映射提供给原生 GraspIt!。候选命令及其位姿扰动经过接近和闭合，为数值力旋量分析生成接触点。抓取记录保留已接受的相对于物体的手掌位姿、闭合后的独立坐标、标称接触点和采样评分，以供后续选择或重建。

2.1. 场景状态与手部运动

运行闭合实验之前，调用方会初始化 GraspIt! 会话并载入所需的手部与物体模型。必须能够导入原生扩展；导入绑定时，如果扩展缺失，就会引发`ImportError`。调用方调用无参数的`initialize()`，该函数返回`None`；随后创建`World`句柄，并在载入机器人、物体或世界描述文件之前调用`World.init()`。环境变量`GRASPIT`用于指定包含 GraspIt! 手部模型、物体描述文件和世界文件的目录；Qt 使用`DISPLAY`和`QT_QPA_PLATFORM`。创建 Qt 应用程序时，如果这两个 Qt 变量均未设置，初始化过程会选择`offscreen` 平台。每个 Python 进程只会执行一次原生初始化，因此重复调用会复用已初始化的核心。已初始化的`World` 句柄共享 GraspIt! 主场景；若要使用独立场景，则需要单独的工作进程。

手部姿态是$u \in \mathbb{R}^m$，即m 个按已加载模型顺序排列的弧度制独立坐标组成的向量。一个 u 中的值可以控制多个铰链，例如第 2.5 节所述的耦合会使最靠近手掌的关节与指尖关节一同闭合。手掌位姿采用

表 1：包接口

公共模块	输入	输出
core.bindings	机器人和物体模型、手部姿态、关节坐标	闭合、物体坐标系接触、原生质量、碰撞和连杆查询
转换。	编译后的 MuJoCo 模型和已选择数据	GraspIt! XML 和 OFF 网格，以及
mjcf_手	手掌或物体子树	用于手部导出的仿射源关节映射
规划	掌坐标系几何形状、接近姿态、预成形、闭合子集、约束，以及不确定性参数	导出 包含相对于物体的掌部位姿、独立坐标，接触点和评分的排序记录
运动学	GraspIt! 机器人 XML 和独立坐标	连杆正向运动学
抓取。质量，风险	接触点和载荷方向，或加权评分样本	力扳手空间度量、任务载荷乘数和尾部平均值
utils.geometry，utils.urdf	几何数据或刚性 URDF 对象以及可选依赖	变换、距离或 OFF/XML 对象模型

表2。直接位姿和姿态设置器会建立所请求的状态。姿态赋值会应用原生坐标限值处理，但不进行碰撞检查。原生场景检查 `World.no_collision()` 当且仅当已加载的世界物体之间没有相互穿透时返回 `True` ，同时遵循原生碰撞过滤规则。

受控闭合则使用 移动_当前_手_自由度_至_接触点，并提供目标坐标向量和可选的逐坐标步长。原生小步碰撞检查控制这一运动。其跟踪变体会返回一个布尔值，报告是否至少成功执行了一步原生移动，并在每个步进信号处记录独立坐标和碰撞计数。这些记录展示了闭合姿态的变化过程，但不会为各步指定时间或力。原生 自动_抓取操作使用手配置的默认速度；将已选择的速度设为零，即可将相应坐标保持在预成形状态。规划器利用此机制执行闭合子集，并在闭合后恢复原始速度。表 3 总结了相应的输入和输出约定。

2.2. 接触几何与力查询

更改位姿或姿态并闭合手后，调用方使用 抓取的 `compute()` 刷新数据。其物体_接触_geometry() 查询会返回物体侧接触位置 p_i （单位为毫米）以及内向法线 n_i，二者均以物体坐标系表示。这种通用表示形式结合物体坐标系下的质心c，可用于数值力旋量分析。矩阵构造需要摩擦系数和力矩尺度 ρ，并生成第 W_m 2.6 节定义的力旋量列。基于数组的模块独立于原生场景状态运行，使用 NumPy 和 SciPy [49, 50]。原生质量 $\varepsilon_{\mathrm{GI}}$、重新计算的 Ferrari–Canny 质量 ε以及最小权重最优解均保留各自的定义；最小权重例程还会返回一个满足秩条件的力闭合布尔值。

准静态力查询需要单独进行原生求解。调用方检查 `compute_quasistatic_wrenches()` 和可用性标志，或者使用可选的法向力值，明确保留缺失的结果。调用 `compute()` 会刷新接触点并清除缓存的准静态解。对于刷新后的接触集，如果尚未尝试求解，力查询会触发新的求解。原生接触力旋量的力分量位于力矩分量之前，并保留其原生单位和接触坐标系轴。对于一个已求解的受压接触，设 $f_{n,i} > 0$ 和 $f_{t,i}$ 分别表示该坐标系中的法向力分量和切向力分量。所需的摩擦系数为

$$\mu_{i,\mathrm{req}} = \frac{\|f_{t,i}\|_2}{|f_{n,i}|}. \tag{1}$$

将 $\mu_{i,\mathrm{req}}$ 与接触系数 μ_i 进行比较，可以检验求解所得力的摩擦需求。对于法向力和摩擦系数均高于各自数值零阈值的受压接触点，返回的摩擦需求指标为 $\min\{\mu_{i,\mathrm{req}}/\mu_i, 1\}$。数值较小表示相对于库仑极限而言切向力较小，而数值为 1 则表示需求达到或超过该极限。

表 2：坐标与数据约定

表示形式	坐标系与单位	接口约定
原生手掌姿态	世界坐标系；平移单位为毫米	四元数（w，x，y，z）；姿态查询包含一个嵌套的矩阵_行_主序包含四行四列
抓取记录手掌姿态	物体坐标系；平移单位为毫米	T_物体_手_毫米 映射手掌坐标转换为物体坐标
独立坐标	加载的手的原生顺序；弧度	共享驱动器可以控制多个铰链
物体侧接触	物体坐标系；位置单位为毫米	位置_毫米 和向内的力法向
原生连杆查询	世界坐标系；顶点单位为毫米	位姿用于标识运动链、连杆和名称
XML正向运动学	掌部坐标系；XML距离通常 mm	连杆_位姿返回齐次按运动链和连杆索引的变换
几何工具	调用方提供的坐标系和长度单位	四元数（x、y、z、w）；网格距离为无符号

2.3. 几何与运动学查询

坐标系复合将原生几何、导出的运动学数据和源模型重建联系起来。设 $\{W\}$、 $\{O\}$ 和$\{H\}$ 分别表示源世界坐标系、物体坐标系和掌部坐标系，设 $\{L\}$ 表示连杆坐标系。所有平移均使用相同的长度单位表示，

$$ {}^{W}T_{L} = {}^{W}T_{O}\,{}^{O}T_{H}\,{}^{H}T_{L}(u). \tag{2} $$

原生连杆查询会返回原生场景世界坐标系中的位姿和顶点。纯正向运动学读取导出的 Denavit–Hartenberg 描述，并根据独立坐标 u 计算手掌坐标系下的连杆位姿。为便于比较，调用方通过应用每个场景的逆物体位姿，将原生连杆位姿和源连杆位姿表示为相对于各自物体坐标系的位姿。共同的物体坐标系对应关系允许将其与 ${}^{O}T_{H}\,{}^{H}T_{L}(u)$ 进行比较，所依据的是运动链和连杆标识。几何工具会变换点，并在输入长度单位下计算无符号的点–三角形距离或点–网格距离。原生位姿使用标量优先四元数（w、x、y、z），而几何工具使用标量末位四元数（x、y、z、w）（表 2）。在这两个接口之间转移方向时，调用方会重新排列组成部分。导出清单提供源关节名称、驱动关节索引和仿射系数，而自由度_从_qpos 通过关节名称和位置地址提取独立坐标。

可选的刚体物体工具urdf_到_graspit_物体_xml会将 URDF 可视网格组合为 OFF 几何形状，并写入 GraspIt! 物体 XML 描述。调用方提供长度比例和以克为单位的质量，以便为原生加载和闭合研究准备刚体物体。这种视觉网格预处理不同于第2.5节中编译后的 MuJoCo 手部和物体导出器，后者使用源模型的碰撞几何体。

2.4. 接近规划与抓取记录

规划器会枚举接近方向、滚转角、预成形和闭合子集的各种组合。调用方可以提供显式接近姿态，也可以使用斐波那契方向和等间距滚转角；每个闭合子集都是一串整数索引，指向独立坐标向量。该闭合_集合配置会将这些子集枚举为备选抓取指令。纯手掌_位姿_用于_接近的函数会构造规划器的静止ff 变换，供外部可达性检查使用。每个原生接近都会重置预成形，并在接受的放置位置闭合之前，逐步执行离散碰撞检查。规划器会保留物体的构造时位姿用于这些计算。最小接触和允许接触区域测试会应用于名义闭合和扰动闭合。支撑平面和禁入框测试会检查最终名义手部顶点，因此其范围仅限于该放置位置处采样的顶点。任务行提供物体坐标系中的施力点和平衡力方向，并在评分前分别进行归一化。第2.6 节将详细介绍接近、载荷和不确定性计算。

返回的字典 GraspCandidate.to_dict()会保留尝试过的命令及其名义闭合（表4）。NumPy 数组会转换为 Python 列表。对于 n^{ξ} 个扰动接近方向和 $n_{\mu} > 0$ 个摩擦节点，评分数组的形状为（$n_{\xi}+1$，n^{μ}），表示方程 16中的经验分布。采用从零开始的索引时，第零行表示名义值；名义标量字段使用列 $\lfloor n^{\mu}/2 \rfloor$，即当 n^{μ} 为偶数时的上方中间列。所有评分均为无量纲值

表 3：已选择的可调用合约

可调用函数族	调用方提供	结果或状态更改
世界载入与设置器	模型路径、世界坐标系中的手掌姿态，独立坐标	共享原生场景状态；加载成功布尔值
移动_当前_手_自由度_至_接触点和跟踪变体	目标坐标和可选步长尺寸	碰撞感知坐标运动；可选坐标/碰撞计数轨迹
Grasp.compute，物体_接触_几何形状	加载的手、选定物体、当前闭合	更新后的抓取和物体坐标系接触字典
准静态求解和力查询	当前抓取	求解状态、可用性、接触坐标系力旋量和法向力诊断
构建_抓取_矩阵，最小化_权重_力_闭合	公共坐标系接触点、中心、摩擦、力矩尺度	力旋量矩阵；最小权重最优解和力封闭谓词
载荷_运动链、连杆_位姿	机器人XML和独立坐标	按以下内容标识的掌部坐标系连杆变换运动链和连杆
下_尾部_均值	评分值、尾部质量、可选权重	加权下尾质量均值
GraspCandidate.to_dict	现有的抓取记录对象	手掌位置、闭合坐标、标称接触点、采样评分

在文档规定的归一化方式下，β表示下尾质量。没有任务载荷时，任务字段为 None。在固定种子和配置下，依赖命令的随机流可确保其样本保持不变，且不受枚举顺序影响。调用方会将搜索配置与记录一并保留，以确定摩擦网格、姿态分布和聚合参数。

2.5. MuJoCo 模型导出

原生闭合需要 GraspIt! 提供手和物体的描述，包括它们的几何形状和关节结构。我们将用于转换的 MuJoCo 模型称为源模型；其内存中的MjModel 表示形式是编译模型。*MuJoCo*源场景包含手、物体及周围场景几何体。手部导出器接收此编译模型和手掌刚体的名称。连杆是手的一个刚性部分。如果手指的运动连杆依次相连且没有分支，则形成一条串联链。每个受支持的运动连杆都有一个转动关节，因此其相对运动是绕固定铰链轴旋转 [35]。手由连接到手掌的若干此类运动链组成。无关节刚体会刚性连接到距离最近的已导出祖先，其碰撞几何体以该祖先的连杆坐标系表示。

多个铰链可以共用一个独立指令。用$u \in \mathbb{R}^m$ 表示独立坐标，用$q \in \mathbb{R}^n$表示以弧度为单位的铰链角。关节i的参数化形式为

$$q_i = r_i u_{j(i)} + b_i, \qquad q_i^{\min} \le q_i \le q_i^{\max}. \tag{3}$$

此处$j(i)$表示驱动关节，r_i 是无量纲比率，b_i 是角偏移。独立铰链的$r_i = 1$和$b_i = 0$。MuJoCo 等式约束通过相对于参考角的位移 [51]，将从动铰链i 与独立铰链j 耦合起来。对于仿射等式约束，转换为绝对角的公式为

$$\begin{aligned} q_i - q_i^0 &= c_0 + c_1(q_j - q_j^0), \\ r_i = c_1, \qquad b_i &= q_i^0 + c_0 - c_1 q_j^0. \end{aligned} \tag{4}$$

导出器会读取编译模型中每个关节位置地址处的参考角，并将生成的绝对坐标映射写入清单。此映射还决定了 GraspIt! 关节的角度表达式和限位。受支持的模型类别采用与独立铰链直接仿射耦合的方式。

清单将标量耦合关系表示为全局坐标映射。令 $q \in \mathbb{R}^n$ 按固定的关节名称顺序收集源铰链角，并令 $u \in \mathbb{R}^m$ 按清单的 dof_关节 顺序包含独立坐标。

表 4：序列化抓取记录模式

Key	序列化类型、形状和单位	含义
T_物体_手_毫米	嵌套浮点数列表， 4×4；平移单位为 mm	接受的名义手掌到物体变换。
自由度_弧度	浮点数列表，长度为 m；弧度	闭合后的独立坐标。
接触点	长度为三的字典列表 浮点数列表	名义物体坐标系位置_毫米（毫米）和朝内的单位法向。
原生_epsilon	浮点数	名义姿态的 GraspIt! 原生抓取质量闭合。
名义_ε	浮点数	在中间摩擦节点处的名义行重新计算的质量中间摩擦节点。
ε_样本	嵌套浮点数列表， $(n_\xi+1)\times n_\mu$	重新计算的 Ferrari–Canny 质量在位姿和摩擦节点上的。
评分	浮点数	排序分数：epsilon_cvar无任务载荷；下尾均值 任务_样本 质量为 β 时；否则。
期望_epsilon	浮点数	对每个epsilon_样本 单元求均值。
接近_方向	浮点数列表，长度为三	物体坐标系单位接近方向。
滚转_度	浮点数；度	尝试的接近滚转角。
预成形	浮点数列表，长度为 m；弧度	尝试的独立坐标 闭合前。
闭合	整数列表或 无	从 0 到的选定的从零开始的索引 $m-1$；None 启用所有坐标。
epsilon_条件风险价值	浮点数	epsilon样本的下尾均值 质量为 β。
任务_样本	嵌套浮点数列表， $(n_\xi+1)\times n_\mu$，或 无	位姿和 摩擦节点上的任务容量样本。
名义_任务	浮点数或 无	名义行任务容量，位于 中间摩擦节点。
预期_任务	浮点数或无	对每个任务_样本单元格取均值。

Then

$$q = Au + b, \qquad A_{ij} = \begin{cases} r_i, & j = d_i, \\ 0, & j \neq d_i, \end{cases} \tag{5}$$

其中 d_i 选择铰链 i 的驱动关节，r_i 是其比率，而 b_i 是经参考值校正的偏移（4）。角度和偏移均以弧度表示；$A \in \mathbb{R}^{n\times m}$ 是无量纲的。独立铰链行包含 I_m 中对应的行，且偏移为零，因此秩$A = m$，即使从动关节的比率为零也是如此。清单中的关节名称、驱动关节索引、比率、偏移和源区间提供了构建该映射所需的数据。

源限制通过相同的仿射映射约束独立坐标。对于铰链区间 $[q_i^-, q_i^+]$ 和$r_i \neq 0$，其可行驱动区间为

$$\begin{aligned} L_i &= \min\left\{\frac{q_i^- - b_i}{r_i}, \frac{q_i^+ - b_i}{r_i}\right\}, \\ U_i &= \max\left\{\frac{q_i^- - b_i}{r_i}, \frac{q_i^+ - b_i}{r_i}\right\}. \end{aligned} \tag{6}$$

端点顺序可处理负比率。零比率铰链则需要$q_i^- \le b_i \le q_i^+$。可行的常值铰链不会对其驱动关节增加限制，而不可行的常值铰链会使完整坐标域变为空集。

对于支持的直接耦合，每个非常值铰链都约束一个独立坐标。当所有零比率铰链都满足 $q_i^- \le b_i \le q_i^+$ 时，将分配给驱动关节 j 的所有区间（包括其自身铰链区间）求交，

得到

$$\mathcal{U} = \{u : q^{-} \le Au + b \le q^{+}\} = \prod_{j=1}^{m} [\underline{u}_j, \overline{u}_j],$$
$$\underline{u}_j = \max_{i:d_i=j,\ r_i\neq 0} L_i, \qquad \overline{u}_j = \min_{i:d_i=j,\ r_i\neq 0} U_i. \tag{7}$$

当所有固定铰链均满足其限制，且$u_j \le u_j$对每个驱动关节都成立时，该区间盒可行。示例工具耦合_自由度_边界 会计算交集，并拒绝不可行或无界的结果。它使用导出的源区间；导出时，没有限制的源铰链会获得 $[-\pi, \pi]$ 。

辅助函数自由度_来自_qpos 通过从源姿态中选取相应的铰链角，恢复独立坐标。设$E \in \mathbb{R}^{m\times n}$ 为相应的选择矩阵。其单位行意味着$EA = I_m$ 且$Eb = 0$；因此

$$u = Eq, \qquad A(Eq) + b = q \quad \text{for } q \in \{Au + b : u \in \mathbb{R}^m\}. \tag{8}$$

选择操作使用每个源关节的名称和位置地址。在仿射耦合子空间上可实现精确重建。对于任意源姿态，调用方可以计算$q - A(Eq) - b$ ，以诊断从动铰链与导出关系之间的不一致。选择操作本身既不会将姿态投影到耦合子空间，也不会检验其残差。清单中的关节标识保留了对应关系，以便解释每个分量。

我们根据模型参考构型中的铰链轴构建 Denavit–Hartenberg坐标系 [35, 1]。设B将手指的基座坐标系映射到掌部坐标系。对于运动链中的连杆i ，其在掌部坐标系中的位姿为

$${}^{H}T_i(u) = B \prod_{k=1}^{i} [R_z(\theta_k(u))\, T_z(d_k)\, T_x(a_k)\, R_x(\alpha_k)], \qquad \theta_k(u) = \theta_k^0 + q_k - q_k^0. \tag{9}$$

乘积从手掌开始，沿运动链向外排列。R_x、R_z 表示旋转，T_x、T_z 表示平移。常数 θ^0 k是参考Denavit–Hartenberg角，d_k 是沿前一铰链轴的距离，a_k 是公法线距离， α_k 是绕公法线的扭转角。两个距离的单位均为毫米。导出器会在机器人XML中以度表示角度常数，并在计算得到的连杆坐标系中重新表达碰撞网格。对于方向为$\hat{}\mathrm{z}_k$、$\hat{}\mathrm{z}_{k+1}$ 的连续单位铰链轴，若 $\|\hat{}\mathrm{z}_k \times \hat{}\mathrm{z}_{k+1}\|_2 < 10^{-2}$，则平行轴构造会设置$\mathrm{d}_k = 0$，以避免公法线端点相距过远。对于近乎平行的坐标轴，这种构造可能会近似原始几何形状，因此第4.1节将通过数值方法评估所得的对应关系。

输出包括机器人XML、连杆网格，以及用于标识对应连杆和关节名称的清单。我们区分用于可视化的源网格与其碰撞表示。MuJoCo会将称为几何体的几何元素附加到刚体上，用于可视化和碰撞检测。对于参与碰撞检测的标准网格几何体，MuJoCo使用凸包，即使源表面存在凹陷也是如此 [52]。复制源三角形可保留其坐标，但可能改变闭合停止时所依据的表面。使用网格_碰撞='凸_包'时，导出器会写入编译模型中存储的凸包顶点和面。每个几何体仍是一个独立的凸组件。因此，由多个凸几何体表示的刚体会保留这些组件的并集；若对所有顶点整体求凸包，则会填满组件之间的空间。默认的'源'模式会保留源三角形表面。导出器会在两种模式下将支持的解析形状（盒体、球体、椭球体、胶囊体和圆柱体）转换为三角形网格。对象导出器使用命名的机体坐标系，并沿用质量和质心，从而提供用于评分和命令重建的物体坐标系。

为说明几何对应关系，设$K_g \subset \mathbb{R}^3$ 为几何体g在其几何坐标系中的编译后的凸包，并设${}^{B}T_g$ 将该坐标系映射到所连接的机体坐标系 $\{B\}$。设 $\{L\}$为其导出的 Denavit–Hartenberg 连杆坐标系，并具有常量变换 $C_B = {}_{B}T_L$。我们以毫米表示每个几何集合以及变换的平移部分。在导出的坐标系中，将该分量写作

$$\widehat{K}_g^L = C_B^{-1}\, {}^{B}T_g K_g. \tag{10}$$

刚体变换作用于集合中的每个点。若相应的连杆运动学满足 ${}_{H}T_L(u) = {}_{H}T_B(u) C_B$，则

$${}^{H}T_L(u)\widehat{K}_g^L = {}^{H}T_B(u)\, {}^{B}T_g K_g. \tag{11}$$

因此，对应的组成部分及其并集在掌部坐标系中占据相同的点集。刚体变换与凸化也可交换，因此转换坐标系不会改变凸包。该恒等式适用于

满足仿射关节映射（3）的构型。对于 Hausdorff近似误差为 δ_g的镶嵌基元，刚体变换会保持该误差；一组配对组成部分的有限并集的误差至多为 $\max_g \delta_g$。在几何评估中，我们比较了每种导出表示中对应的掌部坐标系顶点，并检查了碰撞距离和组件连通性。碰撞对排除会禁用指定物体之间的碰撞检查，例如相邻的手指连杆。在不同仿真器之间比较闭合行为时，这些设置必须保持一致。

清单 1：从编译场景导出一只手和一个刚体物体

```
导入 mujoco从graspit_py.convert.mjcf_手 导


模型 = mujoco.MjModel.from_xml_path("scene.xml") 手 = 导出_手(模
```

输入物体名称手掌 和 key 用于标识手部根节点和刚体物体。生成的机器人和物体路径将作为规划器的输入，而清单则提供关节对应关系，用于源模型重建。第 2.7 节中的可执行工作流会在完整的规划实例中使用这些导出内容。

清单 2 展示了导出文件l6_right.xml中的食指耦合。两个 Denavit–Hartenberg角表达式都使用 XML 中名为 d2 的独立坐标。近端角增加一个以度为单位的常数偏移量，远端角则将该坐标乘以 1.125676。我们为 RealHand L6 导出了十二个连杆网格和六个独立坐标。

列表 2：RealHand L6 食指的 GraspIt! XML 导出摘录

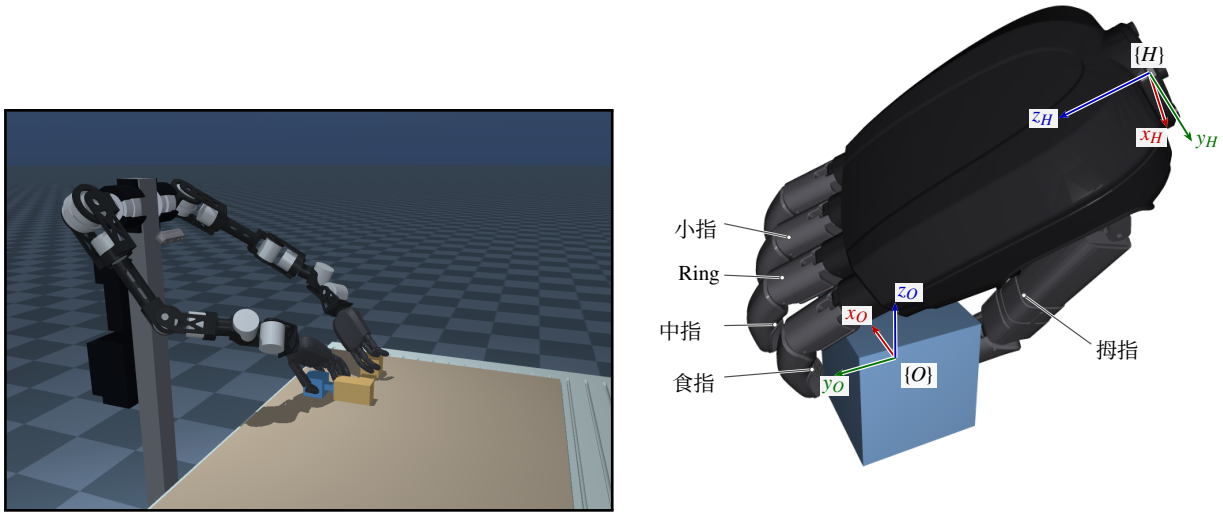
```
<运动链>
    <关节类型='旋转'><θ>d2+-88.085737930</θ></关节><关节类型='旋转'
```

我们将相对于物体的手掌位姿转移到 MuJoCo 源场景，并使用关节映射重建手指构型（图3）。

2.6. 接触分析与不确定性感知规划

规划器通过模拟接近和闭合来评估抓取指令，然后在位姿和摩擦不确定性下对产生的接触点进行评分。对于每条接近指令，手从其初始停靠位姿向物体移动，并从指定的预成形姿态开始闭合所选的独立关节。任务约束限制手指可触碰物体的位置，以及手可占据的周边区域。最小接触数量会剔除记录到的接触点过少的闭合抓取，而允许的接触区域则可保留对工具工作端的接近路径。支撑平面指定名义手部顶点相对于支撑表面的允许一侧，禁入框则表示这些顶点不得进入的夹具区域。对于中心位于k 且半宽为h的盒体，顶点v位于其排除内部区域，当且仅当 $|v_j - k_j| < h_j$对每个坐标 j均成立；边界点不满足这一严格的内部判定条件。接触数量和接触区域筛选器同样适用于扰动闭合。已接受的掌部姿态、闭合坐标、标称接触点和采样评分共同构成第2节定义的抓取记录。

位姿扰动用于模拟初始手掌位置的不确定性，以及其对闭合过程中形成的接触点的影响。对于命令 a，令 (R_a, t_a) 表示待机位姿：即朝物体移动之前，手掌在物体坐标系中的初始位姿。位姿扰动由相互独立的零均值各向同性高斯增量 $\delta p \sim \mathcal{N}(0, \sigma_p^2 I_3)$ 和 $\delta\phi \sim \mathcal{N}(0, \sigma_\phi^2 I_3)$ 构成，其中 σ_p 的单位为毫米，配置的角度尺度则从度转换为弧度。令 $\xi = (\delta p, \delta\phi)$，扰动后的待机位姿为 $(\exp([\delta\phi]_\times) R_a, t_a + \delta p)$，其中 $[\cdot]_\times$ 将向量映射为其叉积矩阵。行进方向也会发生相同的增量旋转。随后重新运行接近和闭合，得到 $C(a, \xi)$，再在每个权重相同的摩擦节点处评估质量。因此，位姿不确定性会改变接触集，而



(a) (b)

图3：**MuJoCo 源场景与导出的手**。 (a) MuJoCo 场景中的源 RealHand L6 模型处于插入状态。(b) GraspIt! 导出的右手处于规划的关键抓取姿态，并在 GraspIt! 中查询了连杆位姿。坐标轴标示掌部坐标系 $\{H\}$ 和关键坐标系 $\{O\}$。标注框指出了五条手指链。

摩擦不确定性会改变每次采样闭合时的力旋量凸包。保留的数组包含标称位姿和配置数量的扰动接近姿态。

2.6.1. 掌部坐标系与离散接近搜索

手部坐标系规定了接近方向和滚转角如何确定手掌位置。其非零接近轴a_H、手指轴 f_H以及抓取中心点 g_H 均以掌部坐标系表示，其中 g_H 的单位为毫米。对于不平行的坐标轴，构造过程会去除f_H 沿a_H 方向的分量，并构造右手正交归一基

$$
\begin{aligned}
f_H^{\perp} &= f_H - a_H \frac{f_H^{\top} a_H}{a_H^{\top} a_H}, \qquad \hat{a}_H = \frac{a_H}{\|a_H\|_2}, \quad \hat{f}_H = \frac{f_H^{\perp}}{\|f_H^{\perp}\|_2}, \\
B_H &= [\hat{a}_H \;\; \hat{f}_H \;\; \hat{a}_H \times \hat{f}_H].
\end{aligned}
\tag{12}
$$

因此，手指轴在正交化后确定掌心横向朝向。抓取中心点则确定手部中与物体中心保持规定待机距离的部分。

设d 为物体坐标系中的单位接近方向， θ 为以弧度表示的滚转角。对于 $|d_z| < 0.9$，构造函数选择$r = e_z$；否则选择 $r = e_x$，然后设置

$$
\begin{aligned}
v_0 &= \frac{d \times r}{\|d \times r\|_2}, \quad v_\theta = \mathrm{Rot}(d, \theta) v_0, \qquad B_O = [d \;\; v_\theta \;\; d \times v_\theta], \\
R &= B_O B_H^{\top}, \qquad t = c - sd - R g_H, \quad {}^{O}T_H = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix},
\end{aligned}
\tag{13}
$$

ˆ 其中$s > 0$ 为以毫米表示的待机距离，$\mathrm{Rot}(d, \theta)$表示绕d旋转。因此，$R\hat{a}_H = d$，$Rf_H = v_\theta$，且$Rg_H + t = c - sd$。改变滚转角会改变掌心朝向，同时保持抓取中心位置不变。公开的掌部坐标系_位姿_用于_接近构造函数接受以度为单位的滚转角，并将相同的变换提供给外部机械臂可达性计算。原生放置会将其与物体位姿复合，${}^{w}T_H = {}^{w}T_O \, {}^{O}T_H$，使用规划器构造时捕获的物体位姿。

有限指令库由J 个接近方向/滚转角组合、P 个预成形和C 个闭合子集组成，从而得到$N_{\mathrm{cmd}} = JPC$个抓取指令。显式接近方向已经包含其滚转角。生成指令库则将每个方向

与等间隔的滚转角组合 $2\pi j/n_{\text{roll}}$。对于n_{dir} 个方向，其斐波那契构造使用

$$\begin{aligned} i_k &= k + \tfrac{1}{2}, & \phi_k &= \arccos(1 - 2i_k/n_{\text{dir}}), \quad \vartheta_k = \pi(1 + \sqrt{5})i_k, \\ d_k &= (\cos\vartheta_k \sin\phi_k, \sin\vartheta_k \sin\phi_k, \cos\phi_k), & k &= 0, \ldots, n_{\text{dir}} - 1. \end{aligned} \tag{14}$$

因此，生成指令库的 $J = n_{\text{dir}} n_{\text{roll}}$ 。预成形用于设定初始独立坐标；闭合子集用于选择闭合过程中移动的坐标。如果未提供预成形，规划器会采用配置的张开姿态；默认的闭合选项会启用所有独立坐标。

对于每条抓取指令，接近操作都会重置预成形，并检查待机距离处的放置是否发生碰撞。以增量 $h > 0$ 为步长，后续碰撞查询会检测平移 $t + jhd$。如果场景首次在第 j 步发生碰撞，则接受的平移为

$$t_{\text{pre}} = t + \max\{0, (j-1)h - \delta\}\, d, \tag{15}$$

其中， $\delta \geq 0$ 表示以毫米为单位的间隙。如果起始位置发生碰撞，或在未发生碰撞的情况下达到行程限制，则接近操作会被拒绝。首次检测到的碰撞可能涉及场景中的任意几何体。闭合操作会重置预成形，将手掌放置在接受的变换处，检查碰撞，并执行自动抓取闭合，同时将未选坐标的速度设为零。恢复速度后，系统会刷新抓取、提取物体侧接触，并执行接触数量和接触区域测试。获准的名义闭合还必须通过支撑平面和禁入区域测试，这些测试针对手部最终顶点进行。

2.6.2. 规划器算法与计算工作量

搜索过程将接触几何的抓取获取与其摩擦相关的扳手凸包评估分离开来。以下算法会保留已接受的标称放置、闭合后的独立坐标和接触点，同时通过后续的本地操作获取扰动后的闭合抓取。

1. 枚举每个抓取指令并构建其支架架 变换。执行离散接近、闭合和标称可行性测试；若失败则跳过该指令。2. 保存标称闭合，并使用配置的种子和指令描述符初始化其随机流。3. 对于每个位姿扰动，扰动原始支架架 和行进方向，然后重复接近与闭合。对新的接触点应用接触数量测试和接触区域测试。对于失败的场景，将评分行设为零。4. 在每个摩擦节点处，对标称接触点及每个成功的扰动接触集进行评分。保留完整的 ε数组，并在任务模式下保留任务容量数组。5. 使用（/28）对所有位姿/摩擦单元格汇总配置的评分；使用保存的标称闭合和采样评分创建抓取记录。6. 按评分降序对所有接纳的记录进行排序，并返回所请求的前缀；如果未指定返回限制，则返回完整列表。

对于 n_ξ 个扰动接近姿态和 $n_\mu > 0$ 个摩擦节点，该数组表示经验乘积分布

$$\widehat{\mathbb{P}} = \frac{\delta_0 + \sum_{r=1}^{n_\xi} \delta_{\xi_r}}{n_\xi + 1} \otimes \frac{1}{n_\mu} \sum_{j=1}^{n_\mu} \delta_{\mu_j}, \tag{16}$$

其中， δ_x 表示位于x处的单位点质量， $\xi = 0$ 表示标称位姿。标称行的质量与每个扰动行相同。若扰动闭合失败，其质量仍保留在所有摩擦列中，质量值为零；不会因此从分布中移除该场景。即使扰动评估会改变原生场景，存储的标称闭合仍是重建目标。调用方会应用保留的手掌位姿和闭合后的独立坐标，以显示或重建已选择的抓取。

令N_{feas} 表示通过所有标称筛选条件的命令数。每个枚举出的命令都会调用标称接近姿态，每个获准命令还会再调用n_ξ 个接近姿态，因此

$$\begin{aligned} N_{\text{approach}} &= N_{\text{cmd}} + N_{\text{feas}} n_\xi, \\ N_{\text{autograsp}} &\leq N_{\text{close}} \leq N_{\text{approach}}. \end{aligned} \tag{17}$$

在精确算术下，一次接近最多会执行 $1+\lceil 2s/h\rceil$ 次场景碰撞查询，包括初始放置；随后的一次闭合还会执行自己的放置查询。浮点行程累积可能会改变边界步长。摩擦节点会增加数值评分，但不会重复执行原生闭合。若$n_{\mathrm{ok}}(a)$表示获准命令a的成功扰动闭合次数，则`analyze_gws`的调用次数为

$$n_{\mu}\sum_{a\in\mathcal{A}}(1+n_{\mathrm{ok}}(a))\le N_{\mathrm{feas}}(n_{\xi}+1)n_{\mu}. \tag{18}$$

每次调用都会计算重新计算的质量 ε 和体积；任务模式还会计算任务容量凸包。因此，该表达式统计的是分析调用次数，与内部凸包计算的次数无关。

保留的 ε 数组包含 $N_{\mathrm{feas}}(n_{\xi}+1)n^{\mu}$ 个标量，任务模式还会保留数量相同的额外容量值。除名义几何形状和记录数据外，它们的存储空间为 $O(N_{\mathrm{feas}}(n_{\xi}+1)n_{\mu})$。排序需要$O(N_{\mathrm{feas}}\log N_{\mathrm{feas}})$次比较。每个获准命令完成评分后，`top_k` 参数会限制返回的记录数量。这些计数将扩大几何搜索库、增加姿态重新获取次数以及细化摩擦分析区分为彼此独立的计算选择。

2.6.3. 接触力旋量模型与容量

扳手分析将闭合形成的接触点与抓取所能承受的外载荷联系起来。我们将每个接触点 i 建模为物体坐标系 [53, 54]中的摩擦点接触，其位置为 p_i，向内单位法向量为 n_i。对于接触力 f 和摩擦系数 $\mu\ge 0$，库仑条件为 $\mathrm{f}_n=\mathrm{n}_i^{\top}\mathrm{f}\ge 0$ 且 $\|\mathrm{f}-\mathrm{f}_n\mathrm{n}_i\|_2\le\mu\,\mathrm{f}_n$。默认的锥近似采用八个等间隔的单位切向方向 s_{ij} 以及边缘力 $\mathrm{f}_{ij}=\mathrm{n}_i+\mu\mathrm{s}_{ij}$。这些力相对于质心 c 产生的力矩给出了归一化力旋量

$$w_{ij}=(f_{ij},\,(p_i-c)\times f_{ij}/\rho). \tag{19}$$

由于每条边的力都具有单位法向分量，因此非负锥边系数之和就是总法向力。力旋量凸包 $W=\mathrm{conv}(\{w_{ij}\})$ 包含单位总法向力 [55, 1]下可获得的力旋量。力闭合要求原点位于W的内部；在此条件下，该凸包表示总法向力上限为一。重新计算的Ferrari–Canny质量 ε 是W中以原点为中心的最大欧几里得球的半径。正长度 ρ 用于相对于力对力矩进行缩放。对于重新计算的质量 ε，该长度为 $\rho_{\varepsilon}=\max\{\max_i\|p_i-c\|_2$，即0.01毫米$\}$。对于任务力旋量容量$Q$，凸包和所需载荷使用相同的 $\rho_Q=100$ 毫米单位。因此，ρ 表示所选的特定度量的力矩尺度，而 ρ_{ε} 和 ρ_Q 标识所实现的两种约定。

对于指定的{style id='1'>K </style}个外部任务力中的每一个，我们将f_k 归一化为单位长度，并保留其在物体坐标系毫米中的作用点x_k。所需的平衡接触力旋量为 $\mathrm{l}_k=-(f_k,\;(x_k-c)\times f_k/\rho)$，均关于同一个质心。相应地，调用方将外部力方向取反后，提供给规划器的平衡力输入。容量是沿支撑能力最低的所需方向、单位总法向力所能支撑的外力大小 [56, 57]，

$$Q(a,\xi,\mu)=\min_{k}\sup\{\lambda\ge 0:\lambda\ell_k\in W\}. \tag{20}$$

未能闭合或实现的抓取未达到力闭合时，赋值为$Q=0$。

数值接口还公开了 FRoGGeR [7] 中使用的最小权重力闭合公式。对于$W_m=[w_1\;\cdots\;w_N]$，线性规划为

$$\ell^{*}=\max_{\alpha\in\mathbb{R}^N,\,\ell\in\mathbb{R}}\ \ell\quad\text{subject to}\quad W_m\alpha=0,\quad \mathbf{1}^{\top}\alpha=1,\quad \alpha_j\ge\ell\quad(j=1,\dots,N). \tag{21}$$

正的最优值意味着原点存在严格正的凸表示形式。综合而言，$\mathrm{l}^{*}>0$ 与 $\mathrm{rank}W_m=6$等价于 $0\in\mathrm{int}W$，即几何上的力闭合条件。秩条件排除了仅限于低维力旋量子空间的平衡。SciPy 实现会返回最优值和一个布尔值；只有最优值高于数值容差且力旋量矩阵的秩为六时，该布尔值才为真。将最优值与秩检验分开保留，可以区分低维力旋量子空间中的正裕度与六维力闭合。可选的无 SciPy 计算会返回最小奇异值代理指标。因此，最小权重系数、Ferrari–Canny 半径和任务力旋量容量各自保留不同的定义和数值输出。

对于任务力旋量容量，（20）中的凸包成员条件也可表述为系数分配公式。使用（W_m （21）中定义的力旋量矩阵，沿所需平衡力旋量 l_k 的容量为

$$q_k^{\mathrm{ray}} = \max_{\alpha \in \mathbb{R}^N_{\geq 0},\, \lambda \geq 0} \lambda \quad \text{subject to} \quad W_m \alpha = \lambda \ell_k, \qquad \mathbf{1}^\top \alpha = 1. \tag{22}$$

系数上的等式对所有接触点施加统一的总法向力预算。在严格的六维力闭合条件下，Q = 最小$_k q^{\mathrm{ray}}{}_k$。这种表述无需构造凸包面，即可通过线性规划计算出相同的方向容量。在评估方向规划之前，我们先使用秩合格最小权重证书来确立严格闭合。数值实现对力旋量矩阵和任务载荷应用相同的可逆行缩放，以平衡力旋量坐标方程。每次求解时都会对任务量级进行归一化，并在返回的乘子中还原。缩放后的等式与不等式残差均会根据 10^{-9} 的容差进行检查。对于 Ferrari–Canny 质量，支撑平面计算仍采用其欧几里得定义。确定性的凸包重试可在不扰动输入力旋量的情况下处理退化的面拓扑，并检查归一化平面是否有限、输入点是否位于凸包内，以及原点是否严格不在支撑平面上。

2.6.4. 力预算与力旋量度量比较

根据接触模型，载荷容量取决于允许的接触力预算。原始力旋量的归一化方式决定了力旋量凸包所表示的预算。数组构建器 `build_grasp_matrix` 通过 `unit_normal_force` 提供此选项，其默认值为 `True`。根据式（19）中的单位法向力约定，非负边系数 α_{ij} 会产生接触力 $f_i = \sum_j \alpha_{ij} f_{ij}$。对于与单位切向方向正交的单位法向量，

$$\sum_i n_i^\top f_i = \sum_{i,j} \alpha_{ij}. \tag{23}$$

因此，凸包系数约束会将所有接触点处的总法向力固定为 1。若 $0 \in W$，则将系数等式替换为上限可得 $\mathrm{conv}(\{0\} \cup W) = W$。严格力闭合满足此原点成员条件。该预算由各接触点共同分担，因此不会对单个接触点施加独立的法向力限制。

将 单位_法向_力=`False` 设为会将每个原始力和力矩除以 $\sqrt{1+\mu^2}$。用上标 n 和 e 分别表示单位法向约定和单位边约定，并定义 $k(\mu) = 1/\sqrt{1+\mu^2}$。在共同的标量摩擦系数下，

$$W_m^{\mathrm{e}} = k(\mu) W_m^{\mathrm{n}}, \qquad W^{\mathrm{e}} = k(\mu) W^{\mathrm{n}}. \tag{24}$$

在理想的正交归一接触坐标系中，每个替代原始力的欧几里得范数均为 1，法向分量为 $k(\mu)$。因此，非负系数之和为 1 时，总法向力固定为 $k(\mu)$，而 $\sum_i \|f_i\|_2 \leq 1$，这是由三角不等式得出的。由于同一接触点处多条边的切向分量可能相互抵消，合接触力范数不一定相等。因此，替代凸包表示的是一种原始边系数预算，用以限制合接触力范数之和。

常见的正比例伸缩会保持原点内点和矩阵秩不变，因此在固定 μ 时，两种约定具有相同的几何力闭合状态。(21)中的最小权重线性规划具有更强的坐标不变性：对于任意可逆的力旋量坐标矩阵 D，方程 $DW_m\alpha = 0$ 和 $W_m\alpha = 0$ 具有完全相同的解。因此，

$$\ell^*(DW_m) = \ell^*(W_m), \qquad \ell^*(W_m) \leq \frac{1}{N}, \tag{25}$$

其中，该界限对于可行程序成立，并可通过对N 列中的 $\alpha_j \geq 1$ 求和得到。最优值衡量系数裕度，并取决于生成元表示：重复列或冗余列可能在不改变凸包的情况下改变该值。该不变性适用于线性规划；奇异值代理会在均匀膨胀下发生缩放。绝对数值阈值也可能影响退化附近计算得到的闭包谓词。

几何评分对力预算变化的响应各不相同。对于正因子 k、固定的任务载荷以及严格的原点内点条件，

$$\varepsilon(kW) = k\varepsilon(W), \qquad Q(kW, \{\ell_k\}) = kQ(W, \{\ell_k\}), \qquad \mathrm{vol}_6(kW) = k^6 \,\mathrm{vol}_6(W). \tag{26}$$

这里，Q 的显式凸包和载荷论证表明了（20）中的成员判定问题。容量定律成立，当且仅当 $\lambda \mathrm{l}_k \in kW$ （λ/k）$\lambda \mathrm{l}_k \in W$。对凸包和所需载荷同时应用相同的可逆坐标变换，可以保持容量不变。下文讨论的共同力矩缩放就是这种变换。一般的各向异性变换会改变欧几里得球，因此未必能保持 Ferrari–Canny质量。仅改变原始量归一化，则会在载荷不变的情况下改变力预算。

采用单位法向力生成元时，对于固定接触点，增加摩擦会扩大可达力旋量凸包。采用替代约定时，容量变为 $k(\mu)Q^{\mathrm{n}}(\mu)$；递增的单位法向容量乘以递减的 $k(\mu)$，并不能保证整体具有单调性。对于单个共同摩擦系数，因子 $k(\mu)$会对所有固定几何形状的 Ferrari– Canny 质量指标和任务容量进行相同缩放，从而保留它们的排序。在不同摩擦节点之间，该因子会因列而异，因此均值和下尾排名未必得以保留。线性规划最小权重最优值在每个节点上仍分别保持不变。规划器使用默认的单位法向约定；可通过调用方定义的数组分析来比较替代约定。

对于固定接触点，增加摩擦系数会扩大每个可行单位法向力多边形。等间距切向方向s^{ij} 的凸包包含原点，因此用更大的非负因子缩放该多边形，会保留较小多边形中的每个点。力到力旋量的线性映射以及对接触点取凸包均会保留这种包含关系。因此，$\mu_1 \le \mu_2$ 意味着相应力旋量凸包之间存在包含关系，并且对于固定闭合，有 $Q(a, \xi, \mu_1) \subseteq Q(a, \xi, \mu_2)$。共同的可逆力矩缩放 $D = \mathrm{diag}(I_3, \gamma I_3)$，$\gamma > 0$，满足 $\lambda \mathrm{l}_k \in W$ 当且仅当 $\lambda D \mathrm{l}_k \in DW$。因此，任务力旋量容量不受共同力矩尺度影响，而 Ferrari–Canny 球的大小则通过欧几里得范数受到该尺度影响。

2.6.5. 质量的下尾聚合

令 $Z_a = Q(a, \xi, \mu)$ 表示所选位姿和摩擦分布上的质量；重新计算得到的 ε 可代替 Q 使用。规划器通过最大化质量为 $\beta \in (0, 1]$ [58], 的下尾条件风险价值来选择可行指令

$$a^{\star} \in \arg\max_{a \in \mathcal{A}} \sup_{\eta \in \mathbb{R}} \left\{ \eta - \frac{1}{\beta} \mathbb{E}[(\eta - Z_a)_+] \right\}, \tag{27}$$

其中，$\mathcal{A}$ 包含通过名义可行性筛选条件的指令，且 $(x)_+ = \max(x, 0)$。目标函数是在 $\beta = 1$ 处的均值，而名义评分则在参考摩擦 μ_0 下计算 $Q(a, 0, \mu_0)$。对于加权样本 $z_1, \ldots, z_M$及其归一化概率 $\omega_i \ge 0$，令 $z_{(1)} \le \cdots \le z_{(M)}$，并令 $\omega_{(i)}$ 表示相应的权重。给定 $\Omega_i = \sum_{j=1}^{i} \omega_{(j)}$ 和 $\Omega_0 = 0$，经验目标函数为

$$\widehat{\mathrm{CVaR}}_{\beta} = \frac{1}{\beta} \sum_{i=1}^{M} z_{(i)} \min\{\omega_{(i)}, (\beta - \Omega_{i-1})_+\}. \tag{28}$$

将样本按质量从低到高排序，并依次纳入，直到累积概率达到 β。如果最后一个纳入的样本会使质量超过该值，则仅将其所需比例的权重计入平均值。权重相等时，取质量最低的 βM 个样本，并按比例计入最后一个样本。当（16）中定义的零质量情景填满下尾时，每个受影响的指令都会得到相同的零分数，其基于尾部的排名也因此并列。高成本和低抓取质量都是不利结果。对质量取负值，可将其下尾映射为成本的上尾，因此数值接口将质量的下尾条件风险价值计算为 –条件风险价值$_{\beta}^{\mathrm{cost}}(-z)$.

在不利的尾部摩擦区间内对质量取平均值，与在平均尾部摩擦处评估质量不同。对于固定的接触点和非递减的质量$g(\mu)$,

$$\mathrm{CVaR}_{\beta}[g(\mu)] = \frac{1}{\beta} \int_0^{\beta} g\left(F_{\mu}^{-1}(u)\right) du, \tag{29}$$

其中 F_{μ}^{-1}是摩擦分位数函数。在尾部平均摩擦下评估质量可得 g(条件风险价值$_{\beta}[\mu]$)；对于尾部区间上的仿射 29，等式（ffine g）成立。采样目标函数在平均之前评估质量，同时还纳入位姿扰动产生的接触变化。无需重复执行闭合抓取，即可对已保存数组应用其他目标函数。

2.7. 导出、闭合与抓取重建

可执行工作流从编译后的 MuJoCo 模型导出 RealHand L6 和一个 50 毫米立方体，在 GraspIt! 中搜索抓取指令，并在 MuJoCo 中重建已选择的指令。清单 3 用于构建包装器镜像并运行公开示例。随附的 Dockerfile 使用 Ubuntu 24.04，以及 Qt 5、Coin3D/SoQt、Qhull、pybind11、NumPy、SciPy 和 MuJoCo。其默认预处理会将上游 GraspIt! 源代码快照固定在提交`de64377`，并应用所提供的兼容性补丁。引擎预处理脚本会记录完整的提交标识符。第4.5 节中的接口测量结果会单独标明其所评估的引擎版本。变量`WRAPPER`和`HAND_ASSETS`分别标识包装器检出目录和外部 RealHand L6 网格目录。我们使用 40 个接近方向、六个滚转角和六个预成形姿态配置了搜索。重新计算的质量 ε针对标称位姿以及四个扰动接近方向进行评估，摩擦节点为 0.4、0.7 和 1.0，从而为每条指令生成十五个等权重的位姿–摩擦值。输出包含机器人和物体 XML、连杆网格、关节映射，以及最多五条按秩排列的抓取记录。评估实例返回了五条记录；其秩最高的抓取包含八个接触点，经验下尾部条件风险价值 条件风险价值$_{0.5}(\varepsilon) = 0.00896$（对应重新计算的质量 ε）。

清单 3：规划 RealHand L6 抓取并重建其 MuJoCo 配置

```
docker build -f "$WRAPPER/docker/Dockerfile" -t graspit-py "$WRAPPER"
bash "$WRAPPER/examples/l6_workflow/run.sh" graspit-py \
  "$HAND_ASSETS" ./l6-workflow-output
```

清单 4展示了公共工作流如何读取抓取记录并恢复源铰链坐标。初始化后的规划器使用导出的手和物体，以及提供的搜索配置。清单将独立坐标映射到具名源铰链，而记录则保留物体坐标系中的接触位置和内向法线，以供数值分析。

清单 4：读取抓取记录并恢复源铰链坐标

```
导入mujoco作为mj
导入NumPy作为 np

已选择数据 = 规划器.plan(top_k=5)[0].to_dict()
T_OH = np.asarray(已选择数据["T_物体_手_毫米"])
接触点_O = 已选择数据["接触点"]
对于 运动链 在 清单["运动链"]:
    对于 关节在运动链:
        u = 已选择数据["自由度_弧度"][关节["自由度"]]
        data.joint(joint["MuJoCo_关节"]).qpos[0] = (
            关节["比率"] *u +关节["偏移_弧度"])
mj.mj_正向运动学(model, data)
```

每个源铰链角度都通过独立坐标，使用（3）进行重建，并根据 MuJoCo 关节名称赋值。机器人配置固定手掌后，工作流会放置物体，使相对于物体的手掌位姿等于规划变换，并将平移从毫米转换为米。随后，正向运动学会恢复所请求的相对位姿。平移和旋转残差达到浮点精度。这是对命令的运动学重建，外部模拟器之后可将其用于动态执行。

3. 用法与研究工作流

接触和坐标表示支持三项研究任务：评估外部抓取提案、为工具使用或装配约束接触点，以及分析已保存评分。这些工作流通过可执行的接口操作，将数学定义与调用方提供的模型、指令和不确定性分布连接起来。

3.1. 评估外部提供的抓取提议

外部抓取生成器提供手掌位姿、预成形和所需的闭合姿态，而包装器提供用于闭合和接触提取的公共原生模型。因此，学习型提案生成器或接触优化器可以使用原生接口，而无需采用所提供规划器的命令库。调用方首先将提议的手掌位置转换为世界坐标系约定，并检查已分配的预成形是否存在场景碰撞。随后，碰撞感知运动朝调用方的闭合坐标前进。清单5保留坐标轨迹，并在获取公共坐标系接触点之前刷新选定物体的抓取。所有位姿、坐标索引和闭合目标均来自调用方加载的手和场景。

清单 5：从调用方提供的提案中获取接触点

```
来自 graspit_py.core.bindings导入 Grasp

world.set_current_手_位姿(
    手掌_四元数_wxyz，手掌_平移_毫米)
world.set_current_手_dof_values(preshape_弧度)
如果不是世界.no_碰撞():
    引发ValueError("预成形与场景几何体相交")
已移动，追踪 = 世界.trace_将当前_手_的自由度_移动_至_接触点(
    闭合_目标_弧度、        所需_步长_大小=闭合_步数_弧度、
    在_接触_闭合=时停止_在_接触时停止)
抓取 = Grasp(world,graspable_body_食指=物体_index)
如果不 grasp.compute():
    引发 ValueError("闭合没有有效的原生抓取")
接触_O = grasp.object_接触_几何形状()
c_O_毫米 = world.get_graspable_body_cog_local(物体_索引)
```

返回接触位置和法向量，可使数值目标函数独立于提议生成器而变化。列表 6 在调用方选择的摩擦系数下构建 W_m ，使用物体坐标系接触和质心，并采用明确的统一力矩尺度 ρ。返回的最小权重最优解和力封闭谓词各自保留不同含义。此扫描会固定实际接触几何；摩擦采样辅助函数同样分析固定接触，而规划器的位姿样本则会重新执行接近和闭合。调用方提供的查询点和物体三角形也可在相同坐标系和长度单位下进行无符号几何距离分析。新的接触模型、载荷方向或摩擦网格需要重新计算数值评分；仅更改已保存评分的聚合方式则无需再次闭合。

列表 6：分析固定接触点和物体几何

```
import numpy as np from graspit_py.grasp.qualityimport ( build_gras
```

3.2. 固定接触的摩擦采样

函数 compute_gws_under_friction_uncertainty 用于评估已获取接触集的摩擦敏感性。接触位置、内向法线和物体中心保持不变，而每次摩擦抽样都会生成新的扳手分析结果。辅助函数从 X $\sim \mathcal{N}(\mathrm{m}_{\mu}, \sigma_{\mu}^{2})$ 中抽样，并使用 μ =clip(X, μ−,μ+)，其中 μ− = 0.1 和 μ+ = 2.0。其 friction_mean 和 friction_std 参数分别指定 m_{μ} 和 σ_{μ}。每次抽样都会调用 analyze_gws，使用默认的单位法向力生成元，以及根据第 2.6.3 节中的最大接触距离定义的力矩尺度 ρ_{ε}。该计算不需要原生接近或闭合，因此抽样变化以提供的接触几何为条件。

截断会产生端点概率质量以及连续内部密度。对于 $\sigma_{\mu} > 0$，其中 Φ表示标准正态累积分布函数，

$$\mathbb{P}(\mu = \mu_{-}) = \Phi\left(\frac{\mu_{-} - m_{\mu}}{\sigma_{\mu}}\right), \qquad \mathbb{P}(\mu = \mu_{+}) = 1 - \Phi\left(\frac{\mu_{+} - m_{\mu}}{\sigma_{\mu}}\right). \tag{30}$$

在区间内部，密度等于原始高斯密度。因此，给定的高斯参数通常不同于截断分布的均值和标准差。在 $\sigma_{\mu} = 0$ 处，所有质量都集中在 clip(m_{μ}, μ−, μ+)。

返回的 epsilon_样本 和 体积_样本 数组中，每次抽样各有一个值。字典还提供它们的均值和总体标准差、重新计算的质量的最小值和最大值、样本数量，以及fc_概率。当 SciPy 成功构建六维凸包时，体积值为$10^{6}\mathrm{vol}_{6}(W)$，即经过缩放的六维归一化扳手凸包体积。对于已处理的凸包构建失败情形，该辅助函数返回零。它们的标准差描述抽样变化，而非标准误。返回的fc_概率 字段表示在已安装的数值后端下，被 analyze_gws 判定为力闭合的摩擦抽样比例。它描述的是固定接触点的、依赖摩擦的数值分类，而非抓取获取成功的频率。特别是，该辅助函数的判定条件独立于满足秩条件的最小权重检验。

列表 7 提供一个带种子的 NumPy 生成器，并根据返回的质量数组计算下尾统计量。该辅助函数会使用提供的生成器状态，并返回得分样本，而不包含采样得到的摩擦值。无需重复分析，即可根据保存的质量评估其他尾部概率质量。若研究需要指定摩擦值下的评分，可使用显式网格，如列表 6所示。规划器数组还包含由位姿扰动产生的 contactchanges，并保留公式（16）中的标称位姿。

列表 7：对已获取的接触集进行摩擦采样

```
导入numpy作为 np
从graspit_py.core.types 导入 ContactPoint
从 graspit_py.grasp.quality导入 (
    计算_gws_在_摩擦_不确定性)
来自graspit_py.risk导入下限_尾部_均值

接触点 = [ContactPoint(np.asarray(c["位置_毫米"]),
                       np.asarray(c["法向"]))
          对于 c在 接触点_O]
摘要 = 计算_gws_在_摩擦_不确定性(
    接触点 , c_O_mm,
    摩擦 –均值=摩擦 _mean,
    摩擦 –标准差=摩擦 _std,
    n_样本=n_摩擦_样本 ,
随机数生成器=np.random.default_rng(种子))
epsilon_样本 =摘要[“epsilon_样本”]
尾部_质量 = 下限_尾部_均值(epsilon_样本, alpha=beta)
```

3.3. 配置工具与夹具约束

工具使用或装配研究会指定物体上的允许接触区域，同时确保其工作端或周围夹具仍可接近。调用方以物体坐标系毫米表示此区域，并提供预

成形形状和适合该手的闭合子集。将选定的独立坐标保持在其预形状值，可使其余坐标围绕允许区域闭合。接触区域测试会约束已获得的接触点；名义支撑和禁入谓词则约束最终的手部顶点。禁入框会排除严格位于框内的点；恰好位于边界上的手部顶点则会被接受。由于框测试使用顶点，因此面相交和边相交需要调用方额外进行几何检查。在指令枚举之前，间隔变换构造器会向外部机械臂可达性例程提供相同的接近位姿。

任务行将这些几何选择与所需载荷联系起来。每一行都指定一个物体坐标系中的施力点和平衡力方向，因此调用方会将该方向取反，以抵消所要抵抗的外力。规划器会分别对每个方向进行归一化；输入的力的大小不会规定不等的载荷大小。由此得到的任务力旋量容量按照第 2.6 节所述约定，对沿指定方向的抵抗能力进行排序。位姿扰动会重新执行接近和闭合操作，使评分数组能够反映摩擦节点上的接触变化和闭合失败。接触数量和接触区域测试适用于这些扰动闭合；支撑和夹具谓词则适用于名义最终顶点。外部机械臂规划为由此产生的指令目标提供可行的机器人运动。

3.4. 检查已保存评分和源对应关系

已保存的位姿-摩擦数组可用于分析名义质量与建模扰动下表现不同的指令。研究人员会检查较弱的位姿行或摩擦列，然后比较均值与选定尾部质量下的下尾均值。清单 8直接使用保留的 Ferrari–Canny 质量样本，并接受与已保存数组对齐的情景权重。独立的风险函数会对提供的非负权重进行归一化，并仅纳入边界样本所需的概率质量。已存储的搜索配置标明了姿态分布和摩擦节点；重新加权的结果描述的是在所提供权重下相同的采样结果。

清单 8：重新聚合已保存的位姿–摩擦评分

```
导入numpy作为 np从 graspit_py.risk导入


样本 = np.asarray(record["epsilon_samples"]) 均值_质量 = 样
```

同一记录还支持通过另一种表示形式重建源模型：不同的表示形式包括已接受的手掌位置和闭合后的独立坐标。导出器清单通过（3）将每个独立坐标展开为具名源铰链。将物体世界变换与记录的物体相对手掌变换合成，并将毫米转换为米后，即可为外部机械臂规划器提供手掌目标。第 2.7 节提供了可执行的重建示例。进行模型诊断时，清单中的关节和连杆标识可将源铰链与原生连杆位姿及纯导出正向运动学关联起来。比较经过（2）中合成后的这些位姿，可以定位坐标或坐标系转换差异，而不会将其与已保存的质量聚合混为一谈。

4. 评估与讨论

4.1. 原生 *GraspIt!* 中的几何

我们评估了不同手部构型下导出的 GraspIt! 连杆几何与 MuJoCo 源几何之间的一致性。我们为原生 GraspIt! 中采样的每个独立坐标向量赋值，并读取应用后的数值。随后，我们直接根据编译版 MuJoCo 的等式约束计算驱动角。我们在 RealHand L6、LEAP、Ability、Shadow 和 Inspire 模型的全部 75 种构型下评估了两种导出模式。源模式误差根据源网格顶点计算，凸包模式误差则根据每个编译后的碰撞几何凸包的顶点计算

表5：导出的连杆几何形状（按网格表示形式）。误差为对应手掌坐标系 顶点集。A - 有列出的差异均在100 µm容差范围内。

Hand	坐标	连杆	构型	源 (µm)	凸包 (µm)
RealHand L6	6	12	49	< 0.01	< 0.01
LEAP	16	17	4	< 0.01	< 0.01
Ability	6	11	12	< 0.01	< 0.01
Shadow，手掌	18	23	5	< 0.01	< 0.01
Inspire，右	6	13	5	0.836	1.29
双铰链模型	1	3	3	< 0.01	—

。对应的顶点集均以各自的掌部坐标系表示。对于有限顶点集A 和B，对称豪斯多夫距离为

$$d_H(A,B) = \max\left\{\max_{a\in A}\min_{b\in B}\|a-b\|_2, \max_{b\in B}\min_{a\in A}\|b-a\|_2\right\}. \tag{31}$$

表 5 报告了所有连杆和构型中的最大偏差。在五种手和两种模式中，最大偏差约为 1.3 µm，与 100 µm 的几何容差相比，请求的独立坐标与实际应用的独立坐标一致。L6、Ability 和 Inspire 模型采用耦合铰链；LEAP 采用独立铰链。Shadow 手掌与手指规划模型使用四组已声明的等角度远端耦合，其腕部位于导出子树之外。

我们在合成模型上评估了参考修正和几何近似。一个双铰链模型的主动与从动参考角分别为 0.35 和 -0.20 弧度，系数为 $c_0 = 0.16$ 弧度和 $c_1 = 0.70$。在三个主动角下，导出的 -0.285 弧度偏移满足源等式，误差在 10^{-12} 弧度以内。一个具有近乎平行的坐标轴、仿射耦合和刚性远端体的三指模型，在三个构型和七个连杆上的最大顶点误差为 14.4 µm，也在几何容差范围内。凸包模式比较使用编译凸包连通性，而源模式比较使用源三角形。我们使用独立的 MuJoCo 碰撞距离探针，在三个耦合关节构型下检查了凹网格凸包的外表面。我们还验证了连通性，以确认导出面没有连接来自不同碰撞组件的顶点。

图4 展示了五个模型的几何评估中的三个构型。关节值按源名称指定，并在 GraspIt! 中查询了导出的连杆位姿和网格。在每个源/导出配对中，共同视图展示了 L6 和 Ability 手的耦合食指链，以及独立的 LEAP坐标。手掌坐标系轴和手指标签标识了相应的组成部分；报告的每个模型误差涵盖了所评估的构型。

4.2. 参考角与碰撞组件

几何对应关系取决于仿射关节映射以及附加到各连杆上的碰撞组件。我们通过在式（4）中省略参考角贡献，或将某个连杆的凸组件替换为一个基于其合并顶点构建的凸包，来单独考察它们的影响。比较采用了表5中的五个手部模型、一个具有非零参考角的耦合关节模型，以及一个带有指定间隙的 U 形组件并集。这两个受控模型分别用于检验特定的转换机制。对于每个模型，我们在其可行独立坐标区间的 0.25、0.50 和 0.75 倍处评估了三种构型。使用完整导出器和去除参考角的导出器评估源表面与编译凸包，并针对编译凸包评估组件合并。最终共进行了 105 次构型比较，其中包括 42 种完整导出构型。各变体中的独立坐标、关节限位、源几何和编译器设置均保持不变。

每个完整导出构型均满足 0.1 毫米的对应容差，无需调整关节限位处的带符号坐标。在非零参考角模型中，仿射修正为 -0.445 弧度。省略该项会使其连杆顶点在规定构型下发生最大 13.4 毫米的位移。在五个手部场景中，参考角贡献为零，因此省略该项不会改变其几何形状。因此，受控比较证实了参考项对于具有非零参考角的模型所起的作用，而手部测量则证实了所评估源模型之间的对应关系。



(a) RealHand L6

$n_u = 6, \quad n_q = 11$

$N = 49, \quad d_H^{\max} < 0.01\ \mu m$

$q_{I,2} = 1.125676 q_{I,1}$

(b) LEAP Hand

$n_u = 16, \quad n_q = 16$

$N = 4, \quad d_H^{\max} < 0.01\ \mu m$

$q^j = u_j, \quad j = 1, \ldots, 16$

(c) Ability Hand

$n_u = 6, \quad n_q = 10$

$N = 12, \quad d_H^{\max} < 0.01\ \mu m$

$q_{I,2} = 1.05851325 q_{I,1}$

图4：**三种手的模型导出**。上排为 MuJoCo 源表面，下排为 GraspIt! 原生导出；这些构型取自几何评估。关节值根据源名称赋值。掌部坐标系 $\{H\}$ 和手指标签标识了对应的组成部分。上排字母表示拇指（T）、食指（I）、中指（M）、无名指（R）和小指（L）。此处，n_u 表示独立坐标的数量，n_q 表示铰链的数量，而 $d_H^{\max}$ 表示每只手在 N 个评估构型上的掌部坐标系顶点豪斯多夫误差最大值。耦合表达式使用食指近端和远端坐标 $q_{I,1}$ 和 $q_{I,2}$。

即使原始组成部分的顶点仍位于合并凸包内部，组件合并也会改变碰撞集合。我们使用 21^3 每个连杆的分量并集包围盒中均匀间隔的单元中心，测量新增占据区域。我们分别根据原始并集和合并凸包对每个点进行分类。将仅被合并凸包占据的点数乘以单元体积，得到表6中的新增体积估计。合并在 Shadow 和 Inspire 模型中引入了占据区域。L6、LEAP 和 Ability 场景中的每个导出连杆均只有一个编译组件，因此其碰撞集合未发生变化。完整导出器在所有情况下都保留了各个独立组件。

对于 U形模型，新增体积的精确值为 64 cm^3, ，固定网格估算值为 60.5 cm^3。预先指定的球体占据了间隙，且未与原始组件并集发生相互穿透。合并各组成部分后，间隙被填满，产生了 11 个物体接触点，并在相同闭合命令下使最终关节坐标改变了 0.25 弧度。

表6：合并碰撞组件的影响。每个连杆在其分量并集包围盒中使用 21^3 个单元中心探针。百分比表示任一连杆上新增探针的最大比例。新增体积为各连杆中点网格估算值的总和。斜体条目表示合并后新增的占用体积。

模型	新增探针的连杆数 / 复合连杆	最大添加量（%）	新增体积 (cm³)
Shadow	1/5	25.8%	*68.2*
Inspire	1/1	13.7%	*87.3*
U形模型	1/1	56.0%	*60.5*

4.3. 原生手部模型与可复用配置文件

原始 Barrett模型通过原生 GraspIt! 机器人接口得到支持 [1]。我们加载了 Barrett模型，该模型有四个独立坐标和九个连杆，并对掌部坐标系中以 (0, 0, 100) 毫米为中心的 50 毫米立方体执行自动抓取闭合。最终抓取有六个接触点，原生 Ferrari–Canny质量 $\varepsilon_{\mathrm{GI}} = 0$。146。此路径使用原生模型的关节传动，并通过 Python 提供闭合、接触几何和力旋量指标。

针对表 5 中评估的 Shadow 和 Inspire 模型，提供了预处理和导出配置文件。配置文件会标明掌部根节点、独立坐标、预成形，以及精细关节关系。预处理过程中会解析 URDF 碰撞网格，并将模仿关系转换为 MuJoCo 等式约束，同时应用式（4）中的参考修正。Inspire 配置文件保留了六个模仿关系及其角度偏移；Shadow 的四个等角度远端规划耦合则被显式声明。将每个从动铰链的限位与其独立坐标区间取交集，即可得到可行区间，用于评估每种模式下的五个内部构型；每个构型均不经坐标调整直接应用。手模型指南提供源模型链接、配置文件、指令、坐标约定和资源许可证。

将导出的 Shadow 和 Inspire 模型闭合在直径为 60 毫米的球体上，分别产生了十一个和七个接触点，分布于全部五根手指上。它们原生的质量指标分别为 $\varepsilon_{\mathrm{GI}} = 0$。423 和 0.539。Barrett、Shadow 或 Inspire 的闭合抓取均未涉及掌部接触。在 Shadow 和 Inspire 闭合构型下，编译模型与导出模型之间的顶点最大差异小于 0.01 µm。图 5 展示了所得抓取构型、物体侧接触位置，以及手部和物体坐标系。

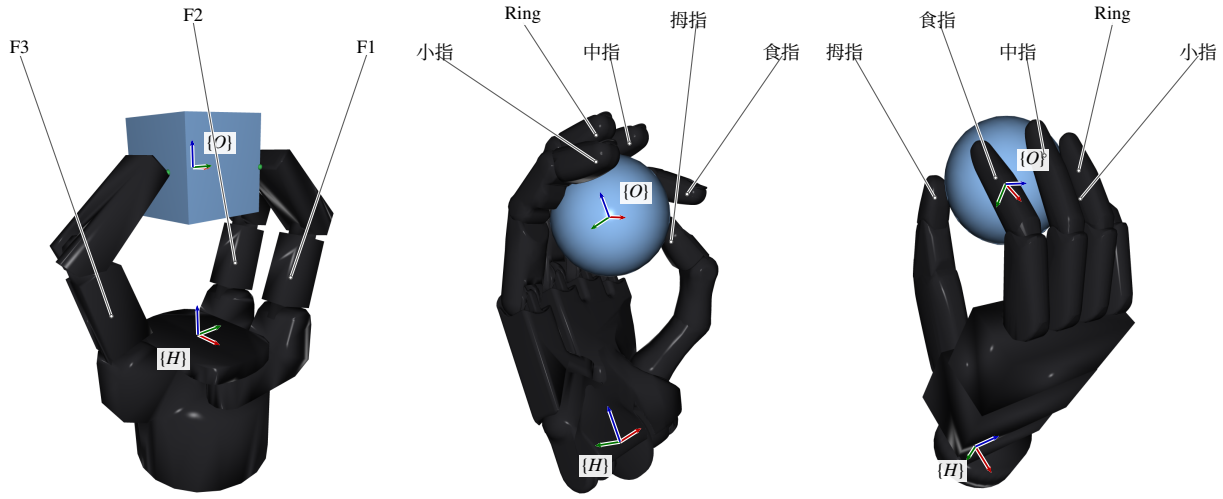
4.4. 独立力旋量力学验证

式（22）中的系数分配程序依赖于锥边表示和严格的力闭合谓词。我们使用一种独立的笛卡尔力公式，以各摩擦多边形的面为基础，评估了两者在数值上的一致性。设 $\mathbf{t}_{i1}$、$\mathbf{t}_{i2}$ 为正交归一的接触切向量，s 为多边形边数。定义 $\mathbf{v}_{ij} = \cos(2\pi(\mathrm{j} + 1/2)/\mathrm{s})\mathbf{t}_{i1} + \sin(2\pi(\mathrm{j} + 1/2)/\mathrm{s})\mathbf{t}_{i2}$。独立参考方法对笛卡尔接触力 $\mathbf{f}_i \in \mathbb{R}^3$上的载荷乘数进行最大化：

$$
\begin{aligned}
q_k^{\mathrm{cart}} = \max_{\lambda \ge 0,\, \{f_i\}} \lambda, \\
\sum_i \begin{bmatrix} f_i \\ (p_i - c) \times f_i / \rho \end{bmatrix} = \lambda \ell_k, \qquad \sum_i n_i^{\top} f_i \le 1, \\
n_i^{\top} f_i \ge 0, \qquad v_{ij}^{\top} f_i \le \mu \cos(\pi/s) n_i^{\top} f_i \quad \text{for every } i, j.
\end{aligned}
\tag{32}
$$

这些面不等式描述的内接摩擦多边形与式（19）中的锥边所描述的相同。参考计算独立于生产用力旋量构造器，构建了其平衡矩阵和多边形约束。另行进行的仿射秩和严格内部计算，在比较 $\min_k q^{\mathrm{cart}}{}_k$ 与 Q之前，应用了无力闭合时取零的约定。作此区分是必要的，因为某个接触集可能能够支撑特定载荷方向，却不具备六维力闭合。

我们使用了六种图示手部模型的接触集、两种解析接触布置，以及十二种带种子的八接触点布置。四种摩擦系数、十二个单位平衡力方向和两个作用点，共进行了 1,920 次方向容量比较。摩擦多边形有八条边，总法向力预算为一。对称解析布置中的十八个中心施加的轴向载荷具有已知的单位容量。坐标比较将共同的力矩长度从 1 毫米变化到 1,000 毫米，并统一转换所有长度单位



(a) 巴雷特灵巧手

6 个接触点

$\varepsilon_{GI} = 0.146$

(b) Shadow Hand

11 个接触点

$\varepsilon_{GI} = 0.423$

(c) Inspire Hand

7 个接触点

$\varepsilon_{GI} = 0.539$

图 5：**原生与导出手部模型上的抓取**。(a) 原生 Barrett 在 50 毫米立方体上的闭合抓取。(b, c) 导出的 Shadow 和 Inspire 在直径为 60 毫米的球体上的闭合抓取。绿色标记表示通过原生的基于碰撞的闭合得到的物体侧接触。Shadow 显示为处于闭合关节构型时的 URDF 可视网格。其接触位置和质量根据碰撞几何体计算。掌部坐标系 $\{H\}$、物体坐标系 $\{O\}$以及手指标签均根据查询构型投影得出。红、绿、蓝坐标轴分别表示相应坐标系的x、y和z 方向。F1–F3 标识 Barrett 的三条运动学手指链。每个面板中的读数给出接触数量和原生 Ferrari–Canny 质量 ε_{GI}。

，旋转物理接触多边形及其坐标系，并同时平移接触力旋量和任务力旋量的力矩原点。在坐标系比较中，旋转现有多边形保留了其物理边缘方向。边界、凸包外以及秩亏力旋量集合提供了其他严格闭合情形。

系数规划在全部 1,920 项方向比较中均与独立的笛卡尔形式一致，最大绝对容量差异为 2。73×10^{-10}（表7）。全部 18 项解析单位容量比较也满足相同的容差。在 960 项坐标比较中，最大变化为 1。30×10^{-13}。容量一致性采用绝对容差 $2{\times}10^{-7}$ 加相对容差 $2{\times}10^{-6}$进行评估。在单位法向力预算下，笛卡尔参考的最大平衡残差为 9。54×10^{-14} 符合秩条件的最小权重和 Ferrari–Canny 判据正确区分了全部 88 个内部、边界、外部及低维情形。

在 448 个构造的力旋量多胞体上考察了坐标均衡化，这些多胞体具有已知的原点几何，分别属于内部、边界、外部或低维情形。坐标轴尺度和任务量级的范围为 10^{-12} 至 10^{12},，其中包括重复的力旋量列。全部 448 项最小权重闭合分类和 9,408 项方向容量比较均符合其解析判据。最大缩放容量差异为 2.22×10^{-16}，均衡化线性规划中的最大平衡残差为 4.44×10^{-16}。这些比较独立评估了数值坐标缩放和载荷缩放，与主要容量研究中使用的接触几何无关。

4.5. Python 和 *ROS* 接口延迟

我们将直接 Python 绑定与 GraspIt! ROS 1 服务接口 [3], 在默认 TCPROS 连接和持久化服务客户端下分别作为不同条件进行比较。在接口基准测试中，我们使用了 GraspIt! 引擎修订版 `5f254577199fe0cab8c2ea094d32a66c118f2ba2`。三个接口均使用 50 毫米立方体和随附的 L6 模型进行评估，该模型具有十一个独立坐标。掌部坐标系中的手掌和立方体旋转均为单位旋转，立方体

表 7：数值扳手验证。粗体数字表示符合比较标准的数量。最大差异为评分的绝对差值，但单调性行（最大降幅）和带星号的尺度行除外。后者衡量正解析容量下容量乘以载荷幅值因子与 1 的误差，以及零容量下容量的绝对值。闭合情况分别使用最小权重和 Ferrari–Canny 判据进行了检查。

数量	情况数	最大差异
任务力旋量容量	**1,920**	2.73×10^{-10}
解析单位容量	**18**	2.00×10^{-10}
坐标变换	**960**	1.30×10^{-13}
摩擦单调性	**1,440**	1.00×10^{-10}
严格闭包分类	**88**	—
缩放最小权重分类	**448**	—
缩放方向容量	**9,408**	$2.22 \times 10^{-16*}$

表 8：Python 和 ROS 接口时延。时延以中位数 [第一四分位数、Q3], 表示，单位为每次操作。完整周期的第95百分位数另行报告。吞吐量为已完成周期数除以测得的总周期时间。粗体表示各项时延统计量中的最低值和吞吐量中的最高值。斜体表示最高时延和最低吞吐量。

操作	直接 Python	ROS 1	持久化 ROS 1
启动（秒）	**1.1226 [1.1198, 1.1256]**	1.2565 [1.2552，1.2593]	*1.2579 [1.2569, 1.2583]*
物体位姿（微秒）	**2.168 [2.140, 2.206]**	*661.7 [612.2，712.0]*	12.68 [12.28, 14.35]
自动抓取（毫秒）	**74.65 [74.47，74.89]**	116.54 [114.73，118.06]	*117.24 [116.08, 118.46]*
力旋量评分（毫秒）	*75.01 [74.88，75.16]*	**43.53 [43.31，43.90]**	43.77 [43.47, 44.04]
完整周期（毫秒）	**149.85 [149.51、150.24]**	*167.87 [165.09，169.92]*	167.79 [166.26，170.27]
周期，第95百分位数（毫秒）	**156.70**	177.43	*177.83*
吞吐量（周期/秒）	**6.64**	5.95	*5.92*

中心为 (40.1, 10.3, 93.1) 毫米，在掌部坐标系 $\{H\}$中，并且每次闭合都从零独立坐标开始。在所有接口中，渲染、值为一的手指闭合速度因子，以及首次接触后继续闭合的选项均保持不变。

我们在 Ubuntu 20.04 系统上使用 Intel Core i7-13700F，搭配 ROS Noetic 1.17.4 和 Python 3.12.10，对每种条件下的六次进程启动进行了计时。每次启动都使用了一个配备 6 吉比字节内存的四 CPU 容器，以及一个软件渲染的虚拟显示器。计时开始前，ROS 主节点和显示器已在运行；启动计时涵盖了从进程启动到世界加载及初始状态获取的全过程。每次启动先进行五个预热周期，随后计时执行 20 个重置—关闭—评分—状态周期，并读取物体位姿 100 次，因此每种条件共计 120 个周期和 600 次位姿查询。原生半径和体积通过两次绑定调用或一次 ROS 质量请求获取。我们依次运行各项条件，并在不同启动之间轮换其顺序。这些计时测量的是完整的操作路径，包括原生计算、访问器和事件处理。

所有 360 次测得的闭合抓取均返回 13 个接触点和相同的闭合坐标。原生 Ferrari–Canny 质量为 $\varepsilon_{\mathrm{GI}} = 0.0317$，扳手空间体积为 0.0902；输出的绝对差异均小于 10^{-9}。原生接口和 ROS 的接触查询均返回了位于相对接触面上的点。将这些点变换到立方体坐标系后进行一对一最小距离匹配，最大配对间隙为 0.188 毫米。

图6 区分了位姿读取和完整周期工作负载。直接读取位姿的中位数为 2.17 µs，而持久运行的 ROS 为 12.7µs。对于完整周期，中位数从 167.79 毫秒降至 149.85 毫秒，差异为 17.9 毫秒，相较于持久运行的 ROS 降低了 10.7%。闭合和评分占用了大部分经过时间。表 8 中的操作专属计时也说明，不能将完整周期的差异解读为单独的传输测量：质量调用所用的访问器和刷新路径有所不同。

4.6. 运行时间和内存扩展性

重复闭合和数值评分的成本决定了姿态–摩擦研究在实际中可达到的规模。我们使用 RealHand L6、Barrett 和 LEAP 模型，以及三种物体几何形状进行了工作负载矩阵测量：边长为 50 毫米的立方体、直径为 50 毫米的球体，以及直径为 44 毫米、高为 50 毫米的圆柱体。每个手–物体场景开始时均无

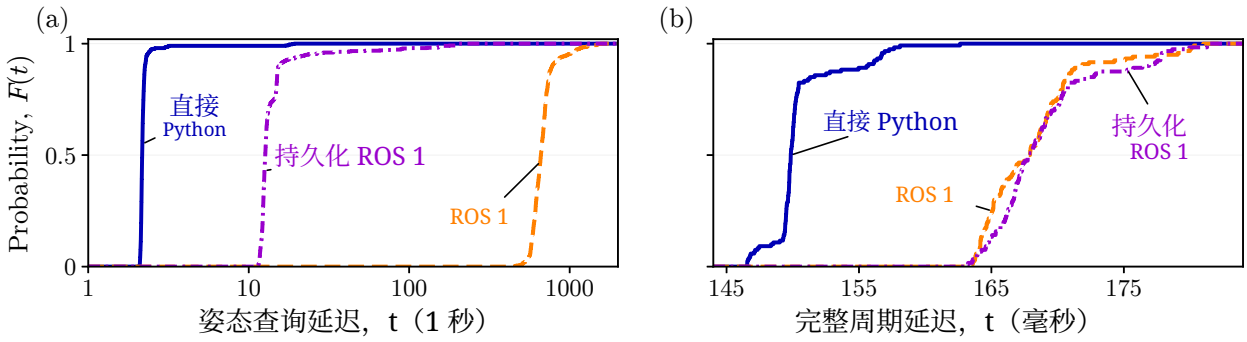


图 6：**按接口和工作负载划分的延迟**。直接 Python、默认连接 ROS 1 和持久化 ROS 1 的经验累积分布。(a) 每种条件下进行 600 次预热读取的物体位姿查询延迟，时间轴采用对数刻度。(b) 每种条件下进行 120 个周期的完整周期延迟。每个周期都会重置手、闭合手、计算力旋量评分并读取最终状态。

相互穿透。直接 Python、默认 ROS 1 连接和持久化 ROS 1 连接使用相同的原生引擎、模型资源、闭合命令和请求输出。每种条件均启动进程五次，执行三个预热闭合周期和十次预热位姿查询。我们在各次重复测量中轮换接口顺序。四个重置–闭合–评分–状态周期构成每项主要工作负载，其他工作负载则改变重复闭合次数、位姿扰动或数值采样数量。完整评估共包含 235 次进程启动，其中测得 975 个闭合周期和 4,875 次物体位姿查询。

我们在 Intel Core i7-13700F 上运行了这些工作负载，配置为双中央处理器配额、3 吉比字节内存限制和一个 BLAS 线程。原生客户端和 ROS 客户端在 ROS Noetic 环境中使用 Python 3.8.10，而数值采样使用经评估的数值实现所运行的 Python 3.12。两条接口路径使用相同的引擎和 Qhull 兼容性修正。引擎中始终启用闭合渲染。直接评分获取器分别刷新各自的值，而 ROS 服务则同时获取质量值对。因此，报告的时间反映了完整的请求操作路径，包括评分和事件处理。

表 9 报告了五次启动的周期成本中位数和四分位距。在全部九个主要场景中，关节坐标和物体位姿均一致。其中六个场景还符合固定的输出标准：质量绝对差异小于 ，接触点数量相同，并且在各接口间一对一配对后，手侧与物体侧相对接触点之间的距离小于 1 毫米。在这六个场景中，两种 ROS 连接模式下，ROS 中位耗时与直接Python 中位耗时的比率为 1.57 至 4.49。L6 立方体场景在不同进程实例之间的接触点数量有所变化；Barrett 立方体和圆柱体场景的体积变化则超出了固定容差。在同一接口内，不同进程启动之间也出现了相同的体积变化。在不计时的控制测试中，我们固定了一个原始力旋量集合，只改变其列顺序。尽管几何力旋量凸包保持不变，返回的原生体积仍随这些排列而变化。我们保留了所有启动结果，并报告了所有场景的绝对成本；耗时比仅针对符合输出标准的场景给出。

图7 区分了重复的原生闭合、位姿相关操作、数值评分和常驻内存。对于 L6 立方体，将闭合次数从 1 次增加到 16 次后，直接工作负载的中位数从 0.150 秒增至 2.38 秒。位姿扰动采用相同的种子，平移标准差为 1 毫米，旋转标准差为 0.25 度。每个指定的位姿都会计入测得的成本。在最大工作负载的 9 个位姿中，有 4 个位姿在每个接口中都进入了碰撞分支，因此位姿曲线包含对每个指定位姿的碰撞处理和状态查询。对于数值采样，我们固定了一个含 13 个接触点的接触集，并改变摩擦节点数。两项数值计算均为每个接触点使用 8 条锥边，力矩长度为 100 毫米。在 64 个节点时，矩阵构造耗时 105 毫秒，重新计算 Ferrari–Canny 评估耗时 1.98 秒，而沿着穿过质心的 6 个单位笛卡尔力方向计算任务力旋量容量耗时 607 毫秒。这些成本体现了位姿变化时重复形成接触与在固定接触点上评估摩擦变化之间的区别。

我们将常驻内存定义为每个完整进程树中进程 RSS 采样总和的最大值，其中包括客户端、引擎、DISPLAY、ROS 主节点和 rosout。采样器的目标间隔为 10 ms，观测到的最大间隔为 41.1 ms。共享常驻页会计入每个进程的 RSS，因此报告的总和描述的是进程树内存占用。采样到的最大总量为 768 MiB。常驻内存中位数保持在 414–

表 9：多手部接口开销。数值为周期延迟中位数，单位为毫秒，括号内为四分位距；每种条件下进行了五次进程启动。每次启动测量四个重置–关闭–评分–状态周期。匕首符号表示接触或体积变化超出固定输出容差。耗时比仅针对经过检查的行计算。粗体条目表示这些输出一致性行中的较低成本。

场景	直接 Python	ROS 1		一致性
RealHand L6 / 立方体	149.1 (0.79)	180.3 (7.60)	176.5 (1.17)	†
RealHand L6 / 球体	**73.4 (0.58)**	119.4 (1.73)	115.4 (0.86)	✓
RealHand L6 / 圆柱体	**43.7 (0.63)**	100.8 (1.84)	94.8 (1.89)	✓
Barrett / 立方体	15.6 (0.69)	38.0 (1.23)	36.2 (1.52)	†
Barrett / 球体	**8.24 (0.08)**	34.1 (0.15)	32.6 (1.86)	✓
Barrett / 圆柱体	24.6 (1.75)	43.5 (1.06)	42.7 (2.78)	†
LEAP / 立方体	**53.7 (0.73)**	226.7 (1.31)	221.5 (0.67)	✓
LEAP / 球体	**39.0 (0.38)**	175.1 (2.60)	171.2 (0.01)	✓
LEAP / 圆柱体	**65.2 (0.38)**	198.3 (2.75)	193.7（0.88）	✓

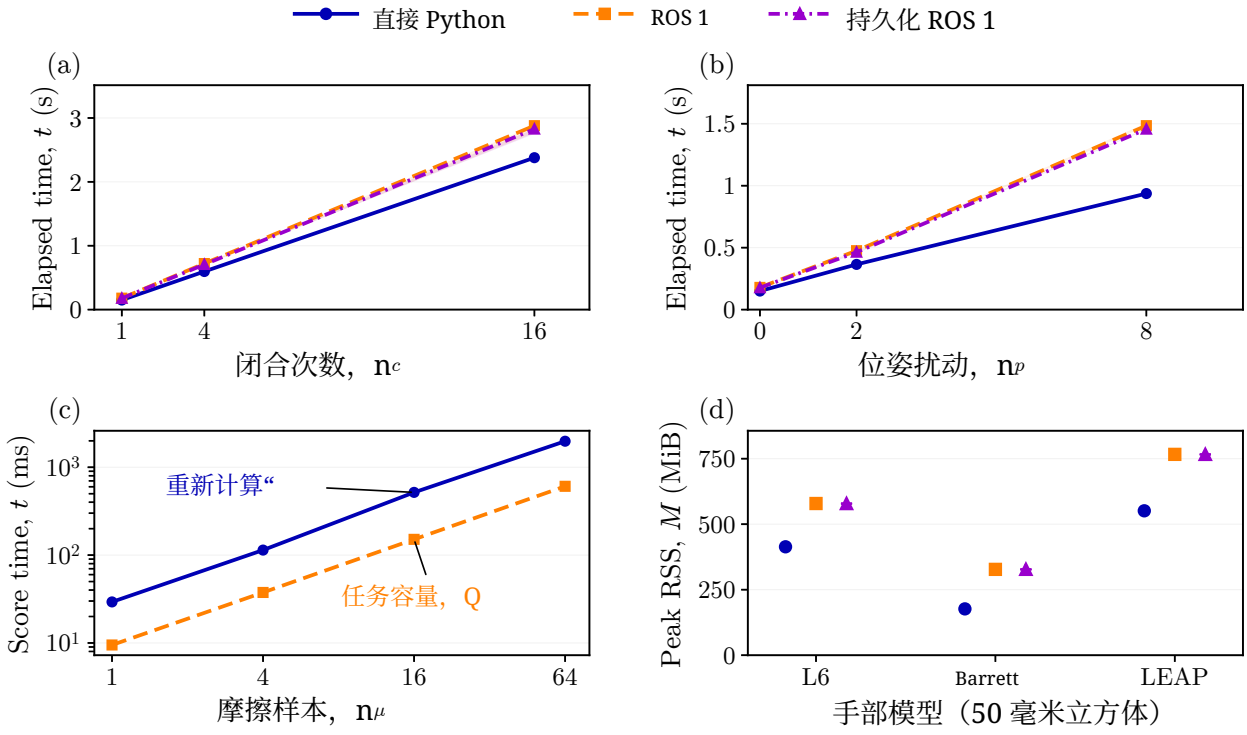


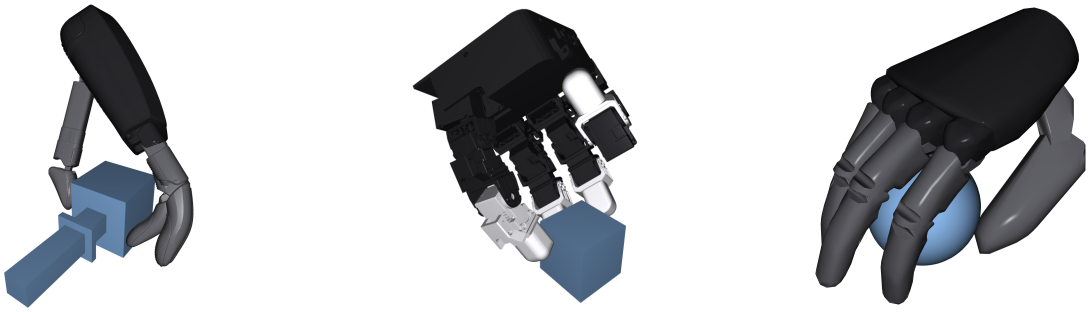
图 7：**闭合抓取、数值计算与内存成本**。五次进程启动的中位数和四分位距。（a）重复的L6立方体闭合。（b）包含碰撞处理的标称位姿，分别进行两次或八次扰动。（c）针对固定的十三接触集进行摩擦节点扫描，分别计时矩阵构造、重新计算的Ferrari–Canny质量 ε，以及对数坐标轴上的任务力旋量容量 Q。（d）立方体场景的采样进程树峰值常驻内存。（a）中的图例同样适用于（b）和（d）。各线段连接实测的工作负载规模。

416 MiB（对应 1 至 16 次重复的 L6 闭合抓取），在 1 至 64 个数值摩擦样本范围内则保持在 82.4–95.7 MiB。对于图 7 （d）中的立方体场景，直接 Python 的中位数峰值范围为 177 至 551 MiB，而 ROS 部署 的范围为 328 至 766 MiB。

4.7. 跨多种具身平台的抓取规划

在任务仿真中使用规划好的抓取，需要在臂手模型中重建其手掌位姿和关节坐标。对于 RealHand L6、LEAP 和 Ability 示例，我们使用 DexReach 重建了指令

臂手模型 [59]。相对于物体的手掌位姿和独立坐标定义了抓取；手部清单会将独立坐标展开为源铰链角。外部机械臂规划器提供到达并搬运载荷所需的运动。图 8 展示了从 LEAP 和 Ability 抓取中选定的 L6关键抓取以及最终相对于载荷的配置。



(a) RealHand L6
$\varepsilon_{\mathrm{GI}} = 0.0559$

(b) LEAP Hand
$\varepsilon_{\mathrm{GI}} = 0.0708$

(c) Ability Hand
$\varepsilon_{\mathrm{GI}} = 0.492$

图8：**三种手部形态下的抓取指令**。(a) 为双手插入重建的关键抓取。(b) 最终立方体抓持，最小上升高度为 53.6 毫米。(c) 最终球体抓持，最小上升高度为 57.8 毫米。抓持构型以各自的载荷坐标系表示。子图说明中报告了原生 GraspIt! Ferrari–Canny质量 $\varepsilon_{\mathrm{GI}}$，该质量使用各手对应的物体模型在规划闭合状态下进行评估。

对于图示的提举，我们使用了 ZArm–LEAP 搭配 50 毫米立方体，以及 Franka–Ability 搭配 60 毫米球体。两个载荷的质量均为 100 克，滑动摩擦系数均为 0.7。机械臂从张开的待机距离开始，经过闭合，再沿向上轨迹移动 60 毫米。在每次图示的两秒抓持过程中，于采样观测时均有至少三根远端手指处于接触状态。对于 Ability，在三次仿真器重置的每次重置中，全部 2,000 个抓持样本都观测到三根承担载荷的远端手指。这些接触历史和最小上升高度描述的是图示执行过程。我们在任务仿真器中指定了机械臂轨迹和动态控制器。

这些示例将数值抓取分析与命令重建和抬升执行联系起来。原生接触和关节映射标识出闭合手部构型，而数值扳手计算揭示了评分惯例。第 2.7 节中的五记录工作流展示了如何从已安装软件包和显式资源中重建命令。运动学残差衡量规划指令的还原程度，而抬升观测则衡量运动过程中的载荷保持情况。接触和坐标表示形式使这两项指标都能追溯到相应的模型和命令。

5. 结论

我们介绍了`graspit-py`，这是一个面向 GraspIt! 的 Python 接口，可提供闭合和接触量，用于评估位姿和摩擦不确定性下的抓取指令。抓取记录保留了已接受的相对物体的放置位置、闭合后的独立坐标、标称接触点和采样评分，从而将数值分析与命令重建联系起来。导出器推导仿射关节对应关系（包括参考角偏移），并保留各个碰撞组件。连杆比较结果证实，所有五种受评估手部模型均实现了微米级对应关系。受控消融实验量化了省略非零参考项贡献所造成的位移，以及组合碰撞组件后产生的占据区域。

我们评估了该接口各项抓取分析操作的数值一致性和计算成本。任务扳手容量通过系数分配线性规划进行评估，并采用统一的法向力预算和平衡扳手坐标。与独立的笛卡尔力公式、坐标变换及解析尺度算例的结果一致，证明了在所评估输入范围内的数值一致性。运行时间研究量化了三种手和三种物体几何形状下的重复闭合、位姿扰动、摩擦采样以及进程树常驻内存。输出一致性检查确定了可用于运行时间比较的场景。可执行工作流和六手示例展示了该接口从模型转换到数值评分与重建的完整流程。未来工作包括开发基于浏览器的查看器，以便检查已保存的抓取几何、坐标系、接触点和质量样本。

6. 数据可用性

源代码可在 `graspit-py` 中获取，手册 介绍了安装方法、支持的手部模型和数值接口。公开的 L6 示例 提供了可执行导出、规划和重建程序及其搜索配置。手部网格来源和许可证列于 手模型指南中。

该接口评估档案包含支持性的几何测量数据、接口计时样本、抓取构型和分析程序 [60]。

期刊补充材料包含导出器消融实验和扩展测量数据、笛卡尔力参考、用于扩展评估的数值实现，以及相应的分析程序。

资助

本研究得到GR100333号资助以及诺斯罗普·格鲁曼公司的资助。 n.

利益冲突

作者声明不存在利益冲突。

参考文献

[1] A. T. Miller, P. K. Allen, 《GraspIt!：用于机器人抓取的多功能仿真器》，IEEE《机器人与自动化杂志》11 (4) (2004) 110–122。doi:10.1109/MRA.2004.1371616。[2] R. B. Rusu, S. Cousins, 《3D 时代已至：点云库（PCL）》，载于：2011年IEEE国际机器人与自动化会议，2011，页码 1–4。doi:10.1109/ICRA.2011.5980567。URL https://pointclouds.org/assets/pdf/pcl_icra2011.pdf[3] GraspIt!开发者，graspit_interface：一个提供 ROS 接口的 GraspIt! 插件，https://github.com/graspit-simulator/graspit_interface，软件仓库（创建于 2016 年），以及 Python 客户端 graspit_commander（2016）。[4] J. Buehler，GraspIt! C++ API，https://github.com/JenniferBuehler/graspit-pkgs/wiki/GraspIt-CPP-A PI，在线文档，访问于2026年11月19日（2018）。[5] E. Todorov, T. Erez, Y. Tassa, 《MuJoCo：用于基于模型控制的物理引擎》，载于：2012年IEEE/RSJ智能机器人与系统国际会议，2012，页码 5026–5033。doi:10.1109/IROS.2012.6386109。[6] M. T. Ciocarlie, P. K. Allen, 《用于灵巧机器人抓取的手部姿态子空间》，《国际机器人研究杂志》28 (7) (2009) 851–867。doi:10.1177/0278364909105606。[7] A. H. Li, P. Culbertson, J. W. Burdick, A. D. Ames, 《FRoGGeR：通过最小权重度量快速生成鲁棒抓取》，载于：2023年IEEE/RSJ智能机器人与系统国际会议（IROS），2023，页码 6809–6816。doi:10.1109/IROS55552.2023.10341806。[8] J. Varley, J. Weisz, J. Weiss, P. Allen, 《通过深度学习生成多指机器人抓取》，载于：2015年IEEE/RSJ智能机器人与系统国际会议（IROS），2015，页码 4415–4420。doi: 10.1109/IROS.2015.7354004。[9] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, H. Wang, 《DexGraspNet：基于仿真的通用物体大规模机器人灵巧抓取数据集》，载于：2023年IEEE国际机器人与自动化会议（ICRA），2023，页码 11359–11366。[10] B. Lim, J. Kim, J. Kim, Y. Lee, F. C. Park, 《EquiGraspFlow：SE(3)等变六自由度抓取姿态生成流》，载于：第八届机器人学习会议论文集，《机器学习研究论文集》第 270 卷，2025，页码 5067–5086。URL https://proceedings.mlr.press/v270/lim25a.html[11] C. Enwerem, J. S. Baras, C. Belta, 《EquiDexFlow：接触约束的 SE(3)等变灵巧抓取生成流》，arXiv:2606.12728 (2026)。URL https://arxiv.org/abs/2606.12728

[12] Y. Yang, C. Fermüller, Y. Li, Y. Aloimonos, 《重新审视抓取类型：视觉经典特征的现代视角》，载于：2015年IEEE计算机视觉与模式识别会议（CVPR），2015，页码 400–408。doi:10.1109/CVPR.2015.7298637。[13] K. R. Barad, A. Orsula, A. Richard, J. Dentler, M. A. Olivares-Mendez, C. Martinez, 《GraspLDM：使用潜在扩散模型进行生成式六自由度抓取合成》，IEEE Access 12 (2024) 164621–164633。doi:10.1109/ACCESS.2024.3492118。[14] E. Chisari, N. Heppert, T. Welschehold, W. Burgard, A. Valada, 《CenterGrasp：用于同时进行形状重建和六自由度抓取估计的物体感知隐式表示学习》，IEEE《机器人与自动化快报》9 (6) (2024) 5094–5101。doi:10.1109/LRA.2024.3388850。[15] B. Zhao, H. Zhang, X. Lan, H. Wang, Z. Tian, N. Zheng, 《REGNet：用于点云端到端抓取检测的基于区域的抓取网络》，载于：2021年IEEE国际机器人与自动化会议（ICRA），2021，页码 13474–13480。doi:10.1109/ICRA48506.2021.9561920。[16] J. Bohg, A. Morales, T. Asfour, D. Kragic, 《数据驱动的抓取合成：综述》，IEEE Transactions On Robotics 30 (2) (2014) 289–309。doi:10.1109/TRO.2013.2289018。[17] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Ojea, K. Goldberg, 《Dex-Net 2.0：利用合成点云和解析抓取度量，通过深度学习规划鲁棒抓取》，Robotics: Science and Systems (2017)。[18] M. T. Mason, J. K. Salisbury, 《机器人手与操作力学》，马萨诸塞州剑桥：MIT Press，1985。URL http://archive.org/details/robothandsmechan00maso[19]A. Bicchi, V. Kumar, 《机器人抓取与接触：综述》，载于：2000年ICRA会议论文集。千年会议。IEEE国际机器人与自动化会议。研讨会论文集（目录号00CH37065），第 1 卷，2000，页码 348–353，第 1 卷。doi:10.1109/ROBOT.2000.844081。[20] M. A. Roa, R. Suárez, 《抓取质量度量：综述与性能》，Autonomous Robots 38 (1) (2015) 65–88。doi:10.1007/s10514-014-9402-3。[21] Y. Zheng, W.-H. Qian, 《多指抓取中接触力的限制与最小化》，Mechanism and Machine Theory 41 (10) (2006) 1243–1257。doi:10.1016/j.mechmachtheory.2005.11.001。[22] Y. Zheng, M. C. Lin, D. Manocha, 《可靠最优抓取力的计算》，IEEE Transactions On Robotics 28 (3) (2012) 619–633。doi:10.1109/TRO.2012.2183057。[23] Y. Zheng, W.-H. Qian, 《处理力闭合分析中的抓取不确定性》，《国际机器人研究杂志》24 (4) (2005) 311–327。doi:10.1177/0278364905049469。[24] J. Weisz, P. K. Allen, 《基于接触扳手空间度量的位姿误差鲁棒抓取》，载于：2012年IEEE国际机器人与自动化会议，2012，页码 557–562。doi:10.1109/ICRA.2012.6224697。[25] A. H. Li, P. Culbertson, A. D. Ames, 《PONG：通过力闭合概率解析界估计用于抓取的概率对象法向量》（2023）。arXiv:2309.16930，doi:10.48550/arXiv.2309.16930。URL https://arxiv.org/abs/2309.16930[26] A. H. Li, P. Culbertson, A. D. Ames, 《迈向灵巧抓取内在鲁棒性的解析理论》，载于：2024年IEEE/RSJ智能机器人与系统国际会议（IROS），2024，页码 2992–2999。doi:10.1109/IROS58592.2024.10802185。[27] D. Turpin, L. Wang, E. Heiden, Y.-C. Chen, M. Macklin, S. Tsogkas, S. Dickinson, A. Garg, 《Grasp'D：面向多指手的可微分接触丰富型抓取合成》，载于：S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, T. Hassner（编），《计算机视觉——ECCV 2022》，2022，页码 201–221。doi:10.1007/978-3-031-2 0068-7_12.[28] T. Tsuji, K. Baba, K. Tahara, K. Harada, K. Morooka, R. Kurazume, 《利用势能分析弹性指尖的抓取稳定性》，载于：2014年IEEE/SICE系统集成国际研讨会，2014，页码 453–458。doi:10.1109/SII.2014.7028081。[29] M. Lauri, D. Hsu, J. Pajarinen, 《机器人学中的部分可观测马尔可夫决策过程：综述》，IEEE Transactions On Robotics 39 (1) (2023) 21–40。doi:10.1109/TRO.2022.3200138。URL http://arxiv.org/abs/2209.10342

[30] J. Zhou, R. Paolini, A. M. Johnson, J. A. Bagnell, M. T. Mason，《不确定性下平面抓取的概率规划框架》，IEEE《机器人与自动化快报》2 (4) (2017) 2111–2118。doi:10.1109/LR A.2017.2720845。[31] C. Zito, V. Ortenzi, M. Adjigble, M. Kopicki, R. Stolkin, J. L. Wyatt，《基于假设的灵巧抓取信念规划》，预印本，arXiv:1903.05517 (2019)。doi:10.48550/arXiv.1903.05517。URL http://arxiv.org/abs/1903.05517[32] C. Enwerem, S. Kalyanaraman, J. S. Baras, C. Belta，《多模态不确定性下鲁棒灵巧抓取的变分神经信念参数化》，预印本，arXiv:2604.25897 (2026)。[33] C. Enwerem, E. Noorani, J. S. Baras, B. M. Sadler，《通过风险敏感增量采样进行鲁棒随机最短路径规划》，载于：2024 IEEE第63届决策与控制会议 (CDC)，2024，第1087–1094页。[34] C. Enwerem, J. S. Baras, C. Belta，《FIRMGrasp：用于鲁棒抓取合成的摩擦信息风险裕度》，arXiv:2607.25049 (2026)。URL https://arxiv.org/abs/2607.25049[35] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo，《机器人学：建模、规划与控制》，Springer，2010。[36] J. Buehler，urdf2graspit：从URDF到GraspIt!的转换，软件，版本1.2.0 (2018)。URL https://github.com/JenniferBuehler/graspit-pkgs/wiki/urdf2graspit[37] D. Morrison, J. Leitner, P. Corke，《闭环机器人抓取：一种实时生成式抓取合成方法》，载于：机器人学：科学与系统第十四届会议，2018，论文21。doi:10.15607/RSS.2018.XIV. 021。[38] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Van-houcke, S. Levine，《面向视觉机器人操作的可扩展深度强化学习》，载于：第2届机器人学习会议论文集，第87卷，2018，第651–673页。URL https://proceedings.mlr.press/v87/kalashnikov18a.html[39] H. Dang, J. Weisz, P. K. Allen，《盲抓取：利用触觉反馈和手部运动学实现稳定的机器人抓取》，载于：2011年IEEE国际机器人与自动化会议，2011，第5917–5922页。doi: 10.1109/ICRA.2011.5979679。[40] R. Calandra, A. Owens, D. Jayaraman, J. Lin, W. Yuan, J. Malik, E. H. Adelson, S. Levine，《不止一种感觉：利用视觉与触觉学习抓取和重新抓取》，IEEE《机器人与自动化快报》3 (4) (2018) 3300–3307。doi:10.1109/LRA.2018.2852779。[41] Z. Deng, Y. Jonetzko, L. Zhang, J. Zhang，《通过触觉感知实现多指机器人手的抓取力控制与物体稳定》，Sensors 20 (4) (2020) 1050。doi:10.3390/s20041050。[42] F. Veiga, B. Edin, J. Peters，《通过独立的手指触觉反馈控制实现抓握稳定》，Sensors 20 (6) (2020) 1748。doi:10.3390/s20061748。[43] Z. Si, Z. Zhu, A. Agarwal, S. Anderson, W. Yuan，《利用触觉感知的仿真到现实迁移预测抓取稳定性》，载于：2022年IEEE/RSJ智能机器人与系统国际会议 (IROS)，2022，第7809–7816页。doi:10.1109/IROS47612.2022.9981863。[44] J. Kim, K. Iwamoto, J. J. Kuffner, Y. Ota, N. S. Pollard，《位姿不确定性下基于物理的抓取质量评估》，IEEE Transactions on Robotics 29 (6) (2013) 1424–1439。doi:10.1109/TRO.2013.2273846。[45] S. Chen, J. Bohg, C. K. Liu，《SpringGrasp：形状不确定性下柔顺灵巧抓取的合成》，载于：机器人学：科学与系统会议论文集，荷兰代尔夫特，2024，论文42。doi:10.15607/RSS.202 4.XX.042。[46] Y. Chen, K. Kimble, H. H. Qian, P. Chanrungmaneekul, R. Seney, K. Hang，《通过柔顺式交互式高接触操作，在不确定性下实现鲁棒销孔装配》，载于：机器人学：科学与系统第二十一届会议，2025，论文60。doi:10.15607/RSS.2025.XXI.060。[47] J. Fan, L. Jin, Z. Xie, S. Li, Y. Zheng，《冗余机械臂的数据驱动运动力控制方案：运动学视角》，IEEE Transactions On Industrial Informatics 18 (8) (2022) 5338–5347。doi:10.110 9/TII.2021.3125449。

[48] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, A. Garg, 《Orbit：交互式机器人学习环境的统一仿真框架》，IEEE《机器人与自动化快报》8 (6) (2023) 3740–3747。doi:10.1109/LRA.2023.32700 34。[49] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau等, 《使用NumPy进行数组编程》，《自然》585 (2020) 357–362。doi:10.1038/s41586-020-2649-2。[50] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau等, 《SciPy 1.0：Python科学计算的基础算法》，《自然·方法》17 (2020) 261–272。doi:10.1038/s41592-0 19-0686-2。[51] Google DeepMind，MuJoCo XML参考，MuJoCo文档，版本3.4.0，关节等式约束；访问日期：2026年11月19日 (n.d.)。URL https://mujoco.readthedocs.io/en/3.4.0/XMLreference.html#equality-joint[52] Google DeepMind，MuJoCo XML参考，MuJoCo文档，版本3.4.0，网格资源与编译后的凸包；访问日期：2026年11月19日 (n.d.)。URL https://mujoco.readthedocs.io/en/3.4.0/XMLreference.html#asset-mesh[53] R. M. Murray, Z. Li, S. S. Sastry, 《机器人操作数学导论》，CRC出版社，1994。[54] E. Rimon, J. Burdick, 《机器人抓取力学》，剑桥大学出版社，2019。doi:10.1017/97 81108552011。[55] C. Ferrari, J. Canny, 《最优抓取规划》，载于：1992年IEEE机器人与自动化国际会议论文集，1992，第2290–2295页。doi:10.1109/ROBOT.1992.219918。[56] Z. Li, S. S. Sastry, 《多指机器人手的面向任务的最优抓取》，IEEE机器人与自动化期刊4 (1) (1988) 32–44。doi:10.1109/56.769。[57] C. Borst, M. Fischer, G. Hirzinger, 《抓取规划：如何选择合适的任务扳手空间》，载于：IEEE机器人与自动化国际会议 (ICRA) 论文集，第1卷，2004，第319–325页。doi:10.1109/ROBOT.2004.1307170。[58] R. T. Rockafellar, S. Uryasev, 《条件风险价值的优化》，《风险期刊》2 (3) (2000) 21–41。doi:10.21314/JOR.2000.038。[59] C. Enwerem, J. S. Baras, C. Belta, 《无需规划器的抓取执行：构型空间抓取距离场，实现经认证的安全性与有保证的质量》，arXiv:2608.00600 (2026)。URL https://arxiv.org/abs/2608.00600[60] C. Enwerem, J. S. Baras, C. Belta，graspit的接口评估数据_python_包装器，GitHub发行版v0.2.0，辅助数据集 (2026)。URL https://github.com/coenwerem/graspit-py/releases/download/v0.2.0/interface-evaluation-d ata.zip