

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

H-MaP: An Iterative and Hybrid Sequential Manipulation Planner

Berk Cicek^{1,†}, Arda Sarp Yenicesu^{1,†}, Cankut Bora Tuncer^{1,†}, Kutay Demiray¹, and Ozgur S. Oguz¹

Abstract—This paper introduces H-MaP, a hybrid sequential manipulation planner that addresses complex tasks requiring both sequential actions and dynamic contact mode switches. Our approach reduces configuration space dimensionality by decoupling object trajectory planning from manipulation planning through object-based waypoint generation, informed contact sampling, and optimization-based motion planning. This architecture enables handling of challenging scenarios involving tool use, auxiliary object manipulation, and bimanual coordination. Experimental results across seven diverse tasks demonstrate H-MaP’s superior performance compared to existing methods, particularly in highly constrained environments where traditional approaches fail due to local minima or scalability issues. The planner’s effectiveness is validated through both simulation and real-robot experiments. <https://sites.google.com/view/h-map/>

Index Terms—Constrained Motion Planning, Manipulation Planning, Motion and Path Planning

I. INTRODUCTION

SEQUENTIAL manipulation is fundamental to robotics, enabling robots to execute complex, multi-step processes by chaining individual actions. Sequential manipulation is fundamental to robotics, and developing robust capabilities requires addressing several significant challenges.

A primary challenge lies in creating generalized methods that can address diverse task scenarios. Consider a task where a robot must pick up a stick and then use it to push an object (Fig. 1). The solution framework must effectively handle such combinations of distinct actions while maintaining flexibility across different task configurations. While [1] made progress in addressing this challenge through an optimization-based approach using kinematic mode abstraction, their method faces notable limitations. In particular, the optimization framework is susceptible to becoming trapped in local minima when operating in constrained environments, compromising its robustness. This limitation becomes evident in scenarios such as pushing an object through a tunnel and retrieving it from the other side (Fig. 3-a).

Such manipulation tasks in constrained environments necessitate dynamic contact mode switches. By “dynamic,” we refer to switching contact modes online based on kinematics, rather than employing a full physical dynamics model. Recent advances in model-based dexterous manipulation have addressed



Fig. 1: (Top) Examples of tool manipulation in nature: a crow using a tool to obtain food [5] and manipulating a sliding bolt latch, alongside a Franka Panda robot performing an analogous obstacle removal task. (Bottom) Complex manipulation scenarios solved by our proposed approach: tool-assisted object retrieval through pushing and picking, latch mechanism manipulation, and tool-based obstacle removal.

this requirement through solvers that explicitly incorporate these mode transitions. These approaches typically achieve active mode switching through contact mode sampling, which helps identify global solutions [2]–[4].

However, current dexterous manipulation methods face significant scalability challenges. This limitation stems from the high dimensionality of their configuration spaces, which must account for both manipulator and object states simultaneously. As a result, these methods struggle to generate extended manipulation sequences efficiently. Furthermore, their applicability to complex scenarios involving tool use and auxiliary object manipulation remains largely unexplored.

Therefore, developing a modular and robust manipulation planner requires addressing two fundamental criteria: the ability to effectively handle sequences of distinct actions and the capacity to generate dynamic contact mode switches.

Drawing inspiration from human physical manipulation capabilities offers promising directions for addressing these sequential manipulation challenges. The human brain demonstrates remarkable capabilities: anticipating outcomes of environmental interactions [6], predicting object trajectories [7], and simulating action consequences [8]. Furthermore, humans naturally determine optimal contact points during object manipulation [9]. These biological insights helped inform the development of our approach.

This paper introduces an Iterative and Hybrid Sequential Manipulation Planner (H-MaP) that combines optimization-based methods [10] with sampling-based approaches to enable complex constrained manipulation planning. The key contribution lies in our object-based trajectory planning methodology and the joint optimization of manipulator and object motion. This hybrid approach effectively addresses two critical challenges in constrained manipulation: the dimensionality explosion of configuration space and the susceptibility to local minima.

H-MaP employs an iterative planning strategy based on two

Manuscript received: November, 9, 2024; Revised January, 29, 2025; Accepted March, 5, 2025.

This paper was recommended for publication by Editor Júlia Borràs Sol upon evaluation of the Associate Editor and Reviewers’ comments. This work was supported by TUBITAK under 2232 program with project number 121C148 (“LiRA”).

[†]These authors contributed equally to this work.

Berk Cicek, Arda Sarp Yenicesu, Cankut Bora Tuncer, Kutay Demiray, and Ozgur S. Oguz are with the Department of Computer Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey. Email: berk.cicek@bilkent.edu.tr

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

fundamental concepts: *waypoints* and *contact points*, representing discrete object poses and physical interaction points, respectively. The iterative computation of contact points between waypoints enhances solver robustness while decomposing complex tasks into manageable segments. Our proposed informed contact sampling methodology ensures the feasibility of contact points relative to waypoints, while contact modes guide the dynamic adjustment of robot-object interactions during motion planning. This integration of waypoint and contact point information within an optimization framework enables the solver to execute sophisticated sequential manipulation tasks.

Our key contributions are summarized as follows:

- A hybrid manipulation planner that combines sampling- and optimization-based methods to solve complex constrained tasks, including scenarios involving auxiliary objects and tool use.
- An iterative planning approach using waypoints and contact points that enhances both modularity and robustness in physical sequential manipulation planning.
- A novel contact sampling methodology that enables effective integration of decoupled waypoint and contact point representations.

We evaluate our proposed planner against existing approaches across seven diverse constrained manipulation tasks, including both single-arm and bimanual scenarios, to demonstrate its scalability. The effectiveness of our method is validated through both simulation studies and real-robot experiments, confirming its practical applicability. Additionally, we provide a dataset of successful manipulation examples used for training our contact point inference model, which can serve as a benchmark for future research in learning-based manipulation planning.

II. RELATED WORK

A. Optimization-Based Manipulation Planning

Manipulation planning encompasses the determination of motion sequences and actions that enable robots to interact effectively with their environment. Optimization-based approaches in this domain seek to generate optimal trajectories by minimizing or maximizing objective functions subject to specific constraints, incorporating criteria such as path efficiency, energy conservation, and obstacle avoidance.

Several significant optimization frameworks have emerged in this field. CHOMP [11] and its derivatives [12], [13] utilize covariant gradient descent for cost functional optimization. STOMP [14] addresses non-differentiable costs through stochastic sampling approaches, while TrajOpt [15] implements sequential quadratic programming with continuous-time collision checking. Logic-Geometric Programming (LGP) [16], built upon KOMO [10], extends these concepts by integrating logic-controlled constraints within geometric optimization problems.

Despite their computational efficiency, these optimization-based methods share a fundamental limitation: they are inherently restricted to finding locally optimal solutions, a limitation that becomes particularly problematic in constrained environments where the solution space is complex and discontinuous.

B. Sampling-Based Manipulation Planning

Sampling-based manipulation planning approaches explore high-dimensional spaces through random sampling to generate diverse action sequences, enabling complex robotic interactions. Notable methods such as CBiRRT [17], IMACS [18], and RMR* [19] focus their search on combinations of primitive actions. However, these methods' reliance on predefined motion primitives inherently constrains robotic dexterity due to their finite solution sets [20].

Recent advances, exemplified by CMGMP [3], have enhanced manipulation capabilities through automatic generation of motion primitives. By integrating rapidly exploring random trees (RRT) [21] with dynamic simulations that consider contact modes (sticking/sliding) and object dynamics, these methods successfully address challenging tasks such as flat object manipulation and sliding in constrained spaces.

While sampling-based approaches offer robust solutions, they face significant computational challenges stemming from the high dimensionality of configuration spaces and the necessity of post-processing for trajectory smoothing.

C. Constrained Sequential Manipulation & Object-Centric Frameworks for Planning

Constrained Sequential Manipulation Planning requires generating action sequences for robots to accomplish complex tasks while maintaining compliance with environmental and task-specific constraints, which restrict both object movement and available contact modes. Success in these scenarios critically depends on precise robot-object contact points, with an additional layer of complexity introduced by tasks requiring dynamic contact mode transitions.

Recent approaches have addressed different aspects of these challenges. Sampling-based methods such as CMGMP [3] and [2] successfully handle dynamic contact mode switches but lack comprehensive sequential manipulation capabilities. Conversely, existing sequential manipulation planners [1], [22] achieve task sequencing but cannot actively manage contact mode transitions.

In parallel, object-centric frameworks have been explored for rearrangement tasks in dynamic or partially known environments [23]–[25]. These methods often rely on Task and Motion Planning (TAMP) or kinodynamic tree search to plan object-centric actions for rearrangement or clustering. While effective for high-level sequencing, they typically do not address detailed constrained motion or tool-use transitions.

This paper addresses this gap by introducing a novel solver that unifies constrained and sequential manipulation planning capabilities, integrating object-based planning with dynamic contact mode switches. This represents the first comprehensive approach that extends beyond purely TAMP-oriented or kinodynamic rearrangement solutions to achieve unified constrained sequential manipulation.

III. PRELIMINARIES

We first explain the k -order motion optimization formulation, KOMO [10], a Non-Linear Programming (NLP) solver. Subsequently, we introduce our problem formulation.

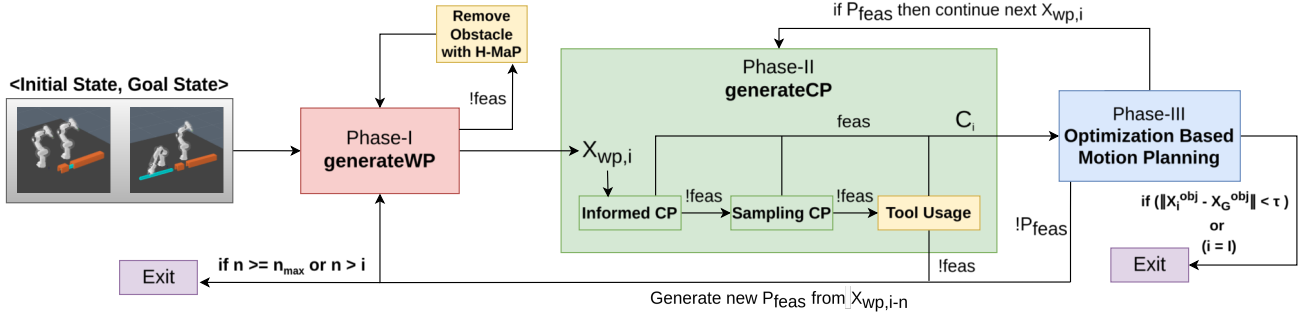


Fig. 2: System flowchart of H-MaP’s three-phase architecture: (I) Bi-RRT-based waypoint generation with obstacle handling, (II) hybrid contact point determination through learning and sampling, and (III) optimization-based motion planning. The system supports both standard end-effector (ID_{manip_0}) and tool-based manipulations (ID_{manip_n}), enabling dynamic replanning and recursive obstacle handling for single and multi-tool scenarios.

A. Optimization Based Motion Planning with KOMO

KOMO focuses on optimizing robot trajectories through a general problem formulation that integrates constraints on movement. Therefore, it aims to find the most efficient path while complying with predefined physical and environmental constraints by solving the k-order non-linear optimization problem formulated in (1).

$$\begin{aligned} \min_{x_{0:T}} \quad & \sum_{t=0}^T f_t(x_{t-k:t})^T f_t(x_{t-k:t}) \\ \text{s.t.} \quad & \forall t : g_t(x_{t-k:t}) \leq 0, \quad h_t(x_{t-k:t}) = 0. \end{aligned} \quad (1)$$

Here, $x_t \in \mathbb{R}^n$ denotes a configuration of robot and objects, while $x_{0:T} = (x_0, \dots, x_T)$ represents a trajectory spanning a horizon T . The expression $x_{t-k:t} = (x_{t-k}, \dots, x_{t-1}, x_t)$ represents sequences of $k+1$ consecutive states, indicating the progression of states over time within the given framework. The functions $f_t(x_{t-k:t})$, $g_t(x_{t-k:t})$, and $h_t(x_{t-k:t})$, each mapping to \mathbb{R}^{d_t} , \mathbb{R}^{m_t} , and \mathbb{R}^{l_t} respectively, are designed as arbitrary, first-order differentiable, non-linear, k-order, vector-valued functions. These functions serve to define cost metrics or set equality/inequality constraints for each timestep t .

B. Problem Formulation

We consider the problem of prehensile and non-prehensile manipulation of rigid objects and tools with a robotic manipulator. For problem formulation, we adopt a constraint optimization framework with kinematic modes as outlined in [16]. We use KOMO as an NLP solver for motion optimization and incorporates kinematic switches for sequential manipulation.

We extend the problem formulation of [1], [26] by discretizing the path optimization. We optimize a path $x_{0:T_I} : [0, I] \rightarrow \mathcal{X}$ consisting of $I \in \mathbb{N}$ discrete states or waypoints. For each of the consecutive waypoints, we optimize a sub-path $x_{0:T_i}^i : [0, K_i T_i] \rightarrow \mathcal{X}_i$ consisting of $K_i \in \mathbb{N}$ phases or modes. For sub-paths, we define a time interval $[0, T_i]$; thus the sub-path is given by $x_{0:T_i}^i$. The overall trajectory is then the concatenation of all sub-paths $\oplus_{i=0}^I x_{0:T_i}^i$. We assume that $\sum_{i=0}^I T_i = T$, which denotes the total planning horizon. For convenience we indicate path $x_{0:T_I}$ as P and sub-paths $x_{0:T_i}^i$ as P_i .

We define the optimization problem for each path segment $x_{0:T_i}^i$ by employing $s_{k,i}$, which specifies the constraints and

cost parameters for a particular phase k_i along the path. A sequence of discrete variables $s_{1:K_i}$ is designated as a *skeleton* [16]. For each sub-path P_i , a different skeleton can be used. The configuration space, denoted as \mathcal{X} , spans both the n -dimensional joint space of the robot and the poses of m rigid objects within $SE(3)^m$, starting from an initial configuration x_0^i within \mathcal{X} .

Given a skeleton $s_{1:K_i}$, where $\forall_i \in [0, I]$, we solve the optimization problem

$$\min_{x_{0:T_i}^i : [0, K_i T_i]} \int_0^{K_i T_i} f_{\text{path}}(\bar{x}_i(t), s_{k,i}) dt + f_{\text{goal}}(x_i(T_i), s_{k,i}) \quad (2a)$$

$$\text{s.t.} \quad x_i(0) = x_0^i, h_{\text{goal}}(x_i(T_i)) = 0, g_{\text{goal}}(x_i(T_i)) \leq 0, \quad (2b)$$

$$\forall t \in [0, T_i] : \quad h_{\text{path}}(\bar{x}_i(t), s_{k,i}(t)) = 0, \quad (2c)$$

$$g_{\text{path}}(\bar{x}_i(t), s_{k,i}(t)) \leq 0, \quad (2d)$$

$$\forall k \in \{1, \dots, K_i\} : h_{\text{switch}}(\hat{x}_i(t_k), s_{k-1,i}, s_{k,i}) = 0, \quad (2d)$$

$$g_{\text{switch}}(\hat{x}_i(t_k), s_{k-1,i}, s_{k,i}) \leq 0,$$

Within this model, the path constraints, labeled as $(h, g)_{\text{path}}$, rely on the comprehensive state $\bar{x}(t) = (x(t), \dot{x}(t), \ddot{x}(t))$ to validate that the trajectory adheres to kinematic constraints, ensuring feasible motion paths. The cost function for the path, f_{path} , incorporates control efforts, specifically chosen as the sum of squared joint accelerations to evaluate and minimize the effort required for path execution. The terms $(f, h, g)_{\text{goal}}$ denote a variety of objectives or conditions that the final state of the system is required to achieve or satisfy, ensuring the fulfillment of specified end goals. The terms $(h, g)_{\text{switch}}$ specify the constraints that facilitate smooth transitions between different operational modes (e.g., tool-use and stable contact with object), s_{k-1} and s_k , relying on the extended state $\hat{x} = (x, \dot{x})$ to ensure these transitions are both feasible and differentiable.

IV. ITERATIVE HYBRID MANIPULATION PLANNER

A. Overview

Fig. 2 illustrates our iterative Hybrid sequential Manipulation Planner (H-MaP), which comprises three fundamental phases: waypoint generation, contact point sampling, and optimization-based motion planning.

The waypoint generation phase constructs a sequence of intermediate poses $\{X_{\text{wp}}\}^{(0:I)}$ for the target object, defining discrete states from the initial pose (X_0^{obj}) to the goal pose (X_G^{obj}). This decomposition strategy serves two key purposes:

Algorithm 1: H-MaP Algorithm

Input: X_0 (initial configuration), X_G^{obj} (goal pose), τ (goal threshold), j_{max} (maximum number of iterations)
Output: P (the final motion plan)

```

1  $P_{\text{feas}} \leftarrow \text{False}$ 
2 while  $\neg P_{\text{feas}}$  do
3    $X_{\text{wp}} \leftarrow \text{ObjectBasedRRT}(X_0, X_G^{\text{obj}})$ 
4    $I \leftarrow |X_{\text{wp}}|$   $\triangleright$  number of waypoints
5    $i \leftarrow 0$ 
6   while  $(\|X_i^{\text{obj}} - X_G^{\text{obj}}\| > \tau) \wedge (i \neq I)$  do
7      $j \leftarrow 0$   $\triangleright$  reset iteration number of inner loop
8      $P_{\text{feas}} \leftarrow \text{False}$   $\triangleright$  re-initialize the flag
9     while  $(\neg P_{\text{feas}}) \wedge (j < j_{\text{max}})$  do
10       $C \leftarrow \text{GenerateCP}(X_i, ID_{\text{obj}}, ID_{\text{manip}})$ 
11       $(X_{i+1}, P_i, P_{\text{feas}}) \leftarrow \text{KOMO}(X_i, C, ID_{\text{manip}}, X_{\text{wp}, i})$ 
12       $j \leftarrow j + 1$ 
13      if  $P_{\text{feas}}$  then
14         $P \leftarrow P \oplus P_i$ 
15         $i \leftarrow i + 1$ 
16      else
17        break
18 return  $P$ 

```

it breaks down complex manipulation tasks into manageable segments and enhances scalability by initially planning in object space, independent of robot configurations.

In the contact point sampling phase, we determine the feasible physical interaction locations between the manipulator and the target object at the intermediate poses generated by the waypoint generation phase. By default, the manipulator (ID_{manip}) is the robot's end-effector, except in tool-use scenarios where the tool serves as the manipulator (Fig. 2). We enhance the contact sampling methodology from [2] with a learning-informed approach to efficiently identify feasible contacts. This phase guides kinematic mode transitions and helps avoid local minima while enabling an initially decoupled object-manipulator motion planning.

Leveraging the sampled waypoints, our approach first determines feasible object trajectories in constrained environments. The final phase then transforms these trajectories into robot motion plans by merging the decoupled waypoints and contact points using an optimization-based solver [10]. This architecture enables H-MaP to generate plans between consecutive waypoints (addressing sequential action challenges) while utilizing sampled contact points to manage dynamic contact mode transitions. The complete iterative algorithm is presented in Algorithm 1.

For scenarios requiring obstacle removal (as shown in the flowchart's recursive obstacle handling), H-MaP recursively plans the motion of obstacle objects to predefined removal poses. Each obstacle is treated as a new target object for a separate instance of H-MaP, with the removal poses serving as goal configurations. This recursive planning continues until the path for the original target object becomes feasible. The predefined removal poses are selected to ensure they do not interfere with subsequent manipulation tasks while remaining within the robot's workspace.

B. Waypoint Generation

The waypoint generation phase reduces problem complexity by focusing solely on the target object's (ID_{obj}) trajectory, thereby limiting the configuration space dimensionality. We employ Bi-RRT [27] to compute the object's path, treating it as a freely moving body in 3D space without considering robot-object interactions. The only constraints considered during this phase are the geometric limits imposed by the task environment and objects.

The generated path undergoes optimization to ensure both optimality and smoothness while maintaining the probabilistic completeness inherited from Bi-RRT. From this optimized path, we extract waypoints X_{wp} that represent the target object's trajectory. The complete waypoint generation algorithm is shown in Algorithm 2.

Algorithm 2: ObjectBasedRRT

Input: X_0 (initial configuration), X_G^{obj} (goal pose)
Output: X_{wp} (List of Waypoints)

```

1 while  $P_{\text{sampled}} \neq \text{Feasible}$  do
2    $P_{\text{sampled}} \leftarrow \text{RRT}(X_0^{\text{obj}}, X_G^{\text{obj}})$ 
3    $P_{\text{optimized}} \leftarrow \text{KOMO}(P_{\text{sampled}})$ 
4    $X_{\text{wp}} \leftarrow \text{ExtractWPs}(P_{\text{optimized}})$ 
5 return  $X_{\text{wp}}$ 

```

C. Contact Point Generation

To generate a feasible motion plan, we must establish physical interactions between the robot and the target object's waypoint trajectory. Our approach combines learning-based inference with sampling methods to efficiently generate valid contact points.

Initially, we employ an Artificial Neural Network (ANN) to propose contact points in 6-dimensional space (3D position and Euler angles) relative to target object frame based on the current configuration space. At step k , this configuration comprises the target object pose (X_k^{obj}) and remaining object poses in the environment (X_0^{env}) (obstacles, robots, tools etc.) present in the environment, which serve as features for our ANN. The model is trained on successful manipulation examples, where labeled contacts represent feasible contact points for each waypoint-defined configuration state. Each inferred contact point undergoes feasibility verification using a NLP solver in an inverse kinematics (IK) setting to ensure the end-effector can achieve the proposed pose relative to the current intermediate waypoint. If the inferred contact is feasible, it is accepted as a valid contact point, confirming that the trajectory plan based on waypoints can be transformed into a robot motion plan.

When the proposed contact point proves infeasible, we fall back to uniform sampling over the target object's point cloud, obtained using a depth camera with known object mask. The iterative contact sampling process continues until a viable solution is identified, terminating after a predetermined threshold of iterations (in our experimental trials, $m_{\text{max}} = 20$). This hybrid approach leverages informed contact point proposals to reduce generation time while maintaining robustness through sampling-based validation. The integration of a learning model

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

enhances both computational efficiency and scalability through supervised learning, particularly beneficial for complex manipulation scenarios. The complete contact point generation process is detailed in Algorithm 3.

Algorithm 3: GenerateCP

Input: X_k (configuration at step k), ID_{obj} (selected (target) object name), ID_{manip} (selected manipulator name)
Output: C (Contact Point)

```

1  $C \leftarrow \emptyset$ 
2  $feasible_{contact} \leftarrow \text{False}$ 
3  $C_{informed} \leftarrow \text{ANN}(X_k)$ 
4 if  $KOMO(X_k, C_{informed}, ID_{manip})$  then
5    $C \leftarrow C_{informed}$ 
6    $feasible_{contact} \leftarrow \text{True}$ 
7  $PointCloud \leftarrow \text{getCameraView}(X_k, ID_{obj})$ 
8  $m \leftarrow 0$ 
9 while  $(\neg feasible_{contact}) \wedge (m < m_{max})$  do
10    $C_{sampled} \leftarrow \text{sampleContactPoint}(PointCloud)$ 
11   if  $KOMO(X_k, C_{sampled}, ID_{manip})$  then
12      $C \leftarrow C_{sampled}$ 
13      $feasible_{contact} \leftarrow \text{True}$ 
14    $m \leftarrow m + 1$ 
15 return  $C$ 

```

D. Optimization-Based Motion Planning

The motion planning phase employs a two-stage optimization approach. The first stage verifies contact point feasibility, as described in Section IV-C, by determining whether the manipulator can achieve the proposed contact configuration. This verification is formulated as an inverse kinematics optimization problem:

$$\begin{aligned} \min_{x_T} \quad & f_T(x_T)^T f_T(x_T) \\ \text{s.t.} \quad & g_T(x_T) \leq 0, \quad h_T(x_T) = 0. \end{aligned} \quad (3)$$

The equality constraint $h_T(x_T) = |X_{pose}^{manip}(x_T) - C_{pose}| = 0$ enforces the touch constraint, where C_{pose} represents the pose of the generated contact point and $X_{pose}^{manip}(x_T)$ denotes the manipulator's pose at configuration x_T . The feasibility of the final configuration x_T is determined by evaluating constraint violations within the optimization process.

In the second optimization phase, motion planning is performed by optimizing from the current state to a designated sub-goal to calculate x_i as formulated in Eq.(3). We follow the constraint definitions from [1]. For contact mode switches, we employ the stable constraint: $h_{switch}(s_{k-1}) = |{}^{obj}X_{pose}^{manip}(s_{k-1}) - {}^{obj}X_{pose}^{manip}(s_k)| = 0$ which enforces a constant relative transformation between manipulator and object in the object frame, effectively adding the object to the manipulator's kinematic chain. The touch constraint: $h_T(x_T) = |X_{pose}^{manip}(x_T) - X_{pose}^{obj}(x_T)| = 0$ ensures zero distance between manipulator and object. For trajectory optimization, we add the pose equality constraint: $h_T(x_T) = |X_{pose}^{obj}(x_T) - X_{wp,i}| = 0$

After incorporating these constraints, we solve the optimization problem. This two-phased constraint optimization ensures kinematically feasible and stable grasps while following feasible paths between waypoints. The feasibility of the resulting path is evaluated based on the solver's computed constraint

violations. If the path is deemed feasible, we append the generated path P_i to the overall trajectory P ($P \oplus P_i$) and proceed to the next waypoint $X_{wp,i+1}$. However, if the path is infeasible, the solver iteratively generates new contact points C at the current waypoint $X_{wp,i}$ until finding a feasible solution, for a threshold of iterations ($j_{max}=40$ in our experiments). If contact generation fails, the algorithm falls back to waypoint generation (Sec. IV-B). We employ a simple backtracking procedure that reverts n steps to refine the existing waypoints with a smaller step size, up to n_{max} or $n > i$ times before discarding all waypoints (Fig. 2).

The proposed method's two-phase optimization strategy first solves an inverse kinematics problem to efficiently identify feasible contact configurations before addressing the full motion planning problem. This hierarchy enhances computational efficiency by establishing optimal final configurations before proceeding with trajectory optimization toward specific sub-goals $X_{wp,i+1}$. The combination of feasibility verification, iterative contact point generation, and waypoint fallback ensures robustness in highly constrained environments where traditional optimization-based approaches often fail due to local minima.

E. Extension to Dynamic and Bimanual Scenarios

H-MaP can be generalized to various complex scenarios. In our experiments, we demonstrate this adaptability through dynamic and bimanual manipulation tasks. In dynamic scenarios, where random moving obstacles appear along the planned path, the agent must replan its trajectory. We handle this by reinitializing H-MaP whenever a new object is detected in the scene using predefined object masks.

For bimanual scenarios, where task completion requires cooperation between multiple robots, we extend the contact point generation methodology described in Sec. IV-C. The extended approach predicts which robot should interact at each given contact point. If the initially selected robot cannot reach the specified contact, the system iteratively attempts the action with other available robots until finding a feasible solution.

V. EXPERIMENTS

We evaluate H-MaP's performance on various constrained manipulation tasks, comparing it with baseline methods and detailing the dataset curation process for informed contact sampling.

A. Tasks & Scenarios

We evaluate our algorithm across seven diverse manipulation tasks, inspired by real-world scenarios, that incorporate varying levels of complexity including multiple contact mode changes, obstacle interactions, tool use, auxiliary object manipulation, dynamic obstacles, and bimanual operations (Fig.3). All experiments were conducted using the RAI simulator, with validation on real-world implementations using Franka Panda robots. We detail each task below.

Object through tunnel (tunnel, Fig. 3-a): The robot must push an object through a tunnel and retrieve it from the

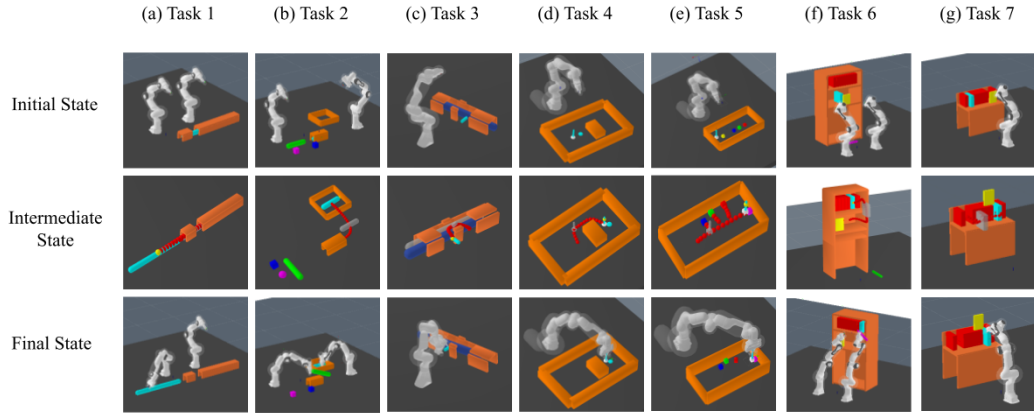


Fig. 3: From left to right: pushing an object through the tunnel; manipulating an object through the tunnel using a tool; operating a sliding latch lock; navigating an object with a tool around a fixed obstacle; and maneuvering an object with a tool while clearing movable obstacles. The top and bottom rows show initial and final configurations, respectively. The middle row displays generated waypoints (red spheres), contact points (yellow spheres), and intermediate object states (gray silhouettes) during task execution. For a complete demonstration: <https://sites.google.com/view/h-map/>

opposite side in a bimanual setting. This task requires iterative pushing actions with multiple contact point adjustments to navigate the tunnel’s constraints while avoiding collisions. The sequential nature of the task prevents direct object retrieval, necessitating coordinated push-then-pick manipulation between multiple robots.

Object through tunnel with tool (*tunnel with tools*, Fig. 3-b): A bimanual task where robots must retrieve an unreachable object from under a tunnel using an appropriate tool, followed by placing it in a box. The task complexity stems from tool selection, contact point sampling, and potential replanning due to dynamic obstacles. This extends the base tunnel task by incorporating tool-assisted manipulation.

Sliding latch lock (*bolt*, Fig. 3-c): The robot must manipulate a lock handle through combined vertical and horizontal movements. The task’s complexity arises from non-trivial rotational movements in a highly constrained environment, challenging both waypoint generation and optimization-based planning due to local minima issues.

Tool-assisted obstacle navigation (*non-movable obstacle*, Fig. 3-d): The robot must use a tool to push an object to a goal position while avoiding fixed obstacles. The task requires strategic selection of multiple pushing contact points to navigate the constrained environment.

Path clearing with tool (*movable obstacles*, Fig. 3-e): The robot must use a tool to clear movable obstacles while pushing a target object to its goal position. The task’s complexity stems from managing multiple object interactions. Initially, Bi-RRT cannot generate feasible paths due to obstacles that require clearance.

Bookshelf organization (*bookshelf*, Fig. 3-f): A bimanual task requiring obstacle removal, book placement on an upper shelf, and tool-assisted alignment. The challenge lies in identifying feasible grasp points (limited to the book’s middle region) and planning collision-free vertical transfers while avoiding local minima through coordinated waypoint and contact point planning.

Single-shelf book placement (*mini bookshelf*, Fig. 3-

g): A simplified version of the bookshelf task where the robot must place a book from left to right on a single level, requiring obstacle removal and potential replanning due to dynamically appearing obstacles. The task combines precise book manipulation with adaptive planning in a constrained environment.

Real robot validation (Fig. 1, Fig. 4): The *bookshelf*, *tunnel*, and *movable obstacles* tasks were replicated in a real robotic setting. To represent the state information accurately, we deployed Vicon motion capture cameras and an Intel RealSense depth camera. Plans generated in simulation were successfully executed in open-loop control using a Franka Panda robot.

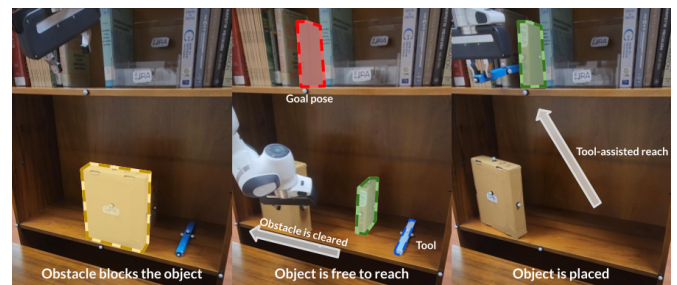


Fig. 4: Real robot execution of bookshelf task: obstacle removal and tool-assisted book placement.

B. Baselines

We compare H-MaP against KOMO [10], Bi-RRT [27], CMGMP [3], and LGP [16]. KOMO and RRT are selected as they represent the foundational optimization-based and sampling-based methods that inform our work. CMGMP is chosen to benchmark against state-of-the-art sampling methods capable of dynamic contact mode switches. LGP represents current capabilities in sequential planning. For a fair comparison, we implemented tool manipulation capabilities across all baseline methods and excluded dynamic obstacle scenarios, as they were beyond the scope of the baseline approaches. For CMGMP, we modified the point manipulator parameters to match our constrained environment requirements.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I: Comparison of success rates (out of 10 random trials) and planning times (seconds) for successful trials. (-) indicates complete failure across all trials. (*) LGP times reflect motion planning with provided task plan skeletons, excluding task planning time, thus representing a lower bound for total LGP planning time.

Method	Tunnel		Tunnel Tool		Bolt		Puck Obs Around		Puck Obs Move		Bookshelf		Bookshelf Mini	
	Success	Time	Success	Time	Success	Time	Success	Time	Success	Time	Success	Time	Success	Time
KOMO	0/10	-	1/10	58.07±14.8	2/10	5.12±2.1	5/10	2.86±0.6	0/10	-	2/10	131.25±10.7	4/10	5.43±2.2
Bi-RRT	0/10	-	0/10	-	0/10	-	8/10	0.38±0.1	0/10	-	0/10	-	0/10	-
CMGMP	0/10	-	0/10	-	0/10	-	5/10	2.29±0.1	4/10	2.30±0.2	0/10	-	0/10	-
LGP*	5/10	6.13±1.9	4/10	4.19±0.6	2/10	6.24±1.2	10/10	2.48±0.7	8/10	10.43±2.2	3/10	35.3±6.0	9/10	45.86±10.6
HMAP	10/10	5.08±2.4	10/10	6.44±4.6	10/10	4.05±0.8	10/10	21.59±6.1	10/10	26.4±8.2	10/10	18.92±12.4	10/10	6.06±0.8

In LGP’s case, we provided planning skeletons containing symbolic and geometric intermediate steps, effectively making it an informed version of KOMO.

C. Dataset Curation

To train our informed contact sampling method, we curated a dataset of successful manipulation examples in task-specific simulation environments detailed in Section V-A. HMaP contact sampling was adjusted for uniform sampling across object points, and feasibility checks were applied (Section IV-C). Generated paths were tested in simulation to ensure successful, feasible motion plans. Negative samples were excluded, leaving this for future work to enable more advanced architectures.

D. Results

1) *Comparisons*: Table I presents the performance comparison between H-MaP and baselines across seven diverse tasks. H-MaP successfully solved all tasks across 10 random trials, while baseline methods showed various limitations.

KOMO and Bi-RRT demonstrated expected limitations: optimization-based KOMO failed in complex manipulation scenarios where manifold discontinuities disrupted local Euclidean properties, leading to local minima, while Bi-RRT failed to generate feasible paths within reasonable time limits (180 seconds). CMGMP, despite being guided by contacts, faced similar scalability challenges as Bi-RRT due to the need to incorporate full manipulator configurations throughout trajectory planning.

LGP achieved moderate success by leveraging predefined symbolic and geometric intermediate states, which effectively decomposed complex tasks into subproblems. However, its reliance on optimization-based motion planning led to failures in scenarios like the bolt task, where it struggled due to the lack of fine-grained action definitions needed to find solutions in complex non-convex manifolds.

H-MaP demonstrates superior performance through its decoupled waypoint and contact point generation strategy, effectively integrating sampling-based and optimization-based methodologies. The learning-informed contact point generation mitigates hybrid planning overhead, often outperforming conventional approaches in planning time. While simpler tasks may benefit from single-approach methods, H-MaP achieves its primary objective of robust sequential manipulation with dynamic contact mode switches. The planner maintains efficacy in dynamic environments, successfully replanning trajectories when obstacles appear within a 0.3-unit radius (approximately 6 RRT extensions) of the manipulation path.

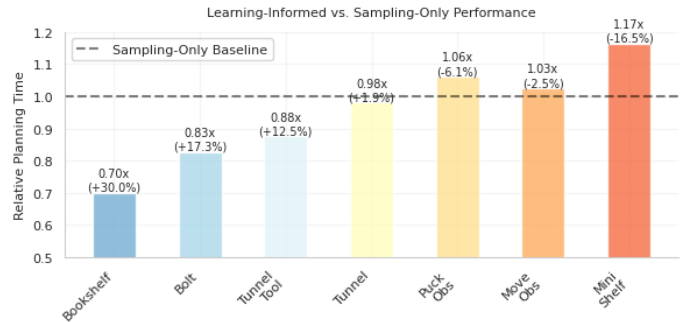


Fig. 5: Performance comparison between learning-informed and sampling-only approaches, normalized to the sampling-only baseline (1.0x). Lower bars indicate better performance, with improvements shown as percentages. Tasks are ordered by relative improvement, highlighting the effectiveness of the learning-informed approach across manipulation scenarios.

2) *Effectiveness of Informed Sampling with Learning*: As shown in Fig. 5, the learning-informed approach demonstrates a clear pattern of improvement across different manipulation tasks when compared to the sampling-only baseline. Most significantly, it reduced planning time by 30.0% for the bookshelf task (0.70x relative time) and 17.3% for the bolt task (0.83x relative time), which represent the most geometrically complex scenarios requiring precise contact selection. The tunnel tool task showed a 12.5% improvement (0.88x relative time), while maintaining comparable performance in simpler scenarios such as bookshelf mini and puck obstacle tasks where single contact points suffice. The reason for the performance decrease in simpler tasks is that overhead from occasional failed predictions may offset the benefits of learning-based sampling. This pattern suggests that our learning-informed sampling strategy provides greater benefits as task complexity increases, particularly in scenarios requiring multiple contact points and precise manipulation. The method’s ability to maintain or improve performance across all task categories, without significant degradation in any scenario, demonstrates its robustness and practical utility for complex manipulation planning.

VI. DISCUSSION & LIMITATIONS

Our approach differs from traditional methods by focusing on sampling-based planners exclusively on object path planning rather than considering combined robot-object kinematic states. This significantly reduces configuration space dimensionality, enabling effective sequential manipulation planning. Unlike existing methods that typically handle single robot-object interactions, our approach successfully manages tool use and auxiliary object manipulation.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

The results demonstrate that enhancing low-level motion planners for sequential manipulation eliminates the need for predefined motion primitives. This reduces reliance on explicit high-level action definitions, allowing implicit execution of complex actions. For instance, in the *Bolt* task, our method eliminates the need to specify discrete actions like *lift up*, *pull left*, and *pull down*. However, our solver has several limitations:

Action Limitations: Our decoupled object and robot planning introduces certain action limitations. For example, in the flip-card scenario from [3], combined planners achieve a better grasp by constraining the path to the manipulator. In contrast, our approach may lead to premature pick actions, resulting in infeasible motion. Addressing this could involve informing waypoint generation with configuration-based action constraints. **Waypoint Quality:** The solver assumes RRT-generated waypoints are viable. Poor waypoint generation may lead to solver failure despite proximity-based searching. We mitigate this by regenerating waypoints when no feasible path is found. **Kinematic Focus:** The current implementation considers only kinematic systems, simplifying optimization but neglecting system dynamics. Future work could incorporate dynamics in planning and implement controllers for execution, enhancing real-world applicability.

VII. CONCLUSION

This paper presented H-MaP, a novel hybrid sequential manipulation planner that effectively addresses two fundamental challenges in robotic manipulation: handling sequences of distinct actions and generating dynamic contact mode switches. By decoupling object trajectory planning from manipulation planning through waypoint generation and integrating learning-informed contact sampling, H-MaP significantly reduces the configuration space dimensionality while maintaining solution completeness.

Our experimental results across seven diverse manipulation tasks demonstrate H-MaP's capability to solve complex scenarios involving tool use, auxiliary object manipulation, and bimanual coordination. The planner's success in handling highly constrained environments, where traditional optimization-based methods often fail due to local minima, validates our hybrid approach. Successful real-robot implementation confirms practical applicability, while the modular architecture enables extension to dynamic and multi-robot scenarios, with future work focusing on system dynamics and backtracking strategies.

REFERENCES

- [1] M. A. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum, "Differentiable physics and stable modes for tool-use and manipulation planning," in *Proc. Robot.: Sci. Syst. Found.*, 2018.
- [2] T. Pang, H. J. T. Suh, L. Yang, and R. Tedrake, "Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models," *IEEE Trans. Robot.*, vol. 39, no. 6, pp. 4691–4711, 2023.
- [3] X. Cheng, E. Huang, Y. Hou, and M. T. Mason, "Contact mode guided motion planning for quasidynamic dexterous manipulation in 3d," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 2730–2736.
- [4] G. Lee, T. Lozano-Pérez, and L. P. Kaelbling, "Hierarchical planning for multi-contact non-prehensile manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 264–271.
- [5] D. E. McCoy, M. Schiestl, P. Neilands, R. Hassall, R. D. Gray, and A. H. Taylor, "New caledonian crows behave optimistically after using tools," *Curr. Biol.*, vol. 29, no. 16, pp. 2737–2742, 2019.
- [6] D. M. Wolpert and J. R. Flanagan, "Motor prediction," *Curr. Biol.*, vol. 11, no. 18, pp. R729–R732, 2001.
- [7] M. Kawato, "Internal models for motor control and trajectory planning," *Curr. Opin. Neurobiol.*, vol. 9, no. 6, pp. 718–727, 1999.
- [8] F. Osiurak and A. Badets, "Tool use and affordance: Manipulation-based versus reasoning-based approaches," *Psychol. Rev.*, vol. 123, no. 5, p. 534, 2016.
- [9] U. Kleinholdermann, V. H. Franz, and K. R. Gegenfurtner, "Human grasp point selection," *J. Vis.*, vol. 13, no. 8, pp. 23–23, 07 2013.
- [10] M. Toussaint, "Newton methods for k-order markov constrained motion problems," *arXiv preprint arXiv:1407.0414*, 2014.
- [11] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *Int. J. Robot. Res.*, vol. 32, no. 9-10, pp. 1164–1193, 2013.
- [12] A. D. Dragan, N. D. Ratliff, and S. S. Srinivasa, "Manipulation planning with goal sets using constrained trajectory optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 4582–4588.
- [13] A. D. Dragan, G. J. Gordon, and S. S. Srinivasa, "Learning from experience in manipulation planning: Setting the right goals," in *Proc. Robot. Res.: Int. Symp. ISRR*. Springer, 2017, pp. 309–326.
- [14] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 4569–4574.
- [15] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1251–1270, 2014.
- [16] M. Toussaint, "Logic-geometric programming: an optimization-based approach to combined task and motion planning," in *Proc. Int. Conf. Artif. Intell., ser. IJCAI'15*. AAAI Press, 2015, p. 1930–1936.
- [17] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, "Manipulation planning on constraint manifolds," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2009, pp. 625–632.
- [18] Z. Kingston, M. Moll, and L. E. Kavraki, "Exploring implicit spaces for constrained sampling-based planning," *Int. J. Robot. Res.*, vol. 38, no. 10-11, pp. 1151–1178, 2019.
- [19] P. S. Schmitt, W. Neubauer, W. Feiten, K. M. Wurm, G. V. Wichert, and W. Burgard, "Optimal, sampling-based manipulation planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 3426–3432.
- [20] Y. C. Nakamura, D. M. Troniak, A. Rodriguez, M. T. Mason, and N. S. Pollard, "The complexities of grasping in the wild," in *Proc. IEEE-RAS Int. Conf. Humanoid Robot.*, 2017, pp. 233–240.
- [21] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Research Report 9811*, 1998.
- [22] S. B. Bayraktar, A. Orthey, Z. Kingston, M. Toussaint, and L. E. Kavraki, "Solving rearrangement puzzles using path defragmentation in factored state spaces," *IEEE Robot. Autom. Lett.*, vol. 8, no. 8, pp. 4529–4536, 2023.
- [23] T. Migimatsu and J. Bohg, "Object-centric task and motion planning in dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 844–851, 2020.
- [24] J. E. King, M. Cognetti, and S. S. Srinivasa, "Rearrangement planning using object-centric and robot-centric action spaces," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 3940–3947.
- [25] K. Ren, G. Wang, A. S. Morgan, L. E. Kavraki, and K. Hang, "Object-centric kinodynamic planning for nonprehensile robot rearrangement manipulation," *arXiv preprint arXiv:2410.00261*, 2024.
- [26] M. Toussaint, J.-S. Ha, and D. Driess, "Describing physics for physical reasoning: Force-based sequential manipulation planning," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6209–6216, 2020.
- [27] J. Kuffner and S. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, 2000, pp. 995–1001 vol.2.