Manip4Care+: Constraint-RRTConnect with Learning-Based Constraint Checking for Assistive Limb Manipulation

Kai Cheng, Yi Wu, and Ahmed H. Qureshi
Department of Computer Science, Purdue University
{cheng753, wu1827, qureshi7}@purdue.edu

***Abstract*—Robotic manipulation of human limbs is an instance of articulated-object manipulation in assistive care: the robot moves not an isolated rigid body but a biomechanically constrained kinematic chain coupled through a grasp. Prior work Manip4Care introduced a modular simulation pipeline for grasping and repositioning human limbs using constrained MPPI and vector-field tracking. This paper presents Manip4Care+, which extends that pipeline with a projection-based Constraint-RRTConnect planner and a learned constraint checker for coupled robot–human states. The planner samples in robot joint space, projects candidate states onto the closed-loop contact manifold induced by the grasp, and grows bidirectional trees using projection-based steering. The same projector is then used to generate a 60,000-sample dataset for training a neural validity model over a 6-DOF UR5 and a 4-DOF human arm. Across supine and seated manipulation tasks, analytical Constraint-RRTConnect achieves 100% success in the main 5-trial-per-posture comparison with initial planning times of 0.06–0.07 s, while the original MPPI baseline requires 6.06–8.57 s. When used as a hard rejection rule, the learned checker is limited by false negatives that remove valid projected states. Follow-up on-policy retraining and recall calibration eliminate these initial planning false negatives, achieving 100% initial plan success, but reveal that the true bottleneck in complex postures shifts to execution-time replanning rather than learned feasibility estimation.**

I. Introduction

Robot-assisted care can reduce caregiver burden and improve quality of life for people with severe mobility impairments [3]. Many care activities, such as bed bathing, dressing, and physical therapy, require repositioning a human limb before another manipulation skill can be performed. This setting differs fundamentally from standard rigid-object manipulation. Once a robot grasps a forearm, the robot arm, grasp frame, and human limb form a coupled articulated system. A motion that is collision-free for the robot alone may still violate the human shoulder range of motion, break the contact geometry, or lead to an infeasible limb posture.

Previous work addresses this need by providing an open-source simulation pipeline for robotic limb manipulation [7]. Its original system combines antipodal limb grasping, MPPI-based constrained trajectory generation, vector-field trajectory following, and downstream demonstrations such as bed bathing. While the Manip4Care pipeline [7] established the feasibility of general-purpose limb repositioning, it leaves open a planning question central to assistive robotics: *how should robots plan efficiently for articulated, non-rigid, human-centered objects when validity is defined by implicit biomechanical and contact constraints?*

Manip4Care+ studies this question by integrating a constrained sampling-based planner and a learned feasibility model into Manip4Care. The original Manip4Care planner is optimization-based and reactive; it does not explicitly expose the structure of the coupled constraint manifold, and its initial planning time is often several seconds. In Manip4Care+, we model the robot and human arm as two kinematic chains connected by a fixed grasp transform, yielding a closed-loop contact constraint that can be evaluated analytically through forward and inverse kinematics. We then use that structure to implement Constraint-RRTConnect: a bidirectional planner that samples robot joint configurations, projects each candidate onto the robot–human contact manifold, and steers by repeatedly projecting interpolated waypoints. This gives the planner direct access to valid coupled states rather than relying only on unconstrained exploration or repeated trajectory optimization.

The second goal of Manip4Care+ is to examine whether a learned checker can approximate this analytical constraint. Using the projective sampler as an oracle, we collect valid projected states and invalid near-manifold states, then train a multilayer perceptron over the 10-dimensional robot–human configuration. This model is integrated back into the planner as a learned filter and, in follow-up experiments, as a calibrated high-recall predicate and a soft sampling bias.

This paper makes three contributions. First, we formulate assistive limb repositioning as projection-based planning over a closed-loop articulated contact manifold, extending the Manip4Care system. Second, we build and evaluate a neural constraint checker trained from analytical projection data. Third, we compare analytical Constraint-RRTConnect, learned-filter Constraint-RRTConnect, and the original MPPI planner in supine and seated simulation tasks, showing that analytical projection substantially reduces planning time while learned filtering requires planner-aware calibration to avoid false-negative failures.

II. Related Work

Assistive robotics has long studied devices that support mobility, rehabilitation, and personal care [3]. Exoskeletons and rehabilitation systems are effective when the device is worn by or mechanically attached to the user, but personal-care tasks such as bathing and dressing often require an external manipulator to reposition a passive or partially passive limb. Prior work has studied short-horizon limb repositioning through model predictive control [2]. Manip4Care differs by providing a modular simulation pipeline for general-purpose upper-limb repositioning and downstream assistive tasks [7].

Articulated object manipulation. A human limb is a compliant, safety-critical articulated object whose feasible motion is governed by joint limits, contact, and collision avoidance. Existing assistive benchmarks such as Assistive Gym [5] often assume a static human pose or focus on local contact control. Manip4Care and Manip4Care+ instead treat repositioning itself as a necessary manipulation primitive for exposing task-relevant regions of the body.

Constrained motion planning. Sampling-based methods on constraint manifolds are well studied [6], including projection-based planners for pose-constrained manipulation [1]. RRT-Connect [8] remains a useful template for single-query planning, but direct sampling in the ambient configuration space is inefficient when the valid set is a narrow lower-dimensional manifold. Learned models have also been used to accelerate or guide constrained planning [9]. In this work, we do not replace analytical constraints outright. Instead, we use Manip4Care+ as a controlled setting for studying when a learned feasibility model helps or hurts a projection-based planner for articulated assistive manipulation.

III. Method

A. Problem Formulation

We consider a UR5 robot arm manipulating a human upper limb in the Manip4Care simulator. Let $\mathbf{q}^R \in \mathbb{R}^6$ denote the robot configuration and $\mathbf{q}^H \in \mathbb{R}^4$ the human-arm configuration (shoulder flexion, external rotation, abduction, and elbow angle). A coupled state is

$$\mathbf{x} = [\mathbf{q}^R, \mathbf{q}^H] \in \mathbb{R}^{10},$$

subject to robot and human joint limits, collision constraints, and posture-dependent validity checks.

After grasping, the contact point is assumed fixed relative to both the robot end effector and the human limb, following the rigid non-slip grasp model of Manip4Care. Let $\mathbf{T}^w_{eef}(\mathbf{q}^R)$ be the robot end-effector pose and $\mathbf{T}^w_{elbow}(\mathbf{q}^H)$ the human elbow pose. Fixed transforms $\mathbf{T}^{eef}_{cp}$ and $\mathbf{T}^{elbow}_{cp}$ map from the end effector and elbow frames to the contact point. The robot-implied and human-implied contact poses are

$$\mathbf{T}^{w,R}_{cp} = \mathbf{T}^w_{eef}(\mathbf{q}^R)\mathbf{T}^{eef}_{cp}, \qquad \mathbf{T}^{w,H}_{cp} = \mathbf{T}^w_{elbow}(\mathbf{q}^H)\mathbf{T}^{elbow}_{cp}.$$

A valid state lies on the contact manifold when these frames agree within position and orientation tolerances and the resulting robot–human state is collision-free and biomechanically valid.

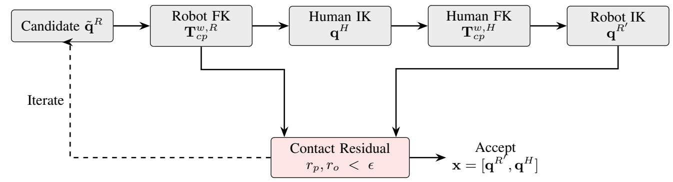


Fig. 1. Analytical projector. Alternates between robot and human kinematics to minimize the closed-loop contact residual while respecting joint limits.

B. Projection-Based Constraint-RRTConnect

Manip4Care+ uses a projection operator (Fig. 1) to map candidate robot configurations onto the coupled manifold. Given a sampled robot state $\tilde{\mathbf{q}}^R$, the projector computes the robot-implied contact point, solves human IK for the compatible limb pose, recomputes the human-implied contact frame, and solves robot IK for the corrected end-effector pose. This handshaking update is repeated until the contact residual and joint update are below tolerance. A projection is accepted only when the final state satisfies robot limits, human limits, collision checks, and contact residual thresholds.

The planner samples $\tilde{\mathbf{q}}^R$ uniformly from robot joint limits and keeps only successful projections. It then grows two trees, one from the initial coupled state and one from the goal state, as in RRT-Connect [8]. To steer from $\mathbf{q}^R_a$ toward $\mathbf{q}^R_b$, the planner interpolates in robot joint space and projects each intermediate waypoint back to the contact manifold. If a waypoint fails projection or state validation, extension stops at the last valid state. This ensures every inserted edge stays close to the coupled robot–human manifold.

C. Learning a Constraint Checker

The analytical projector also provides supervision for a neural constraint checker. We label a state positive if it is produced by successful projection and satisfies contact error, joint-limit, and simulator-validity tests. Negatives are drawn from failed projections, uniform robot–human samples, and hard perturbations around projected positives. This mixture covers both easy invalid states far from the manifold and near-boundary states that are more informative for the classifier.

The learned checker is a multilayer perceptron $f_\theta : \mathbb{R}^{10} \to \mathbb{R}$, with hidden sizes 128, 128, and 64, ReLU activations, and dropout 0.05. Inputs are normalized to $[-1, 1]$ using stored joint bounds. Because the dataset contains a 1:5 positive-to-negative ratio, the binary cross-entropy loss applies positive-class reweighting. We train with AdamW, learning rate 10^{-3}, weight decay 10^{-5}, and batch size 512 for 100 epochs, using 80% of the labeled samples for training and 20% for validation. Rather than fixing the threshold at 0.5, we sweep thresholds from 0.05 to 0.95 and select the value that maximizes F1 subject to a false-positive-rate cap of 0.1. At deployment, the checker can be used in three modes: off, filter, or learned. The main comparison uses filter mode, where analytical projection remains mandatory and the learned model only rejects projected states that it predicts to be invalid. We visualize the full pipeline in Fig. 2.

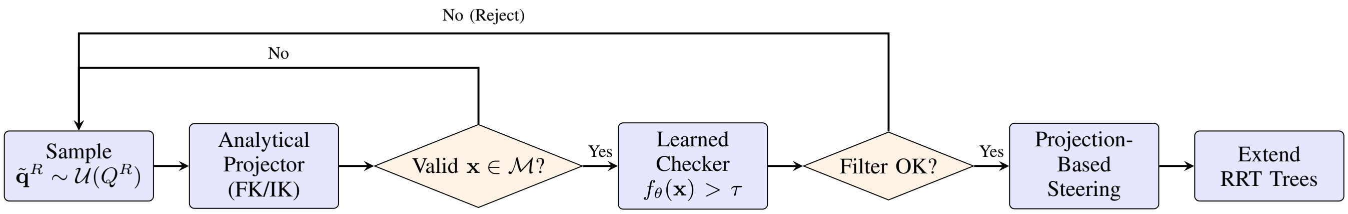
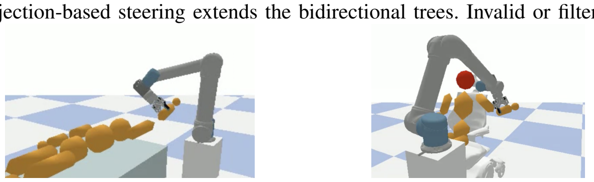


Fig. 2. Manip4Care+ planning pipeline. Candidate robot states are projected onto the coupled manifold. The learned checker acts as an optional filter before projection-based steering extends the bidirectional trees. Invalid or filtered states trigger resampling.



(a) Supine (b) Seated

Fig. 3. PyBullet simulation for manipulation in supine and seated postures.

IV. Experimental Setup

We evaluate our approach in PyBullet using the Manip4Care simulator [4, 7]. Experiments use two body postures: a human lying supine and a human seated upright. The main task is to reposition the human arm from an initial configuration to a goal configuration while maintaining the rigid contact transform and respecting biomechanical limits.

We compare 3 planners. *Analytical Constraint-RRTConnect* uses only the projection and validity checks described above. *Learned-filter Constraint-RRTConnect* adds the neural checker after analytical projection. *MPPI* is the original Manip4Care baseline, which uses constrained model-predictive trajectory generation and vector-field following. We report task success, initial planning success, initial planning time, execution time, path length, contact residual, joint-limit violation fraction, projection calls, and learned acceptance rate where applicable.

The learned checker is trained on 60,000 labeled states: 5,000 projected positives and 25,000 negatives per posture. Initial evaluation pipelines were biased toward short motions, as they accepted the first valid goal returned by the simulator. To probe harder regimes, we also run larger-motion stress tests by sampling 20 valid candidate goals per posture, ranking them by maximum straight-line end-effector displacement (e.g., 0.385 ± 0.006 m in supine), and selecting the top two cases. Follow-up experiments use recall-calibrated thresholds, soft learned bias, and on-policy near-manifold data collected from analytical planner trajectories.

V. Results

A. Analytical Constrained Planning

In a 50-trial-per-posture evaluation, analytical Constraint-RRTConnect solved all 50 supine trials and 45 of 50 seated trials. Mean initial planning time was 0.085 ± 0.032 s (supine) and 0.379 ± 1.655 s (seated). Contact residuals remained small: 0.00205 ± 0.00083 m and 0.0674 ± 0.0241 rad (supine), and 0.00176 ± 0.00069 m and 0.0593 ± 0.0294 rad (seated).

Although global projection success rates were low (0.03 supine, 0.02 seated), the failure distribution indicates that the limiting factor is the narrow feasible manifold under random sampling rather than numerical non-convergence. In the supine posture, 5,000 successful projections required 12,692 attempts, of which 7,623 failed due to state-validity violations (e.g., human joint limits or collisions) and only 69 failed due to numerical non-convergence. In seated, 18,598 attempts yielded 13,414 state-invalid failures and only 184 non-convergence failures. Local projection and bidirectional tree growth were sufficient for high task success.

TABLE I
Learned Checker Performance by Data Source

Data Source	Samples	Accuracy	FPR / Recall
Projected Positives	10,000	0.783	Recall: 0.783
Failed Projections	10,000	0.987	FPR: 0.013
Hard Negatives	20,000	0.829	FPR: 0.171
Uniform Negatives	20,000	1.000	FPR: 0.000

B. Learned Constraint Checking

At the threshold of 0.77, the learned checker achieved 90.47% accuracy, 0.959 AUROC, 68.83% precision, 78.27% recall, and a 7.09% false-positive rate over the 60,000-sample dataset. Performance was nearly identical across postures.

Table I reveals that while the model perfectly classifies uniform negatives (which sit far from the manifold), it struggles with hard negatives near the boundary (17.1% FPR). This boundary uncertainty is exactly what causes false negatives on valid projected states during planning, as the model remains uncertain near the contact manifold boundary.

C. Planner Comparison

Table II summarizes the main 5-trial comparison. Analytical Constraint-RRTConnect is the strongest planner, solving every trial in both postures with initial planning times below 0.08 s. MPPI matches this on seated task success and solves four of five supine trials, but its initial planning time is two orders of magnitude larger. The hard learned-filter variant is less robust: in supine only three of five initial plans succeed, causing MPPI fallback and a large mean planning time; in seated all five initial learned-filter plans succeed, but one execution fails.

The large-motion stress test preserves the same qualitative outcome. Analytical Constraint-RRTConnect solves one of two far-goal supine cases and both seated far-goal cases, with initial planning times of 0.113 ± 0.030 s and 0.071 ± 0.002 s, respectively. MPPI solves the same number of cases but requires 15.377 ± 7.368 s in supine and 7.268 ± 0.239 s in seated. Learned-filter Constraint-RRTConnect matches the analytical planner in seated but fails both harder supine cases.

TABLE II
MAIN PLANNER COMPARISON IN MANIP4CARE+ (5 TRIALS/POSTURE)

Planner	Posture	Task Success	Init Plan Success	Init Plan (s)	Move Dist. (m)	Contact Pos. Error (m)
Constraint-RRTConnect	Supine	$5/5$	$5/5$	0.059 ± 0.011	0.181 ± 0.053	0.00213 ± 0.00033
Constraint-RRTConnect	Seated	$5/5$	$5/5$	0.074 ± 0.059	0.111 ± 0.049	0.00158 ± 0.00036
Learned-filter C-RRTConnect	Supine	$4/5$	$3/5$	11.495 ± 14.774	0.285 ± 0.136	0.00183 ± 0.00048
Learned-filter C-RRTConnect	Seated	$4/5$	$5/5$	0.131 ± 0.138	0.224 ± 0.106	0.00195 ± 0.00044
MPPI	Supine	$4/5$	N/A	6.064 ± 2.086	0.244 ± 0.121	0.00196 ± 0.00047
MPPI	Seated	$5/5$	N/A	8.570 ± 9.668	0.232 ± 0.144	0.00179 ± 0.00060

TABLE III
ON-POLICY RECALL 50-TRIAL TEST

Posture	Task Success	Init Plan Success	Init Plan (s)	Move Dist. (m)
Supine	$18/50$	$50/50$	0.188 ± 0.009	0.715 ± 0.064
Seated	$50/50$	$50/50$	0.054 ± 0.000	0.123 ± 0.000

D. Recall Calibration and On-Policy Learning

The initial evaluation demonstrated that standard classifier thresholds (e.g., maximizing F1 with an FPR cap) yield a recall of only 78.27% on projected positives, starving the RRT of valid nodes. To address this, we implemented planner-recall calibration, lowering the global threshold to 0.58 to achieve 98% recall (with an acceptable 17% FPR, given the analytical safety backstop). We also tested a *soft bias* mode, which samples five projected candidates and expands the one with the highest learned probability without hard rejection, successfully avoiding false negatives in seated tasks.

To better capture the distribution of states the planner actually visits, we generated an *on-policy dataset*. We harvested waypoints from successful analytical plans and added projected seeds near each waypoint to target the exact near-manifold states where false negatives occur. Retraining on this combined dataset with a recall-oriented objective yielded 100% recall on the on-policy validation set.

Table III shows the results of a 50-trial test using this on-policy recall checkpoint. Crucially, *initial plan success reached 50/50 for both postures*, proving that recall calibration entirely eliminates the initial planning bottleneck caused by learned false negatives. However, supine task success remained 18/50. Analysis of the failures reveals that the bottleneck has shifted: failures no longer occur during initial graph search, but during execution-time replanning from invalid intermediate starts. Additionally, probing the network's output probability reveals a strong correlation with actual contact errors (-0.529 for position, -0.544 for orientation), suggesting future models should predict continuous residuals rather than binary validity.

E. Discussion

Compared with MPPI, our analytical Constraint-RRTConnect uses the closed-loop grasp geometry to propose valid coupled states directly, which remains interpretable, exposes projection and steering diagnostics, and provides a clear separation between contact consistency, biomechanical validity, collision checking, and graph search. The learned checker's AUROC shows meaningful structure in the coupled state space, with high performance on uniform and failed-projection negatives. However, filter-mode planning is recall-sensitive. A false negative removes a valid node from the RRT trees and can disconnect the planner. Our on-policy experiments clarify that while increasing the learned checker's recall successfully repairs initial graph connectivity, it is insufficient to guarantee task success in highly constrained postures like supine. Once the initial plan is found, the system must execute and replan under disturbances. If intermediate states drift off the narrow manifold, the learned checker's binary output provides no gradient for repair. This highlights that the challenge is moving from finding a feasible path to maintaining feasibility under execution uncertainty.

VI. CONCLUSION

Manip4Care+ extends Manip4Care with a projection-based constrained planner and a learned constraint checker for assistive limb manipulation. The analytical Constraint-RRTConnect solves supine and seated manipulation tasks with substantially lower initial planning time than the original MPPI baseline while preserving contact consistency and joint-limit safety. The learned-checker results show that hard rejection is sensitive to false negatives; follow-up on-policy recall calibration improve initial planning, while robust execution-time replanning remains the main next step. Overall, Manip4Care+ shows that modular analytical projection and learned feasibility estimation are complementary tools for articulated, human-centered manipulation, with future work focused on continuous residual prediction, physical validation, and robust replanning under execution disturbances.

REFERENCES

[1] Dmitry Berenson, Siddhartha S Srinivasa, Dave Ferguson, and James J Kuffner. Manipulation planning on constraint manifolds. In *2009 IEEE international conference on robotics and automation*, pages 625–632. IEEE, 2009.

[2] Kevin Chow and Charles C Kemp. Robotic repositioning of human limbs via model predictive control. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 473–480. IEEE, 2016.

[3] Henrik Christensen, Nancy Amato, Holly Yanco, Maja Mataric, Howie Choset, Ann Drobnis, Ken Goldberg, Jessy Grizzle, Gregory Hager, John Hollerbach, et al. A roadmap for us robotics–from internet to robotics 2020 edition. *Foundations and Trends® in Robotics*, 8(4):307–424, 2021.

[4] Erwin Coumans and Yunfei Bai. PyBullet, a python module for physics simulation for games, robotics and machine learning. In *Project website*, 2016–2021. URL http://pybullet.org.

[5] Zackory Erickson, Vamsee Gangaram, Ariel Kapusta, C Karen Liu, and Charles C Kemp. Assistive gym: A physics simulation framework for assistive robotics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10169–10176. IEEE, 2020.

[6] Zachary Kingston, Mark Moll, and Lydia E Kavraki. Sampling-based methods for motion planning with constraints. *Annual review of control, robotics, and autonomous systems*, 1(1):159–185, 2018.

[7] Yubin Koh and Ahmed H Qureshi. Manip4care: Robotic manipulation of human limbs for solving assistive tasks. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11435–11442. IEEE, 2025.

[8] James J Kuffner and Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, volume 2, pages 995–1001. IEEE, 2000.

[9] Ahmed H Qureshi, Jiangeng Dong, Austin Choe, and Michael C Yip. Neural manipulation planning on constraint manifolds. *IEEE Robotics and Automation Letters*, 5(4):6089–6096, 2020.