Kinematic Nonlinear Spatio-Temporal Trajectory Warping for Contact-Rich Dexterous Manipulation Demonstrations

Hyojae Park[1], Arjun S. Lakshmipathy[1], Nancy S. Pollard[1,2]

***Abstract*— We present a straightforward but effective method for repurposing existing contact-rich dexterous manipulation demonstrations. Starting from inputs of hand and object trajectories, our method outputs high-quality nonlinear trajectory warps that account for intermediate waypoints, environmental barriers, temporal shifts, and varied start/end configurations. Foundational to our method is the utilization of contact distributions, which we show allows us to reliably compute complex and high-dimensional dexterous hand trajectories following a simple object-centric warp specification pipeline. We evaluate our method across 12 variations sourced from 4 demonstrations in a publicly available dataset of human hand motion data, perform baseline comparisons, and demonstrate generalization of our approach to different manipulators. Results and code will be made available on publication.**

I. INTRODUCTION

Despite significant advances in optical [1], [2], wearable [3], [4], [5], and tele-operative [6], [7], [8], [9] motion capture interfaces and processing algorithms, reconstructing high quality dexterous manipulation demonstrations remains a time-consuming and labor-intensive process. This persistent bottleneck in data acquisition continues to hinder learning-from-demonstration pipelines, driving the demand for methods capable of adapting existing demonstrations to new environmental variations. Furthermore, sampling across variations is vital for hardening sim-to-real transfer strategies against real-world uncertainty. Specifying exactly *how* such variations are generated, as well as what variables to sample over, is a core problem in domain randomization.

Variations can generally be formulated as *nonlinear trajectory warps*, allowing for the modification of spatial constraints (goals, waypoints, obstacles), temporal constraints (timing of targets or segments), and the method of interpolation while attempting to preserve characteristics of the original motion. Manipulation also adds complexity because the hand and object trajectories are implicitly coupled; breaking the coupling can result in serious artifacts such as motion misalignment, penetrations, or loss of contact. Effective nonlinear warping algorithms for dexterous manipulation must account for all these factors simultaneously; however, no simple "off-the-shelf" solver currently exists for this task.

We address this gap by presenting an effective method for nonlinear trajectory warps of long sequence dexterous manipulation demonstrations. Starting from inputs of hand and object trajectories, our method generates high quality coupled warps that respect arbitrary barriers, spatial and temporal waypoints, and start/end configurations. Our main insight is to split the problem into a two-stage object-centric pipeline. We first warp the object's trajectory and then recover the hand pose using contact distributions, defined as per-frame point correspondences between the hand and object meshes. Furthermore, this contact-based approach ensures that our method remains reliable even under large nonlinear spatio-temporal transformations where naive interpolation-based approaches struggle.

We demonstrate our method's effectiveness across 12 variations sourced from 4 demonstrations in a publicly available dataset of human hand motion data, perform baseline comparisons, and demonstrate generalization of our approach to different manipulators. Our method runs in minutes on a CPU and accommodates substantial warps under a consistent strategy with few parameters beyond input specifications. Our method can be used as a *modular drop-in replacement* in dexterous manipulation data augmentation pipelines, which we will facilitate through public code release.

The authors are with the [1]Computer Science Department and the [2]Robotics Institute at Carnegie Mellon University, Pittsburgh, USA.
Corresponding author's contact: hyojae@cmu.edu

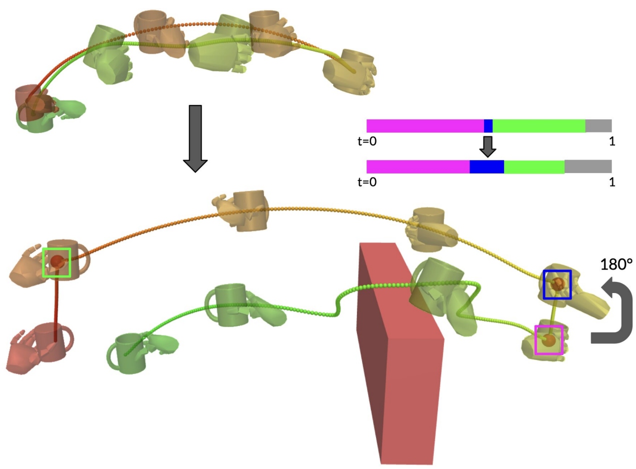


Fig. 1: Our method warps hand and object trajectories from existing contact-rich demonstrations to fit spatial and temporal waypoints, environmental barriers, and new start/end positions. Approximate mappings between the original and warped trajectories are color-coded for visualization.

II. RELATED WORK

We examine three categories of related work: data augmentation, contact-driven grasp and manipulation synthesis, and motion planning.

A. Data Augmentation

Data augmentation techniques enhance a model's robustness and ability to generalize when working with limited original samples. While several techniques focus on image [10], [11] or semantic [12], [13] variation, these techniques do not explicitly alter trajectories; rather, trajectories are either replayed [10] or left to be learned by a model [12].

Trajectory altering augmentation techniques directly output computed warped trajectories from an input specification. Leading the field in adoption, the MimicGen line of work [14], [15] and its variants [16], [17] currently represent one of the most common choices for trajectory-based data augmentation. These techniques, like ours, adopt an object-centric warping strategy; however, the extent of variation is limited to *rigid, linear* transformations ($SE(3)$). More concretely, these works assume that the configuration of the hand relative to the object is fixed under $SE(3)$ equivariance and that new samples are generated by multiplying per-timestep object configurations by a static transformation $\boldsymbol{M} \in SE(3)$. While simple and efficient, these assumptions make it impossible to modify intermediate object paths, adjust timings, or navigate around obstacles. Our method offers generalization to nonlinear warps that accommodate these modifications while maintaining simplicity. Furthermore, our method is specifically designed for dexterous manipulation by optimizing the full hand articulation per timestep rather than applying fixed end-effector transforms [15] and requires only minor additional computational overhead.

B. Contact-Driven Grasp and Manipulation Synthesis

Contacts have played a vital role in grasp and manipulation synthesis techniques. They have frequently been leveraged as a loss term for pose optimization [18], [19], [20], [21], [22], a constraint for training generative models [23], [24], a prior for retargeting human trajectories [25], [26], [27], and a primitive for motion planning [28], [29]. Modern high quality human motion and interaction datasets [1], [2], [30], [31], many of which serve as the backbones and testbeds of manipulation research efforts, are notably prioritizing the inclusion of detailed contact information. Our work expands this literature by using contact to maintain motion coupling during non-trivial spatio-temporal trajectory warps.

C. Motion Planning

Motion planning strives to find a sequence of configurations (kinematics) or states/actions (dynamics) that satisfy terminal or intermediate constraints. These approaches span a range of paradigms including offline collision-free trajectory optimization [32], [33], contact sequence planning [29], [28], [34], [35], and learning-based approaches [36], [37], [38], [39]. These techniques can be broadly grouped into local, global, sampling-based, or gradient-based methods [40], and are typically designed to compute solutions from *scratch* or under *sparse initial conditions*.

While sharing similar input-output profiles, motion planning and trajectory warping employ divergent methodologies: the former constitutes a "bottom-up" constructive approach, while the latter functions as a "top-down" refinement of existing data. Consequently, motion planning often struggles to produce trajectories with nuanced characteristics (e.g., smooth, natural movements) while trajectory warping faces the inverse challenge of preserving these characteristics while adapting to new task goals. Our method lies within the warping paradigm: it adapts the trajectory to new spatio-temporal constraints while attempting to preserve features of the original motion.

III. METHOD

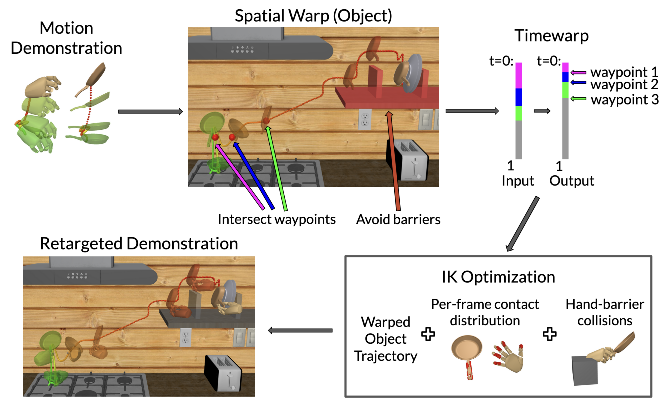


Fig. 2: System overview. Given a motion demonstration, our method first spatially warps the object trajectory to intersect user-specified waypoints and avoid scene barriers. It then applies a time-warp to ensure the trajectory satisfies the time constraints of each waypoint. Finally, an inverse kinematics (IK) optimization recovers the hand motion by tracking the warped object trajectory, subject to hand–barrier collision constraints, using per-frame corresponding contact distributions to yield the complete retargeted demonstration.

Figure 2 illustrates our object-centric warping pipeline. Starting from inputs of polyline-represented trajectories of the object ($\mathcal{T}_o$) and hand ($\mathcal{T}_h$), corresponding hand-object contact distributions per frame, spatio-temporal waypoints, barriers, and terminal configurations p'_{start} and p'_{end}, our pipeline constructs the final warp in four stages. We begin by spatially warping the object trajectory so that it passes through the specified positions and orientations. Next, we further warp this trajectory to prevent object intersections with a set of barrier meshes. Third, we fit a B-spline to the resulting object trajectory to respect temporal constraints. Finally, we utilize hand-object contact distributions to compute the corresponding warped hand trajectory per frame via IK optimization while accounting for barrier constraints.

Each waypoint is defined as $w = (p, r, t, t')$. The positional constraint $p \in \mathbb{R}^3$ specifies a point in world space through which the object must pass. The rotational constraint r defines an imposed rotation applied over the segment between the previous and current waypoint, taking effect when the trajectory reaches p. The temporal constraint is defined by two parameters t and t', where t is the timestep in the original trajectory at which the object must pass through p and t' is the corresponding timestep in the output trajectory. Each barrier is a mesh in world space representing a region that the trajectory must avoid. p'_{start} and p'_{end} alter the starting and ending position of the object, respectively, and are implicitly defined as the first and last waypoints.

A. Spatial Waypoint-Constrained Object Warping

In this first stage, we use p'_{start}, p'_{end} and a tuple of waypoints $\mathcal{W} = w_1, ..., w_k$ to warp the object trajectory. We assume that all trajectories consist of pre-, in-, and post-contact segments. If p'_{start} and p'_{end} are not provided, warping is strictly confined to the in-contact segment of the trajectory, which we assume to be a contiguous. If p'_{start} or p'_{end} are provided, the pre- and post-contact segments are rigidly shifted to the new boundary positions.

Waypoints partition the in-contact sub-trajectory into consecutive segments, which are independently transformed using algorithm 1 to satisfy the positional and rotational constraints at their endpoints. Specifically, algorithm 1 transforms each segment so that the trajectory passes through p_i at t_i and p_{i+1} at t_{i+1}. The algorithm first computes the endpoint displacements $\Delta_s = p_i - \mathcal{T}_o(t_i)$ and $\Delta_e = p_{i+1} - \mathcal{T}_o(t_{i+1})$, then shifts all intermediate points $\mathcal{T}_o|_{(t_i, t_{i+1})}$ by a blended displacement that smoothly transitions from Δ_s to Δ_e.

The interpolation scheme blends two strategies: a quadratic blending Δ_{quad} that preserves the original trajectory shape, and a linear blending Δ_{lin} that produces a more direct path between waypoints. The balance between these two interpolations is controlled by a blend factor $\beta \in [0, 1]$, derived from the maximum endpoint displacement threshold α (default = 0.2 m). When the waypoints are close to the original trajectory ($\beta \approx 0$) quadratic blending dominates, preserving the original motion. When the waypoints are far from the original trajectory ($\beta \approx 1$), linear blending dominates, producing a more direct transition. This adaptive blending produces more natural movements, as opposed to purely using one blending approach.

Lastly, the algorithm imposes a cumulative rotation R_Δ at each waypoint using SLERP interpolation to ensure smooth orientation transitions between segments. Note that t' is not used in this step; temporal retiming is deferred until after barrier constraints are resolved.

B. Barrier-Constrained Object Warping

Next, we warp the trajectory to avoid object intersection with a set of barrier meshes $\{\mathcal{M}_i\}$. We first compute a bounding ball $\mathcal{B}_r$ with radius r for the object. We center $\mathcal{B}_r$ at the origin of the object's frame and compute r by taking the distance to its furthest vertex, which effectively approximates the object's workspace under any relative transform $T \in SE(3)$. For each barrier, $\mathcal{M}_i$, we compute an inflated volume $\mathcal{M}'_i$ that conservatively approximates the Minkowski sum of its convex hull with $\mathcal{B}_r$, i.e., $\textsc{conv}(\mathcal{M}_i) \oplus \mathcal{B}_r$ [41]. Specifically, we uniformly scale $\textsc{conv}(\mathcal{M}_i)$ about its centroid by a factor derived from r, producing a superset of the true Minkowski sum of $\textsc{conv}(\mathcal{M}_i)$. Although this overapproximation may exclude some valid trajectory configurations near barrier surfaces, it is computationally inexpensive and sufficient for resolving object-barrier intersections. Finally, for each trajectory point within $\mathcal{M}'_i$, we project it outward along the ray from the barrier's centroid to the point, stepping incrementally until the point exits the hull surface, resolving all intersections.

C. Temporal Waypoint-Constrained Object Warping

Following spatial constraint fitting, we perform a temporal retiming to realize the $t \mapsto t'$ mapping. We first smooth the trajectory by constructing a chord-length parameterized B-spline [42] for each segment $\mathcal{T}_o\big|_{[t_i,\, t_{i+1}]}$. This induces a normalized timewarp $\tau_i : [0, 1] \to [0, 1]$, which maps uniform time to the spline's chord-length parameterization. Each local parameterization is then scaled and shifted to the temporal interval $[t'_i, t'_{i+1}]$, and the per-segment warps are composed into a global timewarp τ. Then, a global cubic B-spline is fit using τ, ensuring that all positional waypoints and temporal constraints are satisfied while C^2 continuous motion. Finally, we uniformly sample points from this spline to obtain the re-timed contact segment, which is combined with the unchanged pre- and post-contact segments to produce the final object trajectory $\mathcal{T}''_o$.

D. Hand Warping

At this point, the object trajectory is fully resolved: it satisfies all spatial, barrier, and temporal constraints. However, the resulting object trajectory $\mathcal{T}''_o$ and timewarp τ may induce a trajectory that substantially deviates from the original demonstration. Naively recovering the hand trajectory via rigid-body transforms fails under non-uniform timewarps, as we demonstrate in section IV-D. Instead, we recover the hand trajectory by directly optimizing for contact distributions at every timestep in $\mathcal{T}''_o$.

For each timestep t, we retrieve the contact correspondences at t^*, the last input timestep mapped by τ before t. This approach is necessitated by the lack of proven methods for interpolating between arbitrary contact distributions; consequently, we retrieve the distributions directly

---

Algorithm 1 TRANSFORMSEGMENT

Require: Trajectory segment $\mathcal{T}_o|_{[t_i, t_{i+1}]}$, target positions p_i, p_{i+1}, target rotation r_{i+1}, distance threshold α

Ensure: Warped segment $\mathcal{T}'_o|_{[t_i, t_{i+1}]}$

$\mathbf{x}(t),\ R(t) \leftarrow \textsc{Decompose}\big(\mathcal{T}_o(t)\big)$

$\Delta_s \leftarrow p_i - \mathbf{x}(t_i), \quad \Delta_e \leftarrow p_{i+1} - \mathbf{x}(t_{i+1})$

$\beta \leftarrow \textsc{Clip}\left(\dfrac{\max(\|\Delta_s\|,\ \|\Delta_e\|)}{\alpha},\ 0,\ 1\right)^{1/2}$

$R_\Delta \leftarrow \prod_{j=0}^{i+1} r_j$ ▷ Cumulative rotation

for all $t \in [t_i,\ t_{i+1}]$ **do** ▷ Warp each timestep

  $\hat{t} \leftarrow (t - t_i) \,/\, (t_{i+1} - t_i)$

  $\Delta_{\text{quad}} \leftarrow (1 - \hat{t})^2\, \Delta_s + \hat{t}^2\, \Delta_e$

  $\Delta_{\text{lin}} \leftarrow (1 - \hat{t})\, \Delta_s + \hat{t}\, \Delta_e$

  $\mathbf{x}'(t) \leftarrow \mathbf{x}(t) + (1 - \beta)\, \Delta_{\text{quad}} + \beta\, \Delta_{\text{lin}}$

  $R'(t) \leftarrow \textsc{Slerp}\big(I,\ R_\Delta,\ \hat{t}\big)\, R(t)$

end for

return $\mathcal{T}'_o|_{[t_i, t_{i+1}]} \leftarrow \big(\mathbf{x}'(t),\ R'(t)\big)$

---

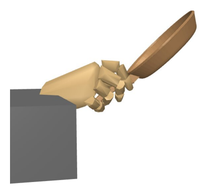
from the original demonstration. We then solve for the hand configuration $\boldsymbol{\theta}$ at each output timestep t using the equation:

$$\boldsymbol{\theta}^*(t) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \ \frac{\sum_{i=1}^{N(t)} \Gamma_{D,i}(\boldsymbol{\theta}, t)}{N(t)} + \sum_{b}^{B} \sum_{p \in \mathcal{P}(\boldsymbol{\theta}, t)} \Gamma_B(p, b) \tag{1}$$

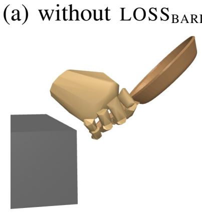
where $N(t)$ is the number of contact correspondences at timestep t, B is the number of barriers, and $\mathcal{P}(\boldsymbol{\theta}, t)$ denotes the hand mesh vertices in world space at timestep t computed via forward kinematics from configuration $\boldsymbol{\theta}$. $\Gamma_{D,i}$ penalizes the L_2 distance between the i-th contact correspondence pair on the hand and object, and Γ_B penalizes hand-barrier intersections. Γ_B is formulated as:

$$\Gamma_B(p, b) = \left(1 - \frac{\text{SDF}(p, b)}{m}\right)^n$$

where m is a margin distance and n controls the penalty steepness. We set $n = 2.0$ and $m = 0.05\,\text{m}$ for all experiments. This formulation encourages a minimum clearance of m from all barriers and increases sharply when the hand penetrates a barrier. We assume barrier meshes are geometric primitives, with complex meshes approximated by their collision geometries for efficient SDF evaluation. The inset figures show an example of a timestep optimized with and without Γ_B. We solve Eq. 1 using the PyTorch Optimization library [43].



(a) without $\text{LOSS}_{\text{BARRIER}}$



(b) with $\text{LOSS}_{\text{BARRIER}}$

Finally, we perform a light smoothing with a Savitzky-Golay filter [44] for the hand's translation and a moving-window quaternion averaging for the joint and wrist rotations. This step reduces jittering across adjacent timesteps without inducing substantial drift from the IK solution.

IV. RESULTS

We generated 12 warping configurations of varying difficulty from 4 baseline demonstrations, which were taken from the GRAB dataset [1]. Contact correspondences were extracted and cleaned using [25]. These configurations were made to resemble common actions in a kitchen. For instance, given a demonstration of pouring a teapot, we warp the motion to pour tea into two separate cups. The background kitchen scene is taken from the RoboCasa dataset [45].

For each configuration, we first warped the object trajectory to satisfy the specified waypoint and barrier constraints. We then optimized the hand trajectory at three optimization thresholds (10 mm, 5 mm, 3.5 mm). At each timestep, we run Adam with a learning rate of 0.01 until the mean contact distance fell below each threshold. Each timestep was initialized using the solution from the previous timestep. All experiments were run on an Apple M2 SoC with 16 GB of memory. All runs are performed exclusively on the CPU.

A. *Qualitative evaluation*

Table I summarizes all warping configurations used. Animations of warping results are available in the supplementary video, which show that our method successfully adapts demonstrations across a variety of constraint configurations.

TABLE I: Evaluation trials and input specifications

Trajectory	# of waypoints	# of barriers	p'_{start} provided	p'_{end} provided
cube_messy	0	7	✓	✓
cube_messy2	4	8	✗	✓
fryingpan_change_tops	1	0	✓	✓
fryingpan_island	3	0	✓	✓
fryingpan_onto_shelf	4	3	✓	✓
mug_pass_wall	2	1	✗	✗
mug_pass_wall_short	3	1	✗	✓
mug_pass_in_sink	0	0	✗	✓
teapot_pour_cup_long	6	0	✗	✓
teapot_pour_cups	6	0	✗	✗
teapot_pour_cups_wall	6	1	✓	✓
teapot_pour_empty	2	0	✗	✓

B. *Quantitative evaluation*

We consider two distance metrics: mean distance between the wrist position and the object's center (D_{WO}), and mean distance between corresponding hand and object contacts (D_C). The first metric measures how well our method retains the demonstration in object space, and the second metric measures the grasping accuracy after warping. Lower values indicate better performance in all tables below.

We also measure the path length ratio (PLR) by dividing the length of the warped path by the original, as well as a timewarp discrepancy metric computed as $D_{\text{timewarp}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(t_i - \frac{i-1}{N-1}\right)^2}$. The latter metric quantifies the deviation of the timewarp from uniform spacing, where a value of zero indicating no temporal distortion and larger values reflecting greater non-uniform timing.

Table II reports D_{WO} across all timesteps. The warped trajectories closely preserve the original hand-object spatial relationship, with D_{WO} differing from the initial by just 5.5 mm at the 10 mm threshold, 2.2 mm at the 5 mm threshold, and just 1 mm at the 3.5 mm threshold. This demonstrates that even under substantial spatio-temporal deformations, the hand's relative positioning is maintained.

TABLE II: D_{WO} across different optimization thresholds. Values are in millimeters (mm).

Trajectory	Init Mean Dist	10 mm	5 mm	3.5 mm
cube_messy	150.7	148.5	150.7	150.8
cube_messy2	150.7	149.0	150.0	149.7
fryingpan_change_tops	164.5	165.7	164.2	165.7
fryingpan_island	164.5	156.6	159.8	162.6
fryingpan_onto_shelf	164.5	165.0	166.1	166.4
mug_pass_wall	145.4	143.3	145.3	145.5
mug_pass_wall_short	145.4	146.7	147.1	147.1
mug_pass_in_sink	145.4	147.2	147.2	147.2
teapot_pour_cup_long	201.8	200.1	202.7	202.6
teapot_pour_cups	201.8	186.9	195.4	201.0
teapot_pour_cups_wall	201.8	196.2	200.1	201.1
teapot_pour_empty	201.8	176.6	195.7	201.4
Average $\lvert\Delta\rvert$	–	**5.5**	**2.2**	**1.0**

Table III reports D_C, the mean per-timestep contact distance averaged over all timesteps. At the 3.5 mm threshold, this metric deviates from the initial value by 1.4 mm on average, and is 1.0 mm lower than the initial value on average. This suggests that our method preserves manipulation quality and can even achieve contact distances tighter than the original demonstration. Furthermore, post-hoc Savitzky–Golay smoothing of the hand shifts D_C by only -0.62 mm on average, indicating that this filter does not meaningfully erode the fidelity of per-frame IK solutions.

TABLE III: D_C across varied optimization thresholds in mm.

Trajectory	Init Cont.	10 mm	5 mm	3.5 mm	Drift
cube_messy	4.6	9.2	4.9	4.3	-0.24
cube_messy2	4.6	9.4	6.0	5.5	0.12
fryingpan_change_tops	5.5	7.9	8.5	5.7	-0.37
fryingpan_island	5.5	9.2	6.9	6.1	-0.70
fryingpan_onto_shelf	5.5	10.0	6.9	6.1	-0.29
mug_pass_wall	8.5	10.6	6.6	6.1	-0.79
mug_pass_wall_short	8.5	8.7	6.5	5.8	-0.72
mug_pass_in_sink	8.5	10.2	6.0	5.4	-0.88
teapot_pour_cup_long	6.4	7.7	6.3	4.9	-0.74
teapot_pour_cups	6.4	9.7	5.7	5.0	-0.95
teapot_pour_cups_wall	6.4	7.4	6.3	5.0	-0.95
teapot_pour_empty	6.4	9.3	6.1	5.0	-0.91
Average $\|\Delta\|$	–	**2.7**	**1.3**	**1.4**	–
Average Δ	–	**2.7**	**0.0**	**-1.0**	–
Average drift	–	–	–	–	**-0.62**

Table IV lists the time taken to run the entire pipeline. On a CPU-only architecture, the most difficult threshold (3.5 mm) takes about 4.6 minutes to complete on average. Table V reports PLR and timewarp discrepancy for all trajectories.

TABLE IV: Runtime across different optimization thresholds. Non-timestep values are in seconds.

Trajectory	# of Timesteps	10 mm	5 mm	3.5 mm
cube_messy	673	74.63	196.93	464.27
cube_messy2	673	96.64	226.91	481.55
fryingpan_change_tops	648	28.39	80.81	206.03
fryingpan_island	648	37.05	118.83	214.40
fryingpan_onto_shelf	648	35.59	110.07	212.03
mug_pass_wall	544	61.79	115.33	174.64
mug_pass_wall_short	544	33.64	87.65	163.01
mug_pass_in_sink	544	42.13	94.98	148.82
teapot_pour_cup_long	740	33.61	112.64	264.46
teapot_pour_cups	740	56.40	156.21	287.05
teapot_pour_cups_wall	740	28.60	113.55	255.16
teapot_pour_empty	740	60.62	161.02	296.12
Average	–	**54.80**	**140.12**	**276.10**

TABLE V: PLRs and timewarp discrepencies across different optimization thresholds.

	PLR Hand				
Trajectory	10 mm	5 mm	3.5 mm	PLR Object	Timewarp
cube_messy	0.999	1.002	0.989	1.005	0.075
cube_messy2	1.358	1.378	1.389	2.024	0.085
fryingpan_change_tops	1.019	0.945	0.933	0.897	0.276
fryingpan_island	1.519	1.522	1.527	2.165	0.072
fryingpan_onto_shelf	1.361	1.314	1.321	1.733	0.075
teapot_pour_empty	1.652	1.649	1.670	3.688	0.073
mug_pass_wall	1.161	1.153	1.151	1.148	0.096
mug_pass_wall_short	1.425	1.433	1.429	1.581	0.094
mug_pass_in_sink	1.084	1.078	1.077	1.121	0.076
teapot_pour_cup_long	1.590	1.560	1.574	2.83	0.139
teapot_pour_cups	1.059	1.111	1.155	1.405	0.117
teapot_pour_cups_wall	1.542	1.565	1.561	2.172	0.117

Figure 4 tests robustness on the teapot_pour_cups_wall scene by corrupting demonstration contacts in two ways: deleting contacts and randomly permuting object-vertex labels among contact pairs. Both remain largely stable through 50%; beyond that, D_C rises as corruption increases and retargeting quality degrades, with the effect more pronounced at higher corruption levels. Permuting is more severe post-50% because it mis-specifies targets, pulling the hand toward incorrect object points; deleting only removes constraints, and since each frame is initialized from the previous pose, the trajectory is more resistant to degradation.

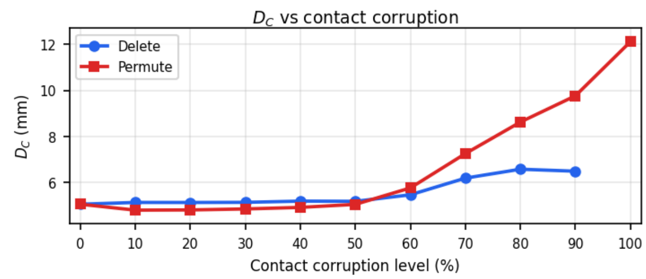
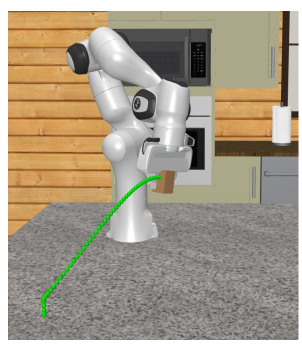


Fig. 4: Contact correspondence robustness test on the teapot_pour_cups_wall scene. Both curves maintain correspondence quality up until 50% (D_c stays near 5.0 mm).

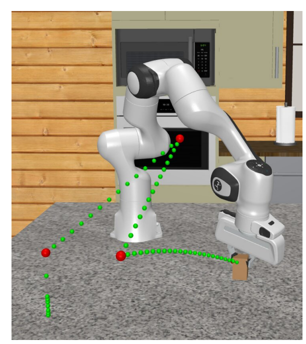
C. Generalization

Although our experiments consider a dexterous human hand and a human motion dataset, our method is not specific to any end-effector. We can easily use our approach on parallel-jaw grippers and multi-fingered robot platforms.

1) Franka arm with parallel grippers: Given a task demonstration, we can trivially compute two contact points (one for each jaw where it intersects with the object).



(a) input trajectory

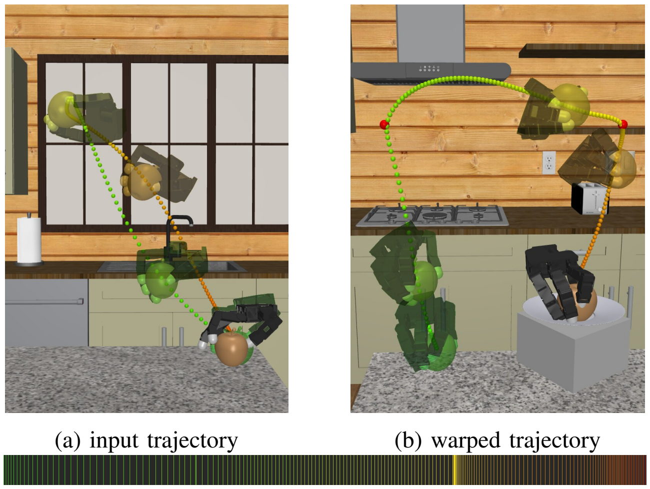


(b) warped trajectory

Fig. 5: An example warping using the Franka arm. The hand must move the box through a series of waypoints.

Figure 5 shows that this is enough to successfully warp a Franka gripper demonstration. In this example, we begin with a demonstration in which the gripper lifts and moves a small box. The trajectory is warped to move the box through a series of waypoints and then place it on the table.

2) Allegro hand: Figure 6 shows that the method generalizes to other dexterous hands such as Allegro. The initial trajectory simply picks up the apple, and our method warps it to move the apple through waypoints and place it on an elevated plate. The resulting timewarp is visualized in Figure 6(c), showing the non-uniform temporal redistribution induced by the warping. Despite significant warping, D_{WO} changes only marginally (0.133 m $\rightarrow$ 0.137 m), suggesting that our method preserves the hand–object spatial relationship.



(a) input trajectory (b) warped trajectory

(c) timewarp

Fig. 6: An example Allegro hand warping where (a) the input apple handoff trajectory is (b) altered to pass through waypoints (red) and terminate on an elevated plate. (c) The timewarp visualization illustrates where frames of the original trajectory fall in the warped sequence, with black portions indicating segments which must be interpolated.

D. Baseline comparison

Previous demonstration augmentation methods rigidly adapt hand trajectories using a constant $SE(3)$ transformation applied to all timesteps [14], [15]. To construct a comparable baseline for our setting, we extend this approach by computing rigid-body transforms at the input timesteps that the timewarp maps directly to output timesteps.

However, because we only know the true hand-object relationship at input timesteps, the only "correct" hand configurations are in τ's codomain. We therefore must interpolate poses at timesteps where no true configuration exists. To do so, we treat these known configurations as control points, fit a B-spline parameterized by the timewarp, and sample it at equal increments to produce the baseline hand trajectory.

This approach introduces a fundamental problem. While the hand-object relationship is preserved at the control point timesteps, the interpolated hand positions between them are independent of the object's position. This issue is especially exacerbated when the timewarp is non-uniform or when the spatial deformation is large. In contrast, our method explicitly optimizes the hand pose at every output timestep to minimize contact point distances, ensuring the hand-object relationship is preserved throughout the entire trajectory rather than just at samples from the original trajectory.

Figure 7 shows an example failure case of the naive version. The graph depicts the z-position of the warped object trajectory, as well as the naive and contact-based hand warped trajectories. The yellow region marks the contact region, and the purple lines represent the timesteps mapped from the input trajectory to the output trajectory by τ.

Between timesteps 529 and 750, there are few purple lines, indicating significant temporal stretching from the input trajectory in this region. Here, the naive method must interpolate the hand trajectory, and regardless of the interpolation scheme used (cubic in this case), the resulting hand motion diverges from the object's path.

At timesteps 528 and 751 where we have the ground truth hand-object relationship, the naive method's hand maintains a proper grasp of the object. However, at interpolated points, the method fails to extrapolate the correct hand transformation, leading to intersections (orange) or failure to maintain contact (gray). Our contact-based method directly combats interpolation errors, leading to a more accurate warping.

V. DISCUSSION

A. Advantages and Disadvantages

Our contact-based method is especially beneficial when the warping configuration introduces substantial spatio-temporal deformation (e.g., far-away waypoints or highly non-linear timewarp). Our approach is also particularly well-suited to interactions where precise hand-object contact is critical, such as grasping, pouring, or tool use. Additionally, many existing methods do not have built-in mechanisms for resolving hand-barrier collisions. Our method naturally handles this through the barrier loss term during per-timestep optimization. This implies our approach is especially valuable in cluttered scenes with multiple obstacles. However, we do note that mild warpings, especially those without temporal shifts or barriers, may be serviced by faster techniques [14].

B. Limitations

Our method currently has a number of limitations. First, we assume that all objects are rigid bodies. We do not support articulated or deformable objects, which would require significant computation to track the motion of individual components during both object and hand warping.

We assume that all barriers are convex. Concave barriers may therefore be too conservatively approximated.

Additionally, our method is evaluated purely in simulation and in isolation rather than as a component in a larger learning pipeline. Although our goal in this work is to focus on the core warp operation, which we view as an upstream processing problem, it is not clear how effective these warps are in sim2real deployment pipelines. A comprehensive investigation into large-scale domain randomization using our approach is an exciting direction for future work.

Finally, our method does not interpolate contact correspondence between timesteps, instead reusing the last available set. This restricts the types of tasks that can be effectively warped. Warping demonstrations involving rapid finger movements or fine-grained intermediate contact changes, for example, may fail as the evolving contact distribution is not captured between available reference timesteps. Formulating simple input specifications that build upon contact interpolation works [46] may address this limitation.

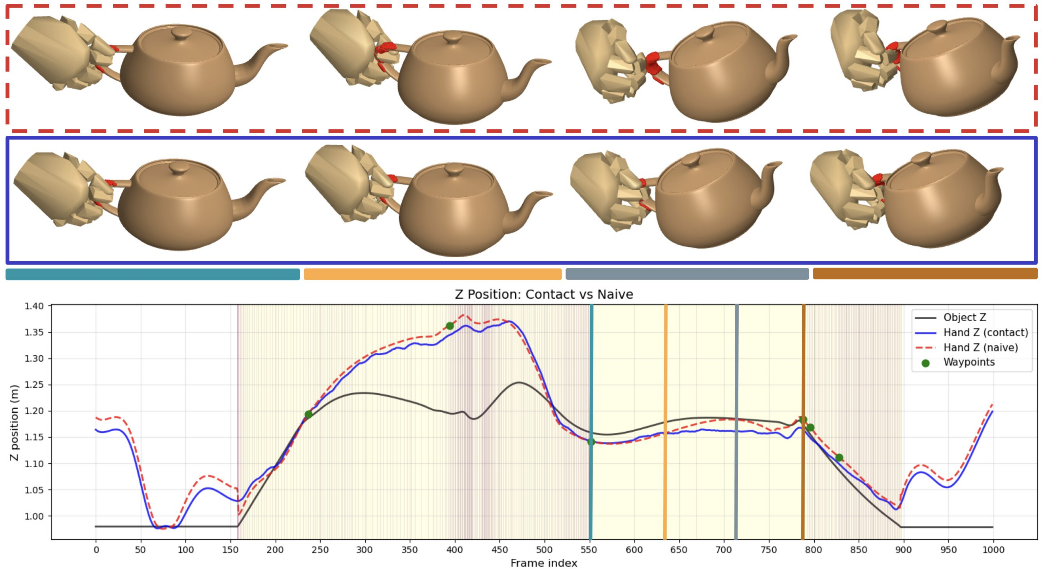
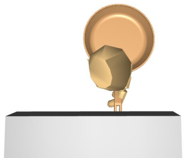


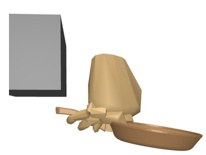
Fig. 7: Comparison of the baseline and our hand warping methods. The yellow region marks the in-contact segment. Light purple lines show the timewarp mapping from input to output timesteps. Four timesteps are visualized for both warping methods (cyan, orange, gray, brown). The baseline method degrades at the orange and gray timesteps, where ground-truth transformation is not available. Our contact-based method continues to maintain a proper grasp across all timesteps.

C. Failure cases

While generally effective, our method does exhibit failure cases which we discuss below.



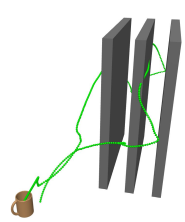
(a) top view



(b) side view

1) Hand-barrier collisions: The warped object trajectory may navigate past barriers such that the hand cannot maintain a proper grasp of the object without intersecting a barrier. The fryingpan inset shows such an instance, where the object barely clears the corner of a box, leaving insufficient space for the hand. These cases can be mitigated by increasing the object's bounding ball to provide additional clearance.

2) Narrow gaps: Barrier-object collisions are resolved iteratively, so pushing trajectory points out of one barrier may displace them into an adjacent one. The right inset shows such a case, where barriers form narrow gaps, causing the barrier resolution to oscillate trajectory points without converging to a valid trajectory.



VI. CONCLUSION

We have presented a straightforward but effective method for nonlinear spatio-temporal warping of contact-rich dexterous manipulations. Our framework effectively reconciles spatio-temporal waypoints and environmental barriers with varying start-to-end configurations. Using hand-object contact distributions, we successfully decomposed the task into an object-centric trajectory pipeline followed by contact-preserving hand warping via per-timestep IK optimization. Our evaluation shows that the warped trajectories preserve the hand-object relationship to within 1 mm on average, while being computationally tractable even on a CPU, and that our method generalizes across different end-effectors, including a parallel-jaw gripper and the Allegro hand. Our method is well poised to serve as a modular, practical upstream component in demonstration augmentation pipelines, enabling efficient generation of spatially and temporally diverse training data. We hope that the simplicity of our approach, together with a public code release, will encourage its adoption within data augmentation pipelines.

ACKNOWLEDGMENTS

This research was partially supported by the National Institute of Biomedical Imaging and Bioengineering of the National Institutes of Health under Award Number R01EB036842, and by the RAI Institute.

REFERENCES

[1] O. Taheri, N. Ghorbani, M. J. Black, and D. Tzionas, "Grab: A dataset of whole-body human grasping of objects," in *European conference on computer vision*. Springer, 2020, pp. 581–600.

[2] Z. Fan, O. Taheri, D. Tzionas, M. Kocabas, M. Kaufmann, M. J. Black, and O. Hilliges, "ARCTIC: A dataset for dexterous bimanual hand-object manipulation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 943–12 954.

[3] Y. Mao, U. Yoo, Y. Yao, S. N. Syed, L. Bondi, J. Francis, J. Oh, and J. Ichnowski, "Visuo-acoustic hand pose and contact estimation," *arXiv preprint arXiv:2508.00852*.

[4] I. Guzey, H. Qi, J. Urain, C. Wang, J. Yin, K. Bodduluri, M. Lambeta, L. Pinto, A. Rai *et al.*, "Dexterity from smart lenses: Multi-fingered robot manipulation with in-the-wild human demonstrations," *arXiv preprint arXiv:2511.16661*, 2025.

[5] J. Yin, H. Qi, Y. Wi, S. Kundu, M. Lambeta, W. Yang, C. Wang, T. Wu, J. Malik, and T. Hellebrekers, "Osmo: Open-source tactile glove for human-to-robot skill transfer," *arXiv preprint arXiv:2512.08920*, 2025.

[6] A. Handa, K. V. Wyk, W. Yang, J. Liang, Y. W. Chao, Q. Wan, S. Birchfield, N. D. Ratliff, and D. Fox, "Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 9164–9170.

[7] A. Sivakumar, K. Shaw, and D. Pathak, "Robotic telekinesis: Learning a robotic hand imitator by watching humans on youtube," in *Robotics: Science and Systems*, 2022.

[8] P. Naughton, J. Cui, K. Patel, and S. Iba, "Respilot: Teleoperated finger gaiting via gaussian process residual learning," *arXiv preprint arXiv:2409.09140*, 2024.

[9] Z. H. Yin, C. Wang, L. Pineda, K. Bodduluri, T. Wu, P. Abbeel, and M. Mukadam, "Geometric retargeting: A principled, ultrafast neural hand retargeting algorithm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2025, pp. 17 376–17 382.

[10] S. Jin, L. Wang, B. Temming, and F. T. Pokorny, "Physically-based lighting generation for robotic manipulation," *arXiv preprint arXiv:2508.01442*, 2025.

[11] I. Kostrikov, D. Yarats, and R. Fergus, "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels," *arXiv preprint arXiv:2004.13649*, 2020.

[12] Z. Chen, S. Kiami, A. Gupta, and V. Kumar, "Genaug: Retargeting behaviors to unseen situations via generative augmentation," *arXiv preprint arXiv:2302.06671*, 2023.

[13] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar, "Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking," in *IEEE International Conference on Robotics and Automation*. IEEE, 2024, pp. 4788–4795.

[14] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox, "Mimicgen: A data generation system for scalable robot learning using human demonstrations," *arXiv preprint arXiv:2310.17596*, 2023.

[15] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. J. Fan, and Y. Zhu, "Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 16 923–16 930.

[16] Z. Xue, S. Deng, Z. Chen, Y. Wang, Z. Yuan, and H. Xu, "Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning," *arXiv preprint arXiv:2502.16932*, 2025.

[17] H. Oh, M. Murooka, T. Motoda, R. Nakajo, and Y. Domae, "Self-augmented robot trajectory: Efficient imitation learning via safe self-augmentation with demonstrator-annotated precision," *arXiv preprint arXiv:2509.09893*, 2025.

[18] A. S. Lakshmipathy, N. Feng, Y. X. Lee, M. Mahler, and N. S. Pollard, "Contact edit: Artist tools for intuitive modeling of hand-object interactions," *ACM Transactions on Graphics*, vol. 42, no. 4, 2023.

[19] D. Turpin, L. Wang, E. Heiden, Y.-C. Chen, M. Macklin, S. Tsogkas, S. Dickinson, and A. Garg, "Grasp'd: Differentiable contact-rich grasp synthesis for multi-fingered hands," in *European Conference on Computer Vision*. Springer, 2022, pp. 201–221.

[20] S. Brahmbhatt, A. Handa, J. Hays, and D. Fox, "Contactgrasp: Functional multi-finger grasp synthesis from contact," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2386–2393.

[21] Y. Ye and C. K. Liu, "Synthesis of detailed hand manipulations using contact sampling," *ACM Transactions on Graphics (ToG)*, vol. 31, no. 4, pp. 1–10, 2012.

[22] C. Hazard, N. Pollard, and S. Coros, "Automated design of robotic hands for in-hand manipulation tasks," *International Journal of Humanoid Robotics*, vol. 17, no. 01, p. 1950029, 2020.

[23] S. Christen, M. Kocabas, E. Aksan, J. Hwangbo, J. Song, and O. Hilliges, "D-grasp: Physically plausible dynamic grasp synthesis for hand-object interactions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 577–20 586.

[24] Y. Wu, J. Wang, Y. Zhang, S. Zhang, O. Hilliges, F. Yu, and S. Tang, "Saga: Stochastic whole-body grasping with contact," in *European Conference on Computer Vision*. Springer, 2022, pp. 257–274.

[25] A. S. Lakshmipathy, J. K. Hodgins, and N. S. Pollard, "Kinematic motion retargeting for contact-rich anthropomorphic manipulations," *ACM Transactions on Graphics*, vol. 44, no. 2, pp. 1–20, 2025.

[26] A. Mandi, Y. Hou, D. Fox, Y. Narang, A. Mandlekar, and S. Song, "Dexmachina: Functional retargeting for bimanual dexterous manipulation," *arXiv preprint arXiv:2505.24853*, 2025.

[27] C. Pan, C. Wang, H. Qi, Z. Liu, H. Bharadhwaj, A. Sharma, T. Wu, G. Shi, J. Malik, and F. Hogan, "Spider: Scalable physics-informed dexterous retargeting," *arXiv preprint arXiv:2511.09484*, 2025.

[28] 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 Transactions on robotics*, vol. 39, no. 6, pp. 4691–4711, 2023.

[29] X. Cheng, E. Huang, Y. Hou, and M. T. Mason, "Contact mode guided motion planning for quasidynamic dexterous manipulation in 3d," in *IEEE International Conference on Robotics and Automation*, 2022, pp. 2730–2736.

[30] J. DelPreto, C. Liu, Y. Luo, M. Foshey, Y. Li, A. Torralba, W. Matusik, and D. Rus, "Actionsense: A multimodal dataset and recording framework for human activities using wearable sensors in a kitchen environment," *Advances in Neural Information Processing Systems*, vol. 35, pp. 13 800–13 813, 2022.

[31] Y. R. Song, J. Li, R. Fu, D. Murphy, K. Zhou, R. Shiv, Y. Li, H. Xiong, C. E. Owens, Y. Du *et al.*, "Opentouch: Bringing full-hand touch to real-world interaction," *arXiv preprint arXiv:2512.16842*, 2025.

[32] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 489–494.

[33] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *IEEE international conference on robotics and automation*, 2011, pp. 4569–4574.

[34] H. T. Suh, T. Pang, T. Zhao, and R. Tedrake, "Dexterous contact-rich manipulation via the contact trust region," *The International Journal of Robotics Research*, p. 02783649251398875, 2025.

[35] S. Le Cleac'h, T. A. Howell, S. Yang, C. Y. Lee, J. Zhang, A. Bishop, M. Schwager, and Z. Manchester, "Fast contact-implicit model predictive control," *IEEE Transactions on Robotics*, vol. 40, pp. 1617–1629, 2024.

[36] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, "Motion planning diffusion: Learning and planning of robot motions with diffusion models," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 1916–1923.

[37] M. Seo, Y. Cho, Y. Sung, P. Stone, Y. Zhu, and B. Kim, "Presto: Fast motion planning using diffusion models based on key-configuration environment representation," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 10 861–10 867.

[38] M. Dalal, J. Yang, R. Mendonca, Y. Khaky, R. Salakhutdinov, and D. Pathak, "Neural mp: A generalist neural motion planner," *arXiv preprint arXiv:2409.05864*, 2024.

[39] A. H. Qureshi, A. Simeonov, M. J. Bency, and M. C. Yip, "Motion planning networks," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2118–2124.

[40] L. Zhang, K. Cai, Z. Sun, Z. Bing, C. Wang, L. Figueredo, S. Haddadin, and A. Knoll, "Motion planning for robotics: A review for sampling-based planners," *Biomimetic Intelligence and Robotics*, vol. 5, no. 1, p. 100207, 2025.

[41] M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars, *Computational geometry: algorithms and applications*. Springer, 2008.

[42] L. Piegl and W. Tiller, *The NURBS book*. Springer Science & Business Media, 2012.

[43] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.

[44] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[45] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, "Robocasa: Large-scale simulation of everyday tasks for generalist robots," *arXiv preprint arXiv:2406.02523*, 2024.

[46] A. Lakshmipathy, D. Bauer, and N. S. Pollard, "Contact tracing: A low cost reconstruction framework for surface contact interpolation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2021, pp. 5165–5172.