DEXFORGE: HIGH-FIDELITY PHYSICS-INFORMED DEXTEROUS RETARGETING

Anonymous authors
Paper under double-blind review

ABSTRACT

Human demonstrations offer rich examples of precise dexterous manipulation and a promising source of robot training data. However, high-fidelity reproduction of demonstrated motions and hand–object interactions across robot embodiments remains challenging under physical constraints. We present DexForge, a differentiable physics-grounded framework for converting human video demonstrations into high-fidelity robot trajectories. We reconstruct spherical-Gaussian object models and hand–object motion from visual observations, then build a differentiable simulator combining efficient Gaussian collision detection with existing differentiable dynamics. Based on this simulator, DexForge combines contact-aware kinematic retargeting with force-aware dynamics retargeting: robot-adapted stable contacts guide kinematic reference construction and subsequent gradient-based control refinement for precise physical motion reproduction. Experiments on 130 DexYCB and HOT3D demonstrations across seven dexterous hands show success-rate gains of approximately 35–53 percentage points over the baseline, with object position and orientation tracking errors on successful trajectories reduced by approximately 34–67% and 71–78%, respectively. Further experiments demonstrate open-loop transfer to MuJoCo and real-robot execution. Our project page is available at `https://dexforge-42306.github.io/project-page/`.

1 INTRODUCTION

Enabling robots with different embodiments to reproduce precise human manipulation is an important step toward transferring human expertise to robots (Xu et al., 2025a; Zhang et al., 2026). High-fidelity reproduction requires more than task completion; the demonstrated motions and hand–object interactions must also be preserved. For example, establishing a stable grasp is only the starting point; precisely controlling the object's subsequent movement and orientation captures finer aspects of manipulation. Human manipulation videos contain rich examples of these behaviors, yet translating them into high-fidelity trajectories executable by different dexterous hands remains challenging.

Recent advances in learning from human demonstrations and retargeting are narrowing this gap. Hand–object reconstruction recovers explicit 3D geometry and motion references from videos (Wang et al., 2025; Shi et al., 2026), while kinematic retargeting adapts these references to robots, increasingly incorporating contact preservation (Lakshmipathy et al., 2025) and collision constraints (Tian et al., 2026) to improve geometric consistency. Building on these references, physics-based retargeting uses policy learning (Li et al., 2025) or trajectory optimization (Pan et al., 2026) to find controls that drive object motion in simulation, with contact guidance helping preserve demonstrated interactions (Mandi et al., 2025). Meanwhile, advances in differentiable collision detection (Beker et al., 2025; Wang et al., 2026) and dynamics modeling (Paulus et al., 2026) enable control optimization through simulation gradients, providing another route to precise trajectory refinement (Liu et al., 2024). Together, these developments extend retargeting beyond geometric hand-motion matching toward physically grounded manipulation reproduction.

Despite these advances, high-fidelity reproduction across different robot hands remains difficult. Several problems need to be solved in sequence. First, the reconstructed object geometry must be accurate enough for collision and contact reasoning. Second, contacts from the human demonstration may not be reachable or stable for a robot with a different hand shape. Third, even a penetration-free robot trajectory with geometrically appropriate contacts may not be executable under contact dynamics and actuation constraints. Physics-based optimization also involves a trade-off. Sampling-based

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

methods can handle complex contact changes, but obtaining very accurate trajectories may require many samples. Gradient-based optimization can refine trajectories more directly, but it is sensitive to initialization and local minima. This motivates us to use robot-adapted contacts to provide a good kinematic reference and continued guidance during gradient-based dynamics optimization.

We introduce DexForge, a differentiable physics-grounded framework for high-fidelity retargeting from human video demonstrations. Our key insight is that human contacts should not be copied directly to a robot with a different hand. Instead, we first adapt these contacts to the target robot, and then use the adapted contacts to guide both kinematic retargeting and dynamic control optimization. To support this process, we reconstruct spherical Gaussian object models that provide both visual appearance for pose estimation and geometry for collision checking. We then extract stable contacts from the human demonstration and refine them together with robot configurations so that they are reachable, collision-free, and stable for the target hand. These contacts are used to build a robot kinematic reference. Finally, we combine differentiable Gaussian collision with ComFree dynamics and optimize robot controls through simulation gradients to reproduce the demonstrated motion more accurately. We further use system identification to reduce the mismatch between simulation and the target system. Our contributions are:

- We propose DexForge, a complete framework that combines visual reconstruction, contact-aware kinematic retargeting, and force-aware dynamics optimization to transform human video demonstrations into high-fidelity, executable demonstrations across robot embodiments.
- We introduce robot-adapted stable-contact guidance through joint anchor and robot-configuration optimization, supporting both kinematic reference construction and dynamic refinement.
- We jointly train spherical Gaussians for appearance and geometry and develop efficient differentiable collision detection for them, supporting integration with existing simulators.
- We evaluate DexForge on 130 DexYCB and HOT3D demonstrations across seven hands, outperforming sampling-based baselines in retargeting success and object-tracking accuracy, and demonstrate open-loop MuJoCo transfer and real-robot execution.

2 Related Work

Dexterous Manipulation from Human Demonstrations. Human demonstrations can be collected through portable motion-capture systems (Wang et al., 2024) and wearable interfaces (Xu et al., 2025a; Tao et al., 2025). Egocentric human videos also provide scalable supervision for dexterous policy learning (Zhang et al., 2026; Zheng et al., 2026). Explicit scene and motion references recovered from videos enable robot trajectory generation (Mu et al., 2026; Paliwal et al., 2026) or guide policy learning in simulation (Lum et al., 2025). DexForge reconstructs object models from separate capture sequences and recovers hand–object motion from demonstration videos, then converts these references into *high-fidelity* robot trajectories through *contact adaptation and physical optimization*.

Dexterous Motion Retargeting. Kinematic retargeting maps human motion to robot configurations through geometric correspondence objectives (Yin et al., 2025; Xin et al., 2026). Interaction-aware methods further preserve contact regions or local hand–object geometry (Lakshmipathy et al., 2025; Wu et al., 2026), while constrained formulations incorporate collision avoidance (Tian et al., 2026). Such interaction structure can also be encoded through interaction meshes for whole-body retargeting (Yang et al., 2026) and dexterous manipulation (Feng et al., 2026). The kinematic stage of C2Dex (Ren et al., 2026) uses stable object-side contacts to guide reference construction. For grasp configurations, GenHand (Qi et al., 2026) refines human-derived contact anchors through force-closure optimization, while GraspQP (Zurbrügg et al., 2025) develops a differentiable force-closure energy for grasp synthesis. DexForge *jointly* optimizes robot keyframe *configurations and anchors* shared within stable-contact intervals, considering *reachability and force closure*.

Physics-based retargeting accounts for contact dynamics, with representative approaches using reference-guided reinforcement learning (Li et al., 2025; Liu et al., 2025; Mandi et al., 2025; Xu et al., 2025b; Ren et al., 2026; Feng et al., 2026) or sampling-based trajectory optimization (Yang et al., 2025; Pan et al., 2026; Paliwal et al., 2026). SPIDER (Pan et al., 2026) progressively relaxes virtual contact guidance during optimization, while Do as I Do (Paliwal et al., 2026) adapts sampling-based retargeting to noisy reconstructed references. Gradient-based transfer is also explored by QuasiSim (Liu et al., 2024), which progressively tightens physical constraints through a curriculum of parameterized quasi-physical simulators. DexForge uses *robot-adapted stable contacts* to construct collision-constrained kinematic references for initialization and retains the same refined anchors as interaction targets during *gradient-based* dynamics optimization.

Differentiable Simulation and Physical Scene Modeling. Differentiable collision detection provides geometric derivatives for optimization (Tracy et al., 2023; Montaut et al., 2023; Beker et al., 2026), while contact models and simulators advance differentiable dynamics (Jin, 2024; Ye et al., 2026; Paulus et al., 2026). Gaussian representations (Kerbl et al., 2023) also support physical modeling: ContactGaussian-WM (Wang et al., 2026) unifies appearance and collision geometry with spherical Gaussians but optimizes geometry and appearance separately, whereas D-REX (Lou et al., 2026) combines Gaussian rendering and mesh-based collisions with differentiable mass identification and grasping policy learning. DexForge *jointly* optimizes spherical Gaussians to better capture appearance and collision geometry, providing an *efficient differentiable collision-query interface* directly on these primitives. Integrating it with ComFree dynamics (Borse, 2026) yields ComFree-GS, supporting contact-aware kinematic retargeting and gradient-based physical retargeting.

3 METHOD

Our goal is to transform human video demonstrations into high-fidelity, physically executable robot rollouts. We formulate the problem in Section 3.1 and address it by progressively establishing the physical structure required for robot execution, as illustrated in Figure 1. We first reconstruct the geometry of the hand and object and recover their pose trajectories from visual observations (Section 3.2). We then extract stable human contacts, adapt them to the target robot, and generate a robot-adapted kinematic reference that satisfies collision constraints (Section 3.3). Finally, we realize this reference by optimizing controls through a differentiable simulator and calibrating the dynamics model against observations from the target system (Section 3.4).

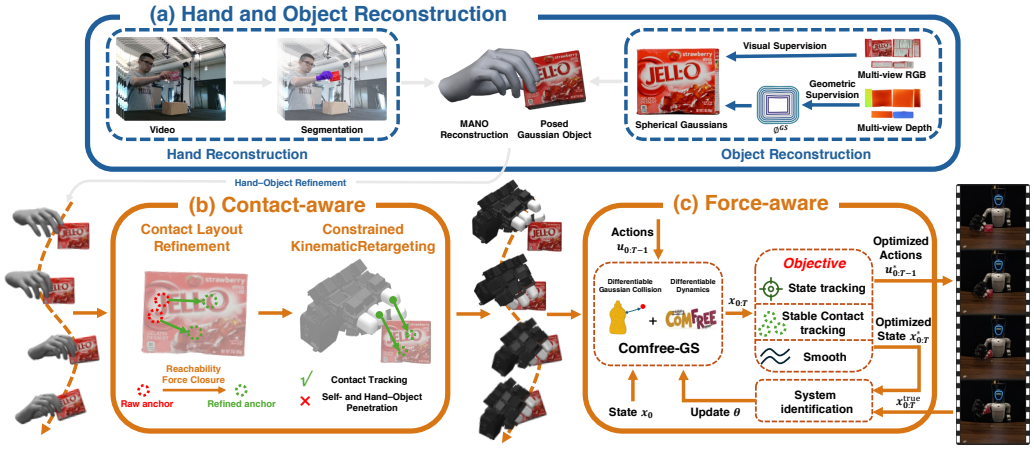


Figure 1: Overview of DexForge. (a) Hand and object reconstruction builds spherical Gaussian object models under RGB-D supervision and recovers hand–object motion from demonstration videos. (b) Contact-aware kinematic retargeting refines demonstrated contacts for robot reachability and contact stability, generating penetration-free references that track them. (c) Force-aware dynamics retargeting optimizes controls through simulation gradients in ComFree-GS, which combines efficient differentiable Gaussian collision detection with ComFree dynamics. System identification further reduces target-system mismatch to support high-fidelity physical reproduction.

3.1 PROBLEM FORMULATION

Given a hand–object interaction video V, we recover geometric and motion references for subsequent retargeting. We construct a Gaussian object model O from a separate multi-view RGB-D video and use it to recover the object pose trajectory $q_{0:T}^{\mathrm{ref,object}}$ in V. MANO (Romero et al., 2022) hand reconstruction from V provides the keypoint trajectory $\mathbf{P}_{0:T}^{\mathrm{mano}}$. Here, $\mathbf{P}_t^{\mathrm{mano}} \in \mathbb{R}^{N_k \times 3}$ contains the positions of N_k hand keypoints, and $q_t^{\mathrm{ref,object}} \in \mathrm{SE}(3)$ denotes the object pose. Both trajectories are expressed in a common coordinate frame to support subsequent retargeting.

Given these geometric and motion references, we formulate kinematic and dynamics retargeting as two sequential constrained optimization problems with complementary roles. **Kinematic retargeting** seeks a robot reference trajectory $q_{0:T}^{\mathrm{ref,robot}}$ that matches the demonstrated motion through keypoint fitting, subject to configuration and collision constraints. Specifically, we use the corre-

spondence set $\mathcal{C}$ to fit robot keypoint positions $\mathbf{P}_i^{\text{robot}}(q_t^{\text{ref,robot}})$, computed by forward kinematics, to their MANO counterparts $\mathbf{P}_{t,j}^{\text{mano}}$, where $(i,j) \in \mathcal{C}$. The robot and object reference trajectories define $q_t^{\text{ref}} = (q_t^{\text{ref,robot}}, q_t^{\text{ref,object}})$ and the reference state $x_t^{\text{ref}} = (q_t^{\text{ref}}, \dot{q}_t^{\text{ref}})$. **Dynamics retargeting** then optimizes robot controls $\mathbf{u}_{0:T-1}$ to track the reference state trajectory $x_{0:T}^{\text{ref}}$, starting from $x_0 = x_0^{\text{ref}}$. The resulting trajectory must satisfy the simulated dynamics $x_{t+1} = f(x_t, \mathbf{u}_t, t)$ and the admissible state and control constraints $x_t \in \mathcal{X}$ and $\mathbf{u}_t \in \mathcal{U}$. Here, $x_t = (q_t, \dot{q}_t)$ denotes the physical state, with $q_t = (q_t^{\text{robot}}, q_t^{\text{object}})$. To reduce mismatch with the target system, we calibrate the simulator's physical parameters θ through system identification, minimizing the discrepancy between simulated trajectories $x(\theta)$ and observed trajectories x^{target} under identical controls.

3.2 Hand and Object Reconstruction

In this section, we reconstruct high-fidelity object assets from multi-view RGB-D videos and recover object pose trajectories and MANO hand motion from hand–object interaction videos. The reconstructed assets must combine accurate geometry for physical simulation in subsequent retargeting with faithful appearance for pose estimation. ContactGaussian-WM (Wang et al., 2026) demonstrates how spherical Gaussians can unify visual appearance and collision geometry, avoiding the need to maintain and align separate models. However, its image-based initialization does not directly constrain interior distances, potentially yielding unreliable penetration-depth estimates. Its subsequent color-fitting stage holds this potentially biased geometry fixed, which can also limit appearance reconstruction accuracy. We therefore introduce depth-derived distance supervision to constrain geometry while allowing geometry and color to be optimized jointly. Specifically, we define the Gaussian collision distance field at a query point $p_i \in \mathbb{R}^3$ as:

$$\phi^{\text{GS}}(p_i) = \min_j \left(\|p_i - \boldsymbol{\mu}_j\|_2 - r_j \right), \quad 1 \le j \le N_{\text{gs}}. \tag{1}$$

Here, N_{gs} is the number of spherical Gaussian primitives, $\boldsymbol{\mu}_j$ and σ_j are the center and scale of the j-th Gaussian, and $r_j = 3\sigma_j$ is its empirically chosen collision radius. We construct a reference signed distance field from multi-view depth maps and use it to supervise $\phi^{\text{GS}}(p_i)$, combining the distance-matching loss with the standard 3DGS objective to jointly optimize geometry and color.

We use GS-Pose (Cai et al., 2024) with the reconstructed Gaussian object model O to recover its pose trajectory $q_{0:T}^{\text{ref,object}}$ in the interaction video V. Keeping the model's geometry and appearance fixed, pose refinement optimizes the object's rotation and translation through differentiable rendering to align its rendered appearance with the observations. We then also recover MANO hand motion from V using HaMeR (Pavlakos et al., 2023) predictions and camera calibration, obtaining the keypoint trajectory $\mathbf{P}_{0:T}^{\text{mano}}$ in the same coordinate frame as the object poses.

To support collision simulation for retargeting, we develop an efficient differentiable interface for the reconstructed Gaussians, first extending Eq. (1) to the general Gaussian–Gaussian form:

$$\phi^{\text{GS}}(\mathcal{A}, \mathcal{B}) = \min_{i,j} \left(\left\| \boldsymbol{\mu}_i^{\mathcal{A}} - \boldsymbol{\mu}_j^{\mathcal{B}} \right\|_2 - r_i^{\mathcal{A}} - r_j^{\mathcal{B}} \right), \quad 1 \le i \le N^{\mathcal{A}},\ 1 \le j \le N^{\mathcal{B}}. \tag{2}$$

Here, $\mathcal{A}$ and $\mathcal{B}$ contain $N^{\mathcal{A}}$ and $N^{\mathcal{B}}$ spherical Gaussians, respectively; $\boldsymbol{\mu}_i^{\mathcal{A}}$ and $r_i^{\mathcal{A}}$ denote the center and collision radius of Gaussian i in $\mathcal{A}$. All centers share a common coordinate frame. We implement efficient differentiable collision backends separately in JAX and Warp. Gaussian training, motion reconstruction and collision detection are detailed in Appendices A, B and C, respectively.

3.3 Contact-Aware Kinematic Retargeting

The hand–object trajectories recovered in Section 3.2 describe the demonstrated interaction but do not directly specify how a robot with a different morphology should reproduce it. In particular, morphological differences can make human contacts unreachable or unsuitable, while MANO reconstruction errors, such as substantial hand–object penetration, can make extracted anchors unreliable. We therefore extract stable contacts and adapt them to the target embodiment while preserving the demonstrated interaction semantics. These adapted contacts then guide full-trajectory optimization under configuration and collision constraints, yielding a robot-compatible kinematic reference without yet assuming that the motion is dynamically executable.

Robot-Adapted Stable-Contact Guidance. We first extract raw stable contacts by evaluating the distance between each MANO fingertip surface point j and the object Gaussians at each frame t using $\phi^{\text{GS}}(O(q_t^{\text{ref,object}}), \mathbf{P}_{t,j}^{\text{mano}})$ in Eq. (2). We summarize stable contacts as triplets $(j, \mathbf{a}_{j,\mathcal{I}}, \mathcal{I})$,

each describing persistent contact between fingertip point j and an object-frame anchor $\mathbf{a}_{j,\mathcal{I}} \in \mathbb{R}^3$ over an interval $\mathcal{I} = [t_1, t_2]$. Using the correspondence set $\mathcal{C}$, we associate each MANO contact point j with robot keypoint i and denote its corresponding anchor by $\mathbf{a}_{i,\mathcal{I}}$. To adapt these contacts to the target hand, we select m keyframes at timesteps $\{t_k\}_{k\in\mathcal{K}}$ based on the extracted stable contacts and jointly optimize their robot configurations and contact anchors. Each anchor is constrained to the object surface and shared across keyframes within its contact interval:

$$\min_{\{q_{t_k}^{\text{key,robot}}\},\,\{\hat{\mathbf{a}}_{i,\mathcal{I}}\}} \quad \sum_{k\in\mathcal{K}} \sum_{(i,\mathcal{I}):\, t_k\in\mathcal{I}} \left\| \mathbf{P}_i^{\text{robot}}(q_{t_k}^{\text{key,robot}}) - q_{t_k}^{\text{ref,object}} \hat{\mathbf{a}}_{i,\mathcal{I}} \right\|_2^2 + \mathcal{L}_{\text{a}} + \mathcal{L}_{\text{fc}} + \mathcal{L}_{\text{reg}} \tag{3a}$$

$$\text{s.t.} \quad q_{t_k}^{\text{key,robot}} \in \mathcal{Q}^R, \qquad \Phi_{t_k} \geq 0, \qquad \phi^{\text{GS}}(O, \hat{\mathbf{a}}_{i,\mathcal{I}}) = 0. \tag{3b}$$

The first term, the contact-fitting loss $\mathcal{L}_{\text{c}}$, fits robot contact points to their shared refined anchors at keyframes within the corresponding contact intervals. The anchor term $\mathcal{L}_{\text{a}}$ penalizes deviations of the refined anchor layout from the original layout. The force-closure term $\mathcal{L}_{\text{fc}}$ aggregates losses over keyframes, penalizing the net wrench of bounded friction-cone contact forces and insufficient force-closure margins. The regularization term $\mathcal{L}_{\text{reg}}$ tracks the MANO palm pose and hand shape, aligns contact normals, and smooths motion across keyframes. Here, Φ_t collects hand–object distances from Eq. (2) and finger self-collision distances between link capsules. The constraints enforce configuration limits, nonpenetration, and anchors on the Gaussian surface. We use sequential quadratic programming (SQP) (Bonnans et al., 2006) with elastic relaxation. The refined anchors guide both retargeting stages. Loss definitions and solver settings are provided in Appendix D.

Constrained Kinematic Retargeting. Using the refined anchors as contact targets throughout their corresponding contact intervals, we next optimize the full kinematic trajectory while fixing collision-feasible key configurations indexed by $\mathcal{K}_{\text{f}} \subseteq \mathcal{K}$ and enforcing per-frame collision constraints:

$$\min_{\{q_t^{\text{ref,robot}}\}} \quad \sum_{t=0}^{T} \sum_{(i,j)\in\mathcal{C}} w_{t,i} \left\| \mathbf{P}_i^{\text{robot}}(q_t^{\text{ref,robot}}) - \mathbf{v}_{t,i} \right\|_2^2 + \mathcal{L}_{\text{reg}} \tag{4a}$$

$$\text{s.t.} \quad q_{t_k}^{\text{ref,robot}} = q_{t_k}^{\text{key,robot}}, \quad k \in \mathcal{K}_{\text{f}}, \qquad \Phi_t \geq 0, \qquad q_t^{\text{ref,robot}} \in \mathcal{Q}^R, \tag{4b}$$

$$\text{with} \quad \mathbf{v}_{t,i} = (1-\alpha_{t,i})\mathbf{P}_{t,j}^{\text{mano}} + \alpha_{t,i} q_t^{\text{ref,object}} \hat{\mathbf{a}}_{i,\mathcal{I}}. \tag{4c}$$

Here, $w_{t,i}$ weights position fitting, and $\mathbf{P}_i^{\text{robot}}$ denotes the position of the mapped contact query point, or the fingertip site ($\alpha_{t,i} \equiv 0$) for fingers without stable contacts. $\alpha_{t,i} \in \{0,1\}$ switches the target from the MANO correspondence to the refined contact anchor during stable contact. $\mathcal{L}_{\text{reg}}$ tracks the MANO hand shape and palm pose, aligns contact normals, and penalizes velocity and acceleration over the full trajectory. The palm reference is adjusted by interpolated keyframe offsets. Starting from a trajectory repaired to be collision-free at every frame, we solve the problem with Gauss–Newton iterations, each linearizing the residuals and constraints and solving a trust-region QP over all frames; a line search maintains collision feasibility at every frame (Appendix D).

3.4 Force-Aware Dynamics Retargeting

Kinematic retargeting provides a reference motion but does not determine the controls and contact forces required for execution. Force-aware dynamics retargeting therefore optimizes controls through a differentiable simulator, accounting for hand–object contact forces to generate trajectories under the modeled actuation and contact physics. We further calibrate the simulator's physical parameters to reduce model mismatch and support transfer to the target system.

Force-Aware Control Optimization. We integrate the Gaussian collision interface from Section 3.2 into ComFree-Sim (Borse, 2026) to obtain our differentiable simulator, ComFree-GS. We use multi-start gradient-based MPC, with gradients backpropagated through ComFree-GS. At each replanning step t, we optimize position-control knots $\mathbf{U}$ over an L-step rollout horizon:

$$\begin{aligned} \min_{\mathbf{U}} \quad & \|\mathbf{e}_{t+L}\|_{\mathbf{Q}_{\text{f}}}^2 + \sum_{m=t+1}^{t+L-1} \|\mathbf{e}_m\|_{\mathbf{Q}}^2 + J_{\text{c}}(\mathbf{x}) + J_{\text{col}}(\mathbf{x}) + \mathcal{R}(\mathbf{U}, \mathbf{x}) \\ \text{s.t.} \quad & x_{m+1} = f_\theta(x_m, \mathbf{u}_m), \quad \mathbf{u}_m = [\text{Interp}(\mathbf{U})]_m, \quad m = t, \ldots, t+L-1. \end{aligned} \tag{5}$$

Here, $\mathbf{e}_m$ measures robot and object configuration errors, and Interp interpolates control knots into per-step commands. J_{c} combines weighted refined-anchor alignment with contact-depth penalties;

J_{col} penalizes excessive penetration and inter-finger collisions. $\mathbf{Q}$ and $\mathbf{Q}_{\text{f}}$ weight intermediate and terminal tracking errors, while $\mathcal{R}$ regularizes control differences and hand velocities and accelerations. Contact forces are computed by the simulator. The current state is fixed, allowing only a bounded initial hand-pose correction in the first window. We optimize N candidates in parallel, keeping one warm start and perturbing the others, and select the lowest-cost numerically valid solution. We execute its initial portion, shift the horizon, and replan to generate optimized robot state and control trajectories. Implementation details are provided in Appendix E.

System Identification. To reduce model mismatch during transfer to the target system f_{target}, we calibrate the object and contact parameters of our simulator f. We replay controls $\mathbf{u}_{\text{stable}}$ from stable-contact segments open-loop in f_{target} to obtain target trajectories $x_{\text{stable}}^{\text{target}}$. Starting each simulated segment from its observed initial state $x_0 = x_0^{\text{target}}$, we identify θ by minimizing $\|x_{\text{stable}}(\theta) - x_{\text{stable}}^{\text{target}}\|_2^2$, with dynamics $x_{k+1} = f(x_k, \mathbf{u}_k; \theta)$ under the same replayed controls. Here, $\theta = (m, \mathbf{I}, \mu_c, \delta)$ comprises the object mass, inertia tensor, contact friction coefficient, and collision-distance offset. The offset adjusts the effective collision surface to compensate for residual geometric error. Parameter coupling may yield a multimodal identification landscape, making gradient-only optimization sensitive to local minima and sampling-only search costly. We therefore use TSMC (Yang, 2026) to combine particle-based exploration with gradient-informed refinement. The calibrated parameters update the simulator for subsequent dynamics retargeting.

4 Experiments

This section presents extensive experiments to address three questions: (1) **Gaussian Representation Quality.** Can our spherical Gaussians better capture object geometry and appearance? (2) **Trajectory Fidelity and Feasibility.** Can our framework generate high-fidelity, physically executable robot trajectories in simulation, and how do individual components contribute? (3) **Trajectory Transferability.** Can the generated trajectories transfer to other simulators and real robots?

4.1 Q1: Gaussian Representation Quality

We evaluate the rendering quality and geometric accuracy of our Gaussian representation on all 20 objects in DexYCB (Chao et al., 2021). For each object, we render RGB-D images from 72 viewpoints. We compare against a spherical variant of standard 3DGS (Kerbl et al., 2023) and ContactGaussian-WM's spherical Gaussian representation, training all three methods using the same RGB-D inputs. Rendering quality is measured by averaging PSNR over 256 novel views per object. To evaluate geometric accuracy, we sample one million query points inside and outside each object and compute the RMSE between the Gaussian collision distances from Eq. (1) and the ground-truth mesh signed distances. As shown in Figure 2, our method substantially improves geometric accuracy over spherical 3DGS while maintaining high rendering quality, and outperforms ContactGaussian-WM on both metrics. These results demonstrate that our representation combines faithful appearance with accurate collision distances, supporting its use in subsequent visual tracking and physical simulation.

Method	PSNR (dB) ↑	RMSE (mm) ↓
3DGS (spherical)	**45.600**	2.5705
ContactGaussian-WM	42.397	1.4979
Ours	43.564	**0.4065**

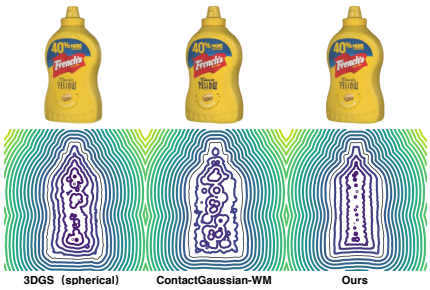


Figure 2: Quantitative results on all 20 objects in DexYCB (top); front-view rendering and distance-field visualization for a representative object (bottom).

4.2 Q2: Trajectory Fidelity and Feasibility.

4.2.1 Experimental Setup

Datasets and Robot Embodiments. We evaluate dexterous retargeting on 130 sequences covering 40 objects: 50 right-hand grasping sequences from DexYCB (Chao et al., 2021) and 80 precise in-hand manipulation sequences from HOT3D (Banerjee et al., 2025). Selection criteria are detailed in Appendix F. We reconstruct spherical Gaussian models for all objects and recover object pose trajectories and corresponding MANO hand motion from the interaction videos provided by both datasets for subsequent retargeting. We use seven target hands—Allegro, LEAP, Sharpa, XHand, Inspire, MetaHand, and Schunk—that vary substantially in degrees of freedom, size, and finger configuration. Figure 3 shows the diverse objects and target robot hands.

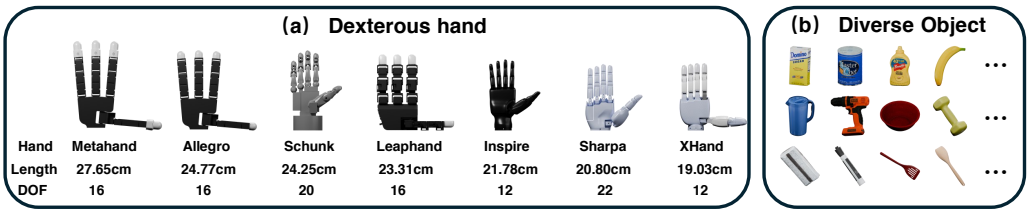


Figure 3: Dexterous hands and diverse objects used in evaluation. DexForge is evaluated on diverse objects and seven robotic hands that differ in size, degrees of freedom, and finger count.

Table 1: Success rates (%) across seven hands, averaged over five seeds. A: visual reconstruction (×: dataset-provided GT hand–object trajectories); B: contact guidance; C: contact refinement. All variants retain both retargeting stages. Best results, including ties, are shaded and bolded.

Dataset	Reconstruction	Contact-aware		Success rate (%) ↑						
	A	B	C	Allegro	Inspire	LEAP	MetaHand	Schunk	Sharpa	XHand
DexYCB	× (GT)	×	×	44	36	24	32	40	44	48
	× (GT)	✓	×	64	56	52	60	72	64	76
	× (GT)	✓	✓	**84**	**80**	**68**	**76**	**80**	**80**	**84**
	✓	✓	✓	**84**	**80**	64	**76**	**80**	**80**	**84**
HOT3D	× (GT)	×	×	4	0	0	0	4	12	8
	× (GT)	✓	×	20	12	24	16	16	24	20
	× (GT)	✓	✓	**32**	**16**	**32**	**28**	**32**	**48**	**36**
	✓	✓	✓	**32**	**16**	28	24	**32**	**48**	**36**

Evaluation Metrics. Let $\boldsymbol{\rho}_t$ and $\mathbf{r}_t$ denote object position and unit quaternion; superscript ref denotes the dataset-provided or reconstructed reference. Mean tracking errors are $E_{\text{pos}} = T^{-1}\sum_{t=1}^{T}\|\boldsymbol{\rho}_t - \boldsymbol{\rho}_t^{\text{ref}}\|_2$ and $E_{\text{rot}} = T^{-1}\sum_{t=1}^{T}\arccos(2\langle \mathbf{r}_t, \mathbf{r}_t^{\text{ref}}\rangle^2 - 1)$. Contact error is $E_{\text{contact}}^{\text{refined}} = (\sum_{(i,\mathcal{I})}|\mathcal{I}|)^{-1}\sum_{(i,\mathcal{I})}\sum_{t\in\mathcal{I}}\|\mathbf{P}_i^{\text{robot}}(q_t^{\text{robot}}) - q_t^{\text{object}}\hat{\mathbf{a}}_{i,\mathcal{I}}\|_2$; $E_{\text{contact}}^{\text{raw}}$ instead uses original anchors $\mathbf{a}_{i,\mathcal{I}}$. A trajectory succeeds if $E_{\text{rot}} < 0.1\,\text{rad}$ and $E_{\text{pos}} < 0.01\,\text{m}$.

4.2.2 Ablation Study

To assess individual component contributions, we compare our full method with three cumulative ablation variants: (1) replacing reconstructed hand–object trajectories with dataset-provided ones; (2) further disabling contact refinement; and (3) finally removing raw contact guidance. We evaluate all variants on five DexYCB and five HOT3D sequences covering diverse objects and contact patterns, using seven robotic hands in ComFree-GS. Stochastic optimization is repeated with five random seeds, and Table 1 reports mean success rates. Detailed settings are provided in Appendix G.

Ablation Analysis. All variants retain dynamics optimization, as kinematic trajectories alone cannot ensure physical executability. With dataset ground-truth (GT) hand–object trajectories, raw contact guidance improves retargeting success, and contact refinement yields further gains. These improvements are consistent with the components' roles: guidance provides explicit targets for establishing and maintaining contacts, while refinement adapts these targets to the robot's morphology through joint anchor and configuration optimization, accounting for reachability and encouraging force closure. Replacing GT trajectories with our reconstructions largely preserves success rates, with small decreases for some hands, supporting the framework's effectiveness with reconstructed inputs.

4.2.3 Retargeting Results

Contact-aware Kinematic Retargeting Results. We compare our method with SPIDER's IK-based stage (Pan et al., 2026) on all selected DexYCB and HOT3D sequences across seven hands. The baseline fits MANO-derived fingertip positions and wrist orientation without considering hand–object interactions or stable contacts. Both methods use dataset-provided GT trajectories for comparison; we additionally evaluate our method on our reconstructed trajectories. We report mean penetration $E_k = T^{-1}\sum_{t=1}^{T}\max(0, -\Phi_t^k)$, where Φ_t^k is the minimum hand–object ($k = \text{ho}$) or self-collision ($k = \text{self}$) distance at frame t (Eq. (4)). Both methods are also evaluated using $E_{\text{contact}}^{\text{raw}}$ and $E_{\text{contact}}^{\text{refined}}$ from Section 4.2.1. Table 2 and Figure 4 show substantially reduced penetration and contacts better adapted to the robot while preserving the demonstrated contact structure. Compa-

Table 2: Kinematic retargeting errors (mm, mean ± SD). E_{self}/E_{ho}: self-collision/hand–object penetration; contact errors measure raw/refined anchor tracking. GT/recon. denote dataset-provided/our reconstructed trajectories. Best results with GT inputs are shaded and bolded.

Dataset	Method	$E_{\text{self}} \downarrow$	$E_{\text{ho}} \downarrow$	$E^{\text{raw}}_{\text{contact}} \downarrow$	$E^{\text{refined}}_{\text{contact}} \downarrow$
DexYCB	IK (GT)	4.73 ± 5.02	6.37 ± 4.24	23.30 ± 32.58	24.93 ± 32.79
	Contact-aware (GT)	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{9.20 \pm 5.43}$	$\mathbf{3.92 \pm 2.12}$
	Contact-aware (recon.)	0.00 ± 0.00	0.00 ± 0.00	10.22 ± 7.64	3.81 ± 3.42
HOT3D	IK (GT)	5.37 ± 5.30	8.80 ± 5.01	23.72 ± 13.16	24.01 ± 13.38
	Contact-aware (GT)	$\mathbf{0.00 \pm 0.01}$	$\mathbf{0.01 \pm 0.01}$	$\mathbf{12.87 \pm 7.53}$	$\mathbf{5.59 \pm 6.59}$
	Contact-aware (recon.)	0.01 ± 0.01	0.02 ± 0.01	15.24 ± 8.91	5.76 ± 5.21

rable results with our reconstructions suggest limited impact on kinematic retargeting performance.

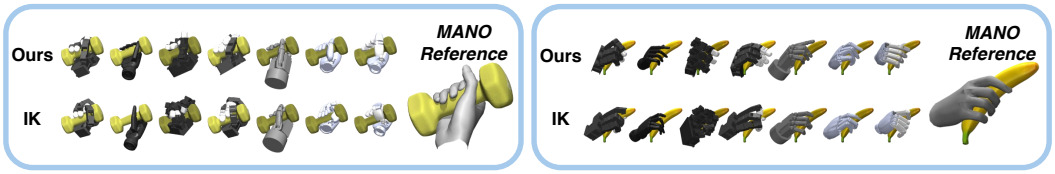


Figure 4: Visualization of selected kinematic retargeting results from Contact-aware and IK.

Force-Aware Dynamics Retargeting Results. We compare DexForge with sampling-based SPIDER (Pan et al., 2026) on all selected sequences across seven hands in ComFree-GS. For a controlled comparison, Contact-aware+SPIDER replaces SPIDER's IK stage with our contact-aware kinematic stage and supplies our contact guidance to SPIDER's optimizer. Both baselines retain official SPIDER hyperparameters, and all methods allow a bounded initial hand-pose correction. All methods use dataset-provided GT trajectories; we additionally evaluate DexForge on our reconstructions. As shown in Table 3 and Figure 5, Contact-aware+SPIDER improves over IK+SPIDER, particularly on DexYCB, showing that robot-adapted contact guidance also benefits sampling-based optimization. With the same kinematic stage and guidance, DexForge achieves the highest success rates and the lowest tracking errors on successful trajectories, a gain attributable to gradient-based force-aware refinement. The gap is largest on HOT3D, where objects are typically in hand from the first frame and SPIDER mostly fails through object drops, suggesting that simulation gradients help sustain precise in-hand contact. DexForge remains effective with reconstructed inputs, and visualizations show more faithful motion reproduction than the baselines across diverse hand morphologies. Hyperparameter settings, sensitivity analysis, and implementation details are in Appendix H.

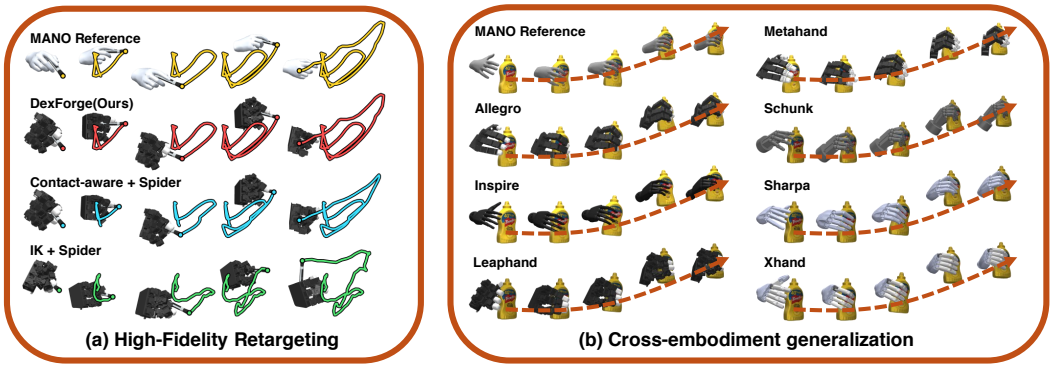


Figure 5: Qualitative results of force-aware dynamics retargeting. (a) Motion reproduction by our method and baselines relative to the MANO reference. (b) Retargeting across seven robotic hands.

4.3 Q3: Trajectory Transferability

Sim-to-Sim Open-Loop Transfer. We evaluate open-loop transfer from ComFree-GS to MuJoCo-CPU on the same subsets as the ablation study, with all methods using our reconstructed trajectories. The target uses mesh-based collision detection and object-specific physical parameters, while

Table 3: Dynamics retargeting results across seven hands (mean ± SD over five seeds). Position and rotation errors are computed over successful trajectories only. GT/recon. denote dataset-provided/our reconstructed trajectories. Best results with GT inputs are shaded and bolded.

Dataset	Method	Position error (mm) ↓	Rotation error (°) ↓	Success rate (%) ↑
DexYCB	IK+SPIDER (GT)	8.21 ± 1.22	4.22 ± 0.84	24.9 ± 1.6
	Contact-aware+SPIDER (GT)	7.57 ± 1.43	3.37 ± 0.92	50.6 ± 2.1
	DexForge (GT)	$\mathbf{2.73 \pm 1.78}$	$\mathbf{0.93 \pm 0.70}$	$\mathbf{77.4 \pm 1.6}$
	DexForge (recon.)	3.24 ± 1.93	0.97 ± 0.63	74.1 ± 1.9
HOT3D	IK+SPIDER (GT)	7.55 ± 1.74	4.70 ± 0.61	0.4 ± 0.1
	Contact-aware+SPIDER (GT)	7.07 ± 2.06	3.29 ± 1.05	0.6 ± 0.2
	DexForge (GT)	$\mathbf{5.01 \pm 2.55}$	$\mathbf{1.34 \pm 0.95}$	$\mathbf{34.9 \pm 2.1}$
	DexForge (recon.)	5.79 ± 2.31	1.68 ± 0.97	31.2 ± 2.6

Table 4: Sim-to-sim transfer success rates (%, mean ± SD across five seeds) before and after system identification. Best means are shaded and bolded.

Dataset	Retargeting method	Before identification	After identification		
			Gradient-only	Sampling-only	TSMC
DexYCB	IK+SPIDER	8.6 ± 2.9	11.4 ± 2.9	18.3 ± 1.6	22.3 ± 3.7
	Contact-aware+SPIDER	15.4 ± 1.6	19.4 ± 1.3	31.4 ± 2.0	36.0 ± 2.6
	DexForge	$\mathbf{25.7 \pm 2.0}$	$\mathbf{36.6 \pm 1.3}$	$\mathbf{53.1 \pm 2.6}$	$\mathbf{58.9 \pm 1.6}$
HOT3D	IK+SPIDER	2.3 ± 1.3	4.6 ± 1.6	8.0 ± 1.3	14.3 ± 2.9
	Contact-aware+SPIDER	4.0 ± 1.6	6.9 ± 2.6	12.0 ± 2.4	15.4 ± 2.6
	DexForge	$\mathbf{9.1 \pm 1.3}$	$\mathbf{13.1 \pm 1.6}$	$\mathbf{26.9 \pm 1.6}$	$\mathbf{33.1 \pm 1.6}$

ComFree-GS uses Gaussian collisions and shared initial parameters. Per-sequence system identification calibrates the physical parameters and collision-distance offset δ to reduce dynamics mismatch and compensate for residual geometric error. We then reoptimize controls in the calibrated simulator and replay them open-loop in MuJoCo-CPU. We compare IK+SPIDER, Contact-aware+SPIDER, and our method, and evaluate TSMC against gradient-only and sampling-only identification. Transfer succeeds when the mean tracking errors satisfy $E_{\text{pos}} < 0.02\,\text{m}$ and $E_{\text{rot}} < 0.2\,\text{rad}$. Table 4 shows that system identification improves transfer success across both datasets, with TSMC outperforming the two alternatives. Experimental details are provided in Appendix I.

Sim-to-Real Open-Loop Transfer. Lacking accurate object-pose measurements, we evaluate transfer without system identification. For five YCB objects, we run the full retargeting pipeline on our reconstructions with known physical parameters. The resulting controls are replayed open-loop on a Unitree G1 with a LEAP Hand v1 (Shaw et al., 2023), from simulation-matched initial positions. Over 10 trials per object, executions (Figure 6) achieve an 80% success rate in stable grasp-and-hold, which measures executability rather than tracking fidelity (Appendix J).

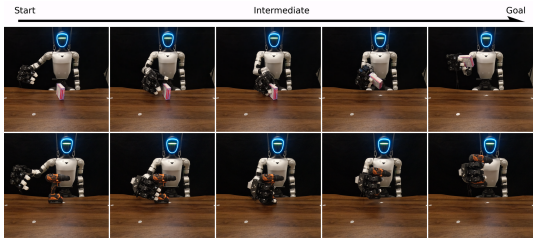


Figure 6: Sim-to-real open-loop transfer. Controls optimized in ComFree-GS are directly replayed on a Unitree G1 equipped with a LEAP Hand v1.

5 Conclusion

DexForge connects visual reconstruction with Gaussian-based differentiable simulation to faithfully reproduce human manipulation from video demonstrations across diverse robot embodiments. Robot-adapted stable contacts guide both collision-constrained kinematic reference construction and gradient-based dynamics refinement, substantially improving retargeting success and object-tracking accuracy. Sim-to-sim transfer and real-world open-loop execution further demonstrate the applicability of the generated trajectories. These results highlight the potential of combining contact adaptation and differentiable physics to translate human manipulation experience into high-quality robot demonstrations for learning precise manipulation skills.

AI USE STATEMENT

Generative AI tools were used to improve the clarity and readability of the manuscript and to assist in retrieving and discovering related work. All retrieved references were verified by the authors against the original sources. The authors reviewed and edited all AI-assisted text and take full responsibility for the final content.

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. We use the publicly available DexYCB and HOT3D datasets in accordance with their respective licenses. We respect the privacy of individuals represented in the data and the conditions governing data use and redistribution. Our research aims to support the responsible development of dexterous robotic manipulation. The authors take full responsibility for the accuracy and integrity of the methods, results, and conclusions presented in this paper.

REPRODUCIBILITY STATEMENT

The appendices provide dataset selection criteria, implementation details, optimization settings, and evaluation protocols. Our experiments use the publicly available DexYCB and HOT3D datasets. Additional results and a link to our anonymous code repository are available on our project website: `https://dexforge-42306.github.io/project-page/`.

REFERENCES

Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, et al. Hot3d: Hand and object tracking in 3d from egocentric multi-view videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7061–7071. IEEE, 2025.

Onur Beker, Nico Gürtler, Ji Shi, A René Geist, Amirreza Razmjoo, Georg Martius, and Sylvain Calinon. A smooth analytical formulation of collision detection and rigid body dynamics with contact. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 18558–18565. IEEE, 2025.

Onur Beker, Andreas René Geist, Anselm Paulus, Nico Gürtler, Ji Shi, Sylvain Calinon, and Georg Martius. Smoothly differentiable and efficiently vectorizable contact manifold generation. *arXiv preprint arXiv:2602.20304*, 2026.

Gino van den Bergen. Efficient collision detection of complex deformable models using aabb trees. *Journal of graphics tools*, 2(4):1–13, 1997.

J Frédéric Bonnans, J Charles Gilbert, Claude Lemaréchal, and Claudia A Sagastizábal. *Numerical optimization: theoretical and practical aspects*. Springer, 2006.

Chetan Borse. Comfree-sim: A gpu-parallelized analytical contact physics engine for scalable contact-rich robotics simulation and control. Master's thesis, Arizona State University, 2026.

Dingding Cai, Janne Heikkilä, and Esa Rahtu. Gs-pose: Generalizable segmentation-based 6d object pose estimation with 3d gaussian splatting, 2024. URL https://arxiv.org/abs/2403.10683.

Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9040–9049. IEEE, 2021.

Yunhai Feng, Natalie Leung, Jiaxuan Wang, Lujie Yang, Haozhi Qi, and Preston Culbertson. A minimalist retargeting-guided reinforcement learning recipe for dexterous manipulation. *arXiv preprint arXiv:2607.11874*, 2026.

Wanxin Jin. Complementarity-free multi-contact modeling and optimization for dexterous manipulation. *arXiv preprint arXiv:2408.07855*, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

Arjun S Lakshmipathy, Jessica K Hodgins, and Nancy S Pollard. Kinematic motion retargeting for contact-rich anthropomorphic manipulations. *ACM Transactions on Graphics*, 44(2):1–20, 2025.

Kailin Li, Puhao Li, Tengyu Liu, Yuyang Li, and Siyuan Huang. Maniptrans: Efficient dexterous bimanual manipulation transfer via residual learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6991–7003. IEEE, 2025.

Xueyi Liu, Kangbo Lyu, Jieqiong Zhang, Tao Du, and Li Yi. Parameterized quasi-physical simulators for dexterous manipulations transfer. In *European Conference on Computer Vision*, pp. 164–182. Springer, 2024.

Xueyi Liu, Jianibieke Adalibieke, Qianwei Han, Yuzhe Qin, and Li Yi. Dextrack: Towards generalizable neural tracking control for dexterous manipulation from human references. In *International Conference on Learning Representations*, volume 2025, pp. 85970–85994, 2025.

Haozhe Lou, Mingtong Zhang, Haoran Geng, Hanyang Zhou, Sicheng He, Zhiyuan Gao, Siheng Zhao, Jiageng Mao, Pieter Abbeel, Jitendra Malik, Daniel Seita, and Yue Wang. D-rex: Differentiable real-to-sim-to-real engine for learning dexterous grasping, 2026. URL https://arxiv.org/abs/2603.01151.

Tyler Ga Wei Lum, Olivia Y. Lee, C. Karen Liu, and Jeannette Bohg. Crossing the human-robot embodiment gap with sim-to-real rl using one human demonstration, 2025. URL https://arxiv.org/abs/2504.12609.

Zhao Mandi, Yifan Hou, Dieter Fox, Yashraj Narang, Ajay Mandlekar, and Shuran Song. Dexmachina: Functional retargeting for bimanual dexterous manipulation. *arXiv preprint arXiv:2505.24853*, 2025.

Louis Montaut, Quentin Le Lidec, Antoine Bambade, Vladimir Petrik, Josef Sivic, and Justin Carpentier. Differentiable collision detection: a randomized smoothing approach. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3240–3246. IEEE, 2023.

Juncheng Mu, Sizhe Yang, Yiming Bao, Hojin Bae, Tianming Wei, Linning Xu, Boyi Li, Huazhe Xu, and Jiangmiao Pang. Deximit: Learning bimanual dexterous manipulation from monocular human videos. *arXiv preprint arXiv:2602.10105*, 2026.

Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1366–1373. IEEE, 2017. doi: 10.1109/iros.2017.8202315. URL http://dx.doi.org/10.1109/IROS.2017.8202315.

Bhawna Paliwal, Haritheja Etukuru, William Liang, Pieter Abbeel, Nur Muhammad Mahi Shafiullah, and Jitendra Malik. Do as i do: Dexterous manipulation data from everyday human videos, 2026. URL https://arxiv.org/abs/2606.19333.

Chaoyi Pan, Changhao Wang, Haozhi Qi, Zixi Liu, Homanga Bharadhwaj, Akash Sharma, Tingfan Wu, Guanya Shi, Jitendra Malik, and Francois Hogan. Spider: Scalable physics-informed dexterous retargeting, 2026. URL https://arxiv.org/abs/2511.09484.

Anselm Paulus, Andreas René Geist, Pierre Schumacher, Vít Musil, Simon Rappenecker, and Georg Martius. Differentiable simulation of hard contacts with soft gradients for learning and control. In *International Conference on Learning Representations*, volume 2026, pp. 40599–40626, 2026.

Georgios Pavlakos, Dandan Shan, Ilija Radosavovic, Angjoo Kanazawa, David Fouhey, and Jitendra Malik. Reconstructing hands in 3d with transformers, 2023. URL https://arxiv.org/abs/2312.05251.

Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. Wilor: End-to-end 3d hand localization and reconstruction in-the-wild, 2025. URL https://arxiv.org/abs/2409.12259.

Liyuan Qi, Olaoluwa Popoola, Muhammad Ali Imran, and Wasim Ahmad. Genhand: generalised human grasp kinematic retargeting. *npj Robotics*, 4(1):19, 2026.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. URL https://arxiv.org/abs/2408.00714.

Jie Ren, Zhehao Jiang, Yinhong Yang, Haorui Jia, Han Jiang, Ben Li, Yao Yao, Cheng Lin, Qiu Shen, Zhenshan Bing, et al. C2dex: Contact-consistent reconstruction and retargeting for dexterous manipulation from monocular video. *arXiv preprint arXiv:2608.07045*, 2026.

Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022.

Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *arXiv preprint arXiv:2309.06440*, 2023.

Jin-Chuan Shi, Binhong Ye, Tao Liu, Xiaoyang Liu, Yangjinhui Xu, Junzhe He, Zeju Li, Hao Chen, and Chunhua Shen. Agile: Hand-object interaction reconstruction from video via agentic generation. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2026.

Tony Tao, Mohan Kumar Srirama, Jason Jingzhou Liu, Kenneth Shaw, and Deepak Pathak. Dexwild: Dexterous human interactions for in-the-wild robot policies. *arXiv preprint arXiv:2505.07813*, 2025.

Yinxiao Tian, Ziyi Yang, Zinan Zhao, and Zhen Kan. Kilohertz-safe: A scalable framework for constrained dexterous retargeting. *arXiv preprint arXiv:2603.29213*, 2026.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Kevin Tracy, Taylor A Howell, and Zachary Manchester. Differentiable collision detection for a set of convex primitives. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3663–3670. IEEE, 2023.

Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C. Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation, 2024. URL https://arxiv.org/abs/2403.07788.

Meizhong Wang, Wanxin Jin, Kun Cao, Lihua Xie, and Yiguang Hong. Contactgaussian-wm: Learning physics-grounded world model from videos. *arXiv preprint arXiv:2602.11021*, 2026.

Shibo Wang, Haonan He, Maria Parelli, Christoph Gebhardt, Zicong Fan, and Jie Song. Magichoi: Leveraging 3d priors for accurate hand-object reconstruction from short monocular video clips. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5957–5968. IEEE, 2025.

Jielin Wu, Shenzhe Yao, Guanqi He, Xiaohan Liu, Zhaoqing Zeng, Xiangrui Jiang, Han Yang, Wentao Zhang, and Hang Zhao. Toporetarget: Interaction-preserving retargeting for dexterous manipulation. *arXiv preprint arXiv:2606.16272*, 2026.

Chendong Xin, Mingrui Yu, Yongpeng Jiang, Zhefeng Zhang, and Xiang Li. Analyzing key objectives in human-to-robot retargeting for dexterous manipulation. *IEEE Robotics and Automation Practice*, 2026.

Mengda Xu, Han Zhang, Yifan Hou, Zhenjia Xu, Linxi Fan, Manuela Veloso, and Shuran Song. Dexumi: Using human hand as the universal manipulation interface for dexterous manipulation. *arXiv preprint arXiv:2505.21864*, 2025a.

Sirui Xu, Yu-Wei Chao, Liuyu Bian, Arsalan Mousavian, Yu-Xiong Wang, Liangyan Gui, and Wei Yang. Dexplore: Scalable neural control for dexterous manipulation from reference scoped exploration. In *Conference on Robot Learning*, pp. 2184–2199. PMLR, 2025b.

Heng Yang. Tempered sequential monte carlo for trajectory and policy optimization with differentiable dynamics. *arXiv preprint arXiv:2604.21456*, 2026.

Lujie Yang, HJ Suh, Tong Zhao, Bernhard Paus Graesdal, Tarik Kelestemur, Jiuguang Wang, Tao Pang, and Russ Tedrake. Physics-driven data generation for contact-rich manipulation via trajectory optimization. *arXiv preprint arXiv:2502.20382*, 2025.

Lujie Yang, Xiaoyu Huang, Zhen Wu, Angjoo Kanazawa, Pieter Abbeel, Carmelo Sferrazza, C. Karen Liu, Rocky Duan, and Guanya Shi. Omniretarget: Interaction-preserving data generation for humanoid whole-body loco-manipulation and scene interaction, 2026. URL https://arxiv.org/abs/2509.26633.

Xiaohan Ye, Kui Wu, Taku Komura, et al. Efficient differentiable contact model with long-range influence. In *International Conference on Learning Representations*, volume 2026, pp. 63535–63559, 2026.

Zhao-Heng Yin, Changhao Wang, Luis Pineda, Krishna Bodduluri, Tingfan Wu, Pieter Abbeel, and Mustafa Mukadam. Geometric retargeting: A principled, ultrafast neural hand retargeting algorithm, 2025. URL https://arxiv.org/abs/2503.07541.

Gu Zhang, Qicheng Xu, Haozhe Zhang, Jianhan Ma, Long He, Yiming Bao, Zeyu Ping, Zhecheng Yuan, Chenhao Lu, Chengbo Yuan, et al. Unidex: A robot foundation suite for universal dexterous hand control from egocentric human videos. *arXiv preprint arXiv:2603.22264*, 2026.

Ruijie Zheng, Dantong Niu, Yuqi Xie, Jing Wang, Mengda Xu, Yunfan Jiang, Fernando Castañeda, Fengyuan Hu, You Liang Tan, Letian Fu, Trevor Darrell, Furong Huang, Yuke Zhu, Danfei Xu, and Linxi Fan. Egoscale: Scaling dexterous manipulation with diverse egocentric human data, 2026. URL https://arxiv.org/abs/2602.16710.

René Zurbrügg, Andrei Cramariuc, and Marco Hutter. Graspqp: Differentiable optimization of force closure for diverse and robust dexterous grasping. *arXiv preprint arXiv:2508.15002*, 2025.

A GAUSSIAN TRAINING DETAILS

We jointly optimize Gaussian geometry and appearance using RGB reconstruction and signed-distance supervision from multi-view observations. This section describes reference SDF construction, supervision sampling, and the joint objective combining geometric and photometric losses.

Reference SDF Construction. We fuse multi-view depth maps into a truncated signed distance field (TSDF) and convert it into a reference Euclidean signed distance field ϕ^{ref} following (Oleynikova et al., 2017). The reference field measures the shortest Euclidean distance to the reconstructed surface, with negative values inside the object and positive values outside, and provides geometric supervision for the Gaussian collision representation during training.

Supervision Sampling. We sample 10^6 points near the object surface and query their reference distances $d_i = \phi^{\mathrm{ref}}(p_i)$. We store these pairs as $\mathcal{D}_{\mathrm{SDF}} = \{(p_i, d_i)\}_{i=1}^{10^6}$ and randomly draw mini-batches $\mathcal{B} \subset \mathcal{D}_{\mathrm{SDF}}$ during training, avoiding repeated reference-distance queries.

Joint Training Objective. We combine photometric and distance-matching losses:

$$\mathcal{L} = \mathcal{L}_{\mathrm{3DGS}} + \lambda_{\mathrm{SDF}}\mathcal{L}_{\mathrm{SDF}}, \qquad \mathcal{L}_{\mathrm{SDF}} = \frac{1}{|\mathcal{B}|} \sum_{(p_i,d_i)\in\mathcal{B}} \left(\phi^{\mathrm{GS}}(p_i) - d_i\right)^2. \tag{6}$$

Here, $\mathcal{L}_{\mathrm{3DGS}}$ combines the standard ℓ_1 and D-SSIM reconstruction losses, and λ_{SDF} balances geometric and photometric supervision. The distance loss updates Gaussian centers and scales, while the photometric objective updates both geometry and appearance.

B HAND AND OBJECT MOTION RECONSTRUCTION

We reconstruct MANO hand motion and object poses from synchronized, calibrated multi-view RGB videos using the pretrained Gaussian object model O. Hand–object foreground masks are obtained with SAM 2 (Ravi et al., 2024) video segmentation, prompted once in the first frame of each view. This stage uses only RGB observations; depth supervision is used separately for object-model training.

Multi-View MANO Fitting. We detect right hands using the WiLoR (Potamias et al., 2025) hand detector and obtain per-view MANO estimates with HaMeR (Pavlakos et al., 2023). Confidence-weighted rotation averaging and keypoint triangulation initialize the hand motion in world coordinates. We jointly optimize per-frame poses and translations with a shared hand shape, minimizing robust multi-view reprojection error, temporal acceleration, and deviations from the initial estimates. A strong pose prior limits articulation changes caused by noisy keypoint predictions. Optimization uses Adam for 800 iterations.

Object Pose Tracking. We extend the render-and-compare refinement of GS-Pose (Cai et al., 2024) to multiple views while keeping Gaussian geometry and appearance fixed. The fitted MANO mesh provides hand masks that exclude occluded pixels. Object poses are optimized through differentiable rendering using silhouette overlap and photometric losses over the remaining pixels. The first frame uses multiple rotation hypotheses, while subsequent frames are initialized from the previous pose and a constant-velocity prediction. Coarse-to-fine refinement, rotation regularization, and loss-triggered re-initialization improve tracking robustness.

Hand–Object Refinement. With object poses fixed, we further refine the hand using the Gaussian collision distances in Eq. 2. Vertex–frame pairs within $5\,\mathrm{mm}$ of the object, including penetrating vertices, form a fixed candidate contact set. We augment the hand-fitting objective with squared penetration and contact-distance penalties, weighted by 10 and 1, respectively, with distances measured in millimeters, and optimize for another 400 iterations. The resulting MANO motion and object poses provide the reference trajectories for retargeting.

Reconstruction Evaluation. We evaluate all 130 selected sequences using object translation and rotation errors, world-frame and wrist-aligned MPJPE, and hand–object penetration depth and penetrating-frame percentage. Table 5 reports means and standard deviations across sequences, while Fig. 7 shows a qualitative DexYCB example. Our method achieves accurate reconstruction with eight views on DexYCB and one RGB plus two grayscale SLAM views on HOT3D, with substantially less penetration than dataset annotations on both datasets.

Table 5: Hand–object reconstruction results on real multi-view videos from DexYCB and HOT3D (mean ± SD across sequences). GT denotes dataset annotations. Lower is better for all metrics.

Dataset	Method	Object pose		Hand accuracy		Geometric consistency	
		Translation (mm)	Rotation (°)	World-frame MPJPE (mm)	Wrist-aligned MPJPE (mm)	Penetration depth (mm)	Penetrating frames (%)
DexYCB	GT	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	3.15 ± 1.79	59.5 ± 13.6
	Ours	0.40 ± 0.16	0.46 ± 0.38	6.42 ± 1.18	4.82 ± 0.86	0.24 ± 0.15	3.9 ± 8.6
HOT3D	GT	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	6.13 ± 3.04	97.8 ± 7.3
	Ours	12.23 ± 3.11	4.83 ± 0.78	23.47 ± 2.34	9.21 ± 1.21	1.38 ± 0.21	7.8 ± 4.7

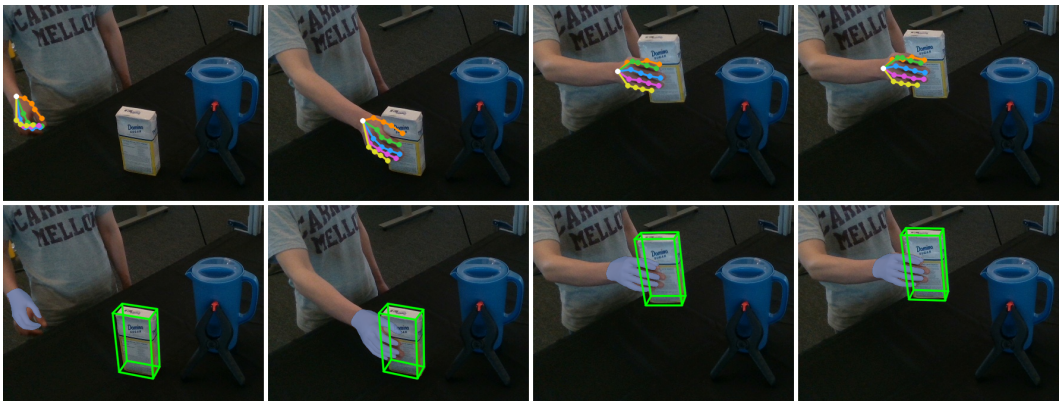


Figure 7: Visualization of the reconstructed hand motion and the estimated object pose on a DexYCB sequence, overlaid on the original video from one camera view. Columns show four evenly spaced frames in temporal order. Top: the reconstructed 21 hand keypoints projected into the image. Bottom: the reconstructed MANO hand mesh and the estimated object pose, visualized as the green 3D bounding box of the object's Gaussian model.

C Gaussian Collision Detection

We describe the spatial acceleration, contact geometry, differentiation, and simulator integration of Gaussian-based collision detection. Each object is represented by spherical Gaussian primitives rigidly attached to its body, with collision radii $r_i = 3\sigma_i$. Robot surface query points are represented as zero-radius primitives and handled by the same collision formulation. All centers used in the following distance and contact formulas are expressed in a common world coordinate frame.

BVH Construction. We construct binary bounding volume hierarchies (BVHs) (Bergen, 1997) over each object's primitives in its body-local coordinate frame using a binned surface-area heuristic. Each node stores an axis-aligned bounding box of the primitive centers and their maximum collision radius. The distance between two center bounding boxes, minus their maximum radii, provides a conservative lower bound on the corresponding sphere-pair distances. The hierarchies are constructed during model loading and reused under rigid-body motion. During traversal, relative body transforms are applied to express the bounding boxes in a common coordinate frame.

Distance Evaluation. Collision queries are performed for each source–target primitive-group pair; an ungrouped object is treated as a single group. Let $\mathcal{I}_\mathcal{A}$ and $\mathcal{I}_\mathcal{B}$ denote the primitive index sets of the queried groups. We compute the sphere-pair separation and select the minimum-distance pair:

$$\begin{aligned} d_{ij} &= \left\|\boldsymbol{\mu}_j^\mathcal{B} - \boldsymbol{\mu}_i^\mathcal{A}\right\|_2 - r_i^\mathcal{A} - r_j^\mathcal{B}, \\ (i^\star, j^\star) &= \arg\min_{i,j} d_{ij}, \qquad i \in \mathcal{I}_\mathcal{A}, \quad j \in \mathcal{I}_\mathcal{B}. \end{aligned} \tag{7}$$

Brute-force evaluation of Eq. (2) requires $O(N^\mathcal{A} N^\mathcal{B})$ pairwise distance computations. Joint BVH traversal maintains the best sphere-pair distance found so far and prunes node pairs whose conservative bounds cannot improve it, subject to a numerical tolerance. Unpruned leaf-node pairs are evaluated by enumerating their sphere pairs. The initial search cutoff is determined by the contact-

generation threshold. This procedure accelerates the minimum-distance search over the queried groups without materializing all primitive pairs. Each group pair contributes at most one contact, and no contact is generated when no sphere pair satisfies the generation threshold.

Contact Points and Normals. For the selected sphere pair, we define the relative center displacement and a regularized direction from $\mathcal{A}$ to $\mathcal{B}$:

$$\mathbf{v} = \boldsymbol{\mu}^{\mathcal{B}}_{j^\star} - \boldsymbol{\mu}^{\mathcal{A}}_{i^\star}, \qquad \mathbf{n} = \frac{\mathbf{v}}{\sqrt{\mathbf{v}^\top \mathbf{v} + \epsilon^2}}, \quad \epsilon = 10^{-9}. \tag{8}$$

The contact distance and representative contact point are computed as

$$\phi = d_{i^\star j^\star} + \Delta, \qquad \mathbf{p} = \boldsymbol{\mu}^{\mathcal{B}}_{j^\star} - r^{\mathcal{B}}_{j^\star}\mathbf{n} + \Delta\mathbf{n}, \tag{9}$$

where Δ is a distance offset, set to zero by default. With $\Delta = 0$ and noncoincident centers, $\mathbf{p}$ approximates the point on the selected target sphere facing the source sphere. This point is not guaranteed to lie on the outer boundary of the complete sphere union. The contact frame is constructed from $\mathbf{n}$, with its orientation adjusted to match the geometry ordering expected by the dynamics solver. The regularizer prevents division by zero in the direction computation, although coincident centers remain a degenerate contact configuration.

Differentiation. In the differentiable simulation path, BVH traversal and minimum-distance pair selection are performed outside the differentiation graph. The selected primitive indices are then held fixed while their world-space centers, contact distance, contact point, and contact frame are recomputed within the differentiable computation. Gradients propagate through these continuous quantities to body poses and, through the dynamics, to the optimized controls. We do not differentiate through BVH traversal, discrete pair selection, or contact activation decisions. For a unique minimizing pair and nondegenerate contact geometry, this computation differentiates the selected contact branch; pair switches and contact activation changes can introduce nonsmooth transitions.

Simulator Integration. We provide JAX and Warp collision backends for MJX and MJWarp, respectively. Our scene-loader extensions accept Gaussian collision declarations such as `<geom type="gs" file="model.ply"/>` and load the corresponding primitive parameters. The collision routine supplies distances, contact points, and oriented contact frames through MuJoCo-style contact interfaces (Todorov et al., 2012), together with the contact parameters specified by the scene. These extensions modify geometry loading and collision dispatch while preserving the remaining scene configuration, downstream solver, and simulation loop. We integrate the routine into ComFree-Sim (Borse, 2026) to obtain ComFree-GS.

Effect of BVH Acceleration. We compare BVH-based queries with GPU brute-force evaluation in a fixed hand–object scene on an NVIDIA RTX 4090D GPU. Each environment contains 2,000 query points sampled from the robot hand surface and an object represented by 10,000 spherical Gaussians. Both methods use identical geometry and poses to ensure a controlled comparison. Figure 8 reports the forward collision-query time per batch for 1–64 parallel environments. Brute-force runtime grows approximately linearly with the number of environments, whereas BVH runtime increases only modestly across the tested batch sizes. BVH achieves speedups of $1.75\times$ and $62.09\times$ at 1 and 64 environments, respectively, while producing identical contact outputs. Timings exclude backward computation, BVH construction, compilation, and dynamics solving.

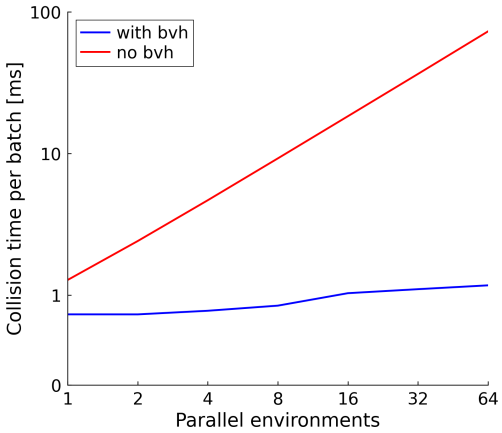


Figure 8: Forward collision-query time with and without BVH for 1–64 parallel environments on an NVIDIA RTX 4090D GPU.

D Stable Contact Extraction and Robot Adaptation

D.1 Stable Contact Extraction

For each finger, the candidate contact points are the MANO vertices on the palmar side of its distal pad. At each frame t, we query their Gaussian distances at the reference object pose $q_t^{\mathrm{ref,object}}$ and project them onto the Gaussian surface in the object frame. A vertex is in contact when its distance lies within $\pm 10\,\mathrm{mm}$, and contact runs shorter than eight frames at $30\,\mathrm{Hz}$ are discarded. Each run is split into maximal intervals $\mathcal{I}$ over which the projected points stay consistent, meaning that at least 80% of them lie within $8\,\mathrm{mm}$ of their medoid; this medoid defines the anchor $\mathbf{a}_{j,\mathcal{I}}$. For each finger, we then select a non-overlapping set of these candidates that covers the most frames, penalizing switches between candidates. Each selected candidate yields a triplet $(j, \mathbf{a}_{j,\mathcal{I}}, \mathcal{I})$, so a finger can have multiple anchors at different stages of the demonstration.

D.2 Joint Contact and Configuration Optimization

Keyframes and Initialization. We split the trajectory whenever a stable contact starts or ends and select one keyframe per segment: frame 0 for the initial segment and the middle frame otherwise. We also add frames $t-1$, t, and $t+1$ around each transition at t to support smooth configuration changes; the deduplicated frames form $\mathcal{K}$. All keyframes are initialized from a single seed, computed at the first frame with the most fingers in stable contact. This seed fits the robot fingertip pads to the anchors and their normals under joint limits and collision constraints, starting from several postures aligned with the MANO palm, and is repaired to the nearest collision-free configuration when needed. DexYCB keyframes reuse the seed directly, whereas for HOT3D the seed palm follows each keyframe's MANO palm.

Anchor Regularization. The anchor loss $\mathcal{L}_{\mathrm{a}}$ aggregates contributions over keyframes with active contacts. For a keyframe with n active anchors, its contribution is

$$\frac{\|\bar{\hat{\mathbf{a}}} - \bar{\mathbf{a}}\|_2^2}{R^2} + \frac{1}{nR^2}\sum_i \|(\hat{\mathbf{a}}_i - \bar{\hat{\mathbf{a}}}) - (\mathbf{a}_i - \bar{\mathbf{a}})\|_2^2,$$

where bars denote centroids and R is the object radius. The two terms penalize centroid displacement and deviations from the original relative anchor layout, respectively, both with unit weight; the latter is averaged over active anchors.

Force-Closure Residuals. We distinguish incomplete contact sets, which contain fewer than three contacts or are strict subsets of larger observed contact sets, from complete sets, which are maximal contact sets with at least three contacts. We use $\mu = 0.2$ throughout. For an incomplete set, each contact force is represented by four linearized friction-cone edges:

$$\mathbf{f}_i = \sum_{\ell=1}^{4} \beta_{i\ell}\mathbf{e}_{i\ell}, \qquad \mathbf{e}_{i\ell} = \frac{\mathbf{n}_i + \mu\mathbf{t}_{i\ell}}{4}, \qquad 1 \le \beta_{i\ell} \le 21,$$

where $\mathbf{n}_i$ is the contact normal and $\mathbf{t}_{i\ell}$ are tangential directions. The coefficient bounds correspond to $F_{\max} = 20$ plus a lower bound of 1, which excludes the trivial zero-force solution. We solve bounded least squares to minimize the six-dimensional net-wrench residual

$$\mathbf{w} = \begin{bmatrix} \sum_i \mathbf{f}_i \\ \sum_i \left((\mathbf{p}_i - \mathbf{c}_{\mathrm{obj}})/R\right) \times \mathbf{f}_i \end{bmatrix},$$

where $\mathbf{p}_i$ is the contact position and $\mathbf{c}_{\mathrm{obj}}$ is the object center of mass. The moment arms are normalized by R.

For a complete contact set, we solve a second-order cone program:

$$\max_{\{\mathbf{f}_i\},\, r} \quad r \qquad \text{s.t.} \quad \mathbf{w} = 0, \qquad \sum_i f_{i,\mathrm{n}} = 1, \qquad \mu f_{i,\mathrm{n}} - \|f_{i,\mathrm{t}}\|_2 \ge r \quad \forall i,$$

where $f_{i,\mathrm{n}}$ and $f_{i,\mathrm{t}}$ are the normal and tangential force components. Normalizing the total normal force fixes the force scale. The scalar residual is $\exp(-r/(2\mu))$, with a linear continuation when $r/(2\mu) < -3$. Both residual types enter the least-squares objective with weights proportional to

the duration of the corresponding contact state; the shared force-closure weight is 1. Both residuals are differentiated analytically with respect to the contact points and normals, and chained through the forward-kinematics Jacobians of the robot pad points and the derivative of the nearest-sphere Gaussian normal; the anchors do not enter $\mathcal{L}_{\mathrm{fc}}$. For the bounded least-squares residual, bound-active coefficients are held fixed and the free coefficients are differentiated implicitly through their normal equations. For the margin r, we use the envelope derivative given by the optimal primal and dual solutions of the SOCP, which accounts for the redistribution of the optimal forces.

Regularization. We use $\mathcal{L}_{\mathrm{reg}}$ for the regularization terms in both optimization stages, with stage-specific targets and active terms. Per-frame residuals use trapezoidal temporal weights Δt over the relevant keyframes or trajectory frames. Positions and rotations are expressed in meters and radians, respectively.

The MANO palm frame is centered at the mean position of the wrist and tracked finger-base joints. Its x-axis points from the little-finger base to the index-finger base, using the ring-finger base when the little finger is absent; its z-axis is perpendicular to the x-axis and the wrist-to-mean-fingertip direction. Palm-position tracking has weight 10, while the rotation residual $\log((\mathbf{R}^*)^\top \mathbf{R})$ has weight 1, where $\mathbf{R}^*$ is the target orientation. At keyframe k, the target is the MANO palm pose and both weights are multiplied by $1 - 0.99c_k$, where $c_k = \max_i \alpha_{t_k,i}$.

Hand-shape residuals compare unit phalanx directions expressed in the respective robot and MANO palm frames. Each finger has weight $0.5(1-\alpha_f)$, where α_f is its selected contact's blending weight and is zero for fingers without stable contacts. The contact-normal residual is $r_i(\mathbf{n}_{\mathrm{robot}} + \mathbf{n}_{\mathrm{object}})$ with weight $100\alpha_{t,i}$, encouraging opposing contact normals. Here, r_i is the RMS radius of the fingertip surface, converting the normal discrepancy to a positional scale.

For temporal regularization, let $\mathbf{z}$ stack the palm position, the palm rotation vector relative to the seed orientation, and the finger joint angles. We penalize

$$\tau^2 \int \|\dot{\mathbf{z}}\|^2_{\mathbf{W}}\, dt + \tau^4 \int \|\ddot{\mathbf{z}}\|^2_{\mathbf{WM}}\, dt, \qquad \tau = 0.1\,\mathrm{s},$$

using nonuniform finite differences. The diagonal entries of $\mathbf{W}$ are 100 for palm translation and 1 for palm rotation and finger joints. The corresponding entries of $\mathbf{M}$ are 32 for palm coordinates and 16 for finger joints. The same temporal operator is used over keyframes and over the full trajectory. Keyframe optimization additionally includes the proximity prior $10^{-3}\|(q - q_{\mathrm{init}})/s\|_2^2$, where division is componentwise and the translation components are normalized by the object radius.

Solver. We solve Eq. 3 using Gauss–Newton SQP within a trust region. Violated hand–object constraints are added as they appear, elastic slack handles inconsistent linearizations, and a backtracking line search with second-order corrections controls the updates. Anchors are retracted onto the Gaussian surface at every trial, and geometric feasibility is checked with a tolerance of $0.1\,\mathrm{mm}$. We return the best feasible iterate, or the least-violating one otherwise.

D.3 Kinematic Trajectory Optimization

Contact Targets. For each finger and frame, we select the temporally nearest stable-contact interval. The distance is zero inside an interval and otherwise equals the distance to its nearest endpoint; ties are resolved in favor of the earlier interval. Both the anchor and robot contact query point are taken from the selected interval. The blending weight $\alpha_{t,i}$ is one within that interval and zero elsewhere. The contact-position fitting weight is $100(1 + 99\alpha_{t,i})$. Fingers without stable-contact intervals instead track the MANO fingertips with weight 1 in the position-fitting term.

Full-Trajectory Regularization. In Eq. 4, $\mathcal{L}_{\mathrm{reg}}$ applies the palm, hand-shape, and contact-normal penalties at every frame, together with the temporal operator over the full trajectory. The palm target is the MANO palm pose composed with spline-interpolated keyframe palm offsets. The keyframe proximity prior is not used in this stage.

Initialization and Solver. The trajectory is initialized by interpolating the keyframes; frames without demonstrated contact, such as approach and release, instead follow the MANO motion while matching every keyframe exactly. Keyframes that violate the collision constraints are released, and the remaining keyframes, indexed by $\mathcal{K}_{\mathrm{f}}$, are fixed. After repairing the initial trajectory to be collision-free at every frame, we run up to 40 Gauss–Newton SQP iterations with a trust region

and an Armijo line search. Collision checks are performed at every frame and, where applicable, at 16 substeps between consecutive frames. Substep checks are skipped for intervals with residual between-frame collisions after repair, which are reported, and for intervals between adjacent fixed keyframes. Trial updates are accepted only if they pass all frame-wise checks and all applicable substep checks, maintaining frame-wise collision feasibility. For DexYCB, the hand and object are shifted vertically together so that the object rests on the table, and the object is then settled once in simulation. The rigid transformation from the pre-settling to the settled object pose is then applied uniformly to the entire hand–object trajectory.

E Force-Aware Optimization Details

We optimize position-control knots $\mathbf{U} = \mathbf{U}^{\mathrm{ref}} + \Delta\mathbf{U}$, which are linearly interpolated over the L state-transition steps in Eq. (5). The tracking residual is $\mathbf{e}_m = \mathrm{col}(q_m^{\mathrm{robot}} - q_m^{\mathrm{robot,ref}}, \mathbf{p}_m^{\mathrm{object}} - \mathbf{p}_m^{\mathrm{object,ref}}, \vartheta_m)$, where ϑ_m is the relative object rotation angle. Let η_t equal one in the first optimization window and zero otherwise. Subject to the simulated dynamics, the complete objective is

$$\begin{aligned}\min_{\mathbf{U},\mathbf{z}} \quad J_t = \|\mathbf{e}_{t+L}\|^2_{\mathbf{Q}_{\mathrm{f}}} + \sum_{m=t+1}^{t+L-1} \|\mathbf{e}_m\|^2_{\mathbf{Q}} \\ + \sum_{m=t+1}^{t+L} (\ell_{\mathrm{a},m} + \ell_{\mathrm{d},m}) + \sum_{m=t}^{t+L} \ell_{\mathrm{col},m} \\ + \mathcal{R}(\mathbf{U},\mathbf{x}) + \eta_t \lambda_{\mathrm{init}} \|\mathbf{b} \odot \tanh(\mathbf{z})\|_2^2.\end{aligned} \tag{10}$$

The initial hand configuration is $q_t^{\mathrm{robot}} = \bar{q}_t^{\mathrm{robot}} + \eta_t \mathbf{b} \odot \tanh(\mathbf{z})$, where $\bar{q}_t^{\mathrm{robot}}$ is the supplied hand configuration and $\mathbf{b}$ specifies componentwise correction bounds. The variable $\mathbf{z}$ is optimized only in the first window; the remaining initial-state components are fixed. Subsequent windows start from the current executed state. Terminal tracking replaces the final intermediate tracking term. All weights absorb the corresponding unit conversions and component normalization.

For correspondence $k = (i, \mathcal{I})$, define $\mathbf{r}_{k,m} = \mathbf{P}_k^{\mathrm{robot}}(q_m^{\mathrm{robot}}) - \mathbf{T}_m^{\mathrm{object}} \hat{\mathbf{a}}_k$, where the robot material point is evaluated by forward kinematics and the refined anchor is transformed by the simulated object pose. The contact and collision terms are

$$\begin{aligned}\ell_{\mathrm{a},m} &= \lambda_{\mathrm{a}} \sum_k w_{k,m}^{\mathrm{a}} \gamma_{k,m} \|\mathbf{r}_{k,m}\|_2^2, \\ \ell_{\mathrm{d},m} &= \lambda_{\mathrm{d}} \sum_k w_{k,m}^{\mathrm{d}} [\phi_{k,m} - \phi_\star]_+^2, \\ \ell_{\mathrm{col},m} &= \sum_b \left(\lambda_{\mathrm{s}} [-\phi_{b,m} - d_{\mathrm{s}}]_+^2 + \lambda_{\mathrm{p}} [-\phi_{b,m} - d_{\mathrm{p}}]_+^2\right) \\ &\quad + \lambda_{\mathrm{f}} \sum_{(a,b)\in\mathcal{P}} [\epsilon_{\mathrm{f}} - d_{ab,m}]_+^2.\end{aligned} \tag{11}$$

Here, $[z]_+ = \max(z, 0)$, and $w_{k,m}^{\mathrm{a}}$ and $w_{k,m}^{\mathrm{d}}$ incorporate guidance activation, confidence, and contact-onset ramps. The distance-dependent anchor weight is $\gamma_{k,m} = \sigma((\|\mathbf{r}_{k,m}\|_2 - d_0)/\tau)$, where σ is the sigmoid function; it is recomputed during forward evaluation and held fixed during the corresponding backward pass. The quantity $\phi_{k,m}$ is the simulator collision distance for the hand collision output associated with correspondence k, rather than the material-point anchor distance, and $\phi_\star < 0$ specifies the target contact compression. The penetration term covers hand–object collision outputs, with thresholds $d_{\mathrm{s}} = -\phi_\star$ and $d_{\mathrm{p}} > d_{\mathrm{s}}$. The set $\mathcal{P}$ contains selected inter-finger capsule pairs, $d_{ab,m}$ denotes their surface separation, and ϵ_{f} is the desired margin. These are soft penalties. Their weights follow the selected control profile and may be zero when a term is disabled.

The smoothness regularizer penalizes control-knot differences and simulated hand velocities and accelerations:

$$\begin{aligned}\mathcal{R} &= \sum_{k=1}^{K} \left(\|\mathbf{v}_k^u\|_{\mathbf{S}_1}^2 + \|\mathbf{a}_k^u\|_{\mathbf{S}_2}^2 \right) + \sum_{m=t}^{t+L-1} \left(\|\mathbf{v}_m^{\text{hand},+}\|_{\mathbf{V}}^2 + \|\mathbf{a}_m^{\text{hand}}\|_{\mathbf{A}}^2 \right), \\ \mathbf{v}_k^u &= \frac{\mathbf{U}_k - \mathbf{U}_{k-1}}{\Delta_k}, \\ \mathbf{a}_k^u &= \frac{2(\mathbf{v}_k^u - \mathbf{v}_{k-1}^u)}{\Delta_k + \Delta_{k-1}}.\end{aligned} \tag{12}$$

Here, K is the number of control knots, Δ_k is the corresponding time interval, and executed-control history supplies the boundary values. The quantities $\mathbf{v}_m^{\text{hand},+}$ and $\mathbf{a}_m^{\text{hand}}$ are the post-step hand velocity and the hand acceleration returned by simulation step m. Reference velocities are not tracked, and no absolute control-energy penalty is used. We minimize Eq. (10) using parallel Adam optimization from one warm start and $N-1$ perturbed candidates. Each candidate retains its best finite iterate and is evaluated with the execution contact model; the lowest-cost candidate without numerical or simulator failures is selected. Contact forces are computed within the dynamics, and simulator parameters remain fixed during each optimization.

F Dataset selection

We select 50 right-hand grasping sequences from DexYCB and 80 right-hand in-hand manipulation sequences from HOT3D. For HOT3D, we extract clips with consecutive annotations for both the right-hand MANO pose and the target object pose. We retain only clips longer than $1\,\text{s}$ in which the right hand contacts only the target object and the left hand is uninvolved throughout.

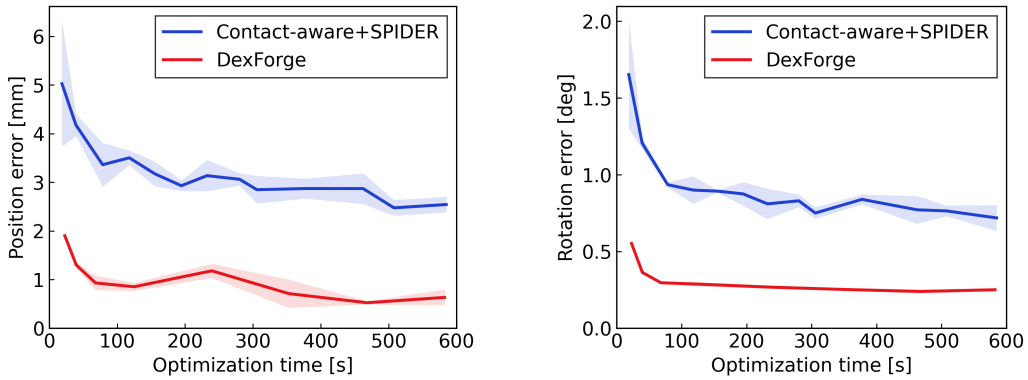
For ablation and system identification, we use five right-hand sequences from DexYCB subject 01: 20200709_142802 (sugar_box), 20200709_144357 (gelatin_box), 20200709_145401 (banana), 20200709_150613 (bowl), and 20200709_153322 (foam_brick). Each contains a single grasp-and-lift interaction and is used in full. We also select five HOT3D sequences: P0002_5d79f6b9 (dumbbell_5lb), P0003_90feb35f (whiteboard_eraser), P0003_8243ea4a (food_waffles), P0003_ebdc6ff7 (can_soup), and P0015_b0c5102b (birdhouse_toy), extracting clips using the same criteria above.

G Ablation Study Details

Starting from the full method, we first replace reconstructed hand–object trajectories with dataset annotations while keeping the Gaussian object models unchanged. We then disable contact refinement by skipping the joint optimization of contact anchors and key robot configurations. The raw anchors directly guide joint kinematic optimization, initialized by fitting robot keypoints to MANO keypoints. Finally, disabling contact guidance removes stable-contact extraction, contact refinement, and contact-guidance objectives from both retargeting stages. All variants retain joint kinematic optimization with collision constraints and subsequent dynamics optimization. We use the five DexYCB and five HOT3D sequences listed in Appendix F, evaluating all variants across seven robotic hands in ComFree-GS. All variants share optimization hyperparameters and use the success criteria defined in Sec. 4.2.1. For each dataset and hand, we compute the success rate over the five sequences for each seed and report the average across five seeds in Table 1.

H Dynamics Retargeting Details and Additional Results

Both methods use warm-started MPC and interpolate control nodes to produce commands at $100\,\text{Hz}$. The horizon H is measured in control intervals, so the prediction-window duration equals H times the control-node spacing. SPIDER retains its default configuration: a control-node spacing of $0.4\,\text{s}$, $H = 4$, a parallel batch size of 1024, and at most 16 optimization iterations. Our method uses a control-node spacing of $0.06\,\text{s}$, $H = 5$, a parallel batch size of 64, and at most 40 iterations. The resulting prediction windows are $1.6\,\text{s}$ and $0.3\,\text{s}$, respectively. We retain method-specific temporal configurations because imposing the same prediction window can disadvantage either optimizer.



(a) Object position error

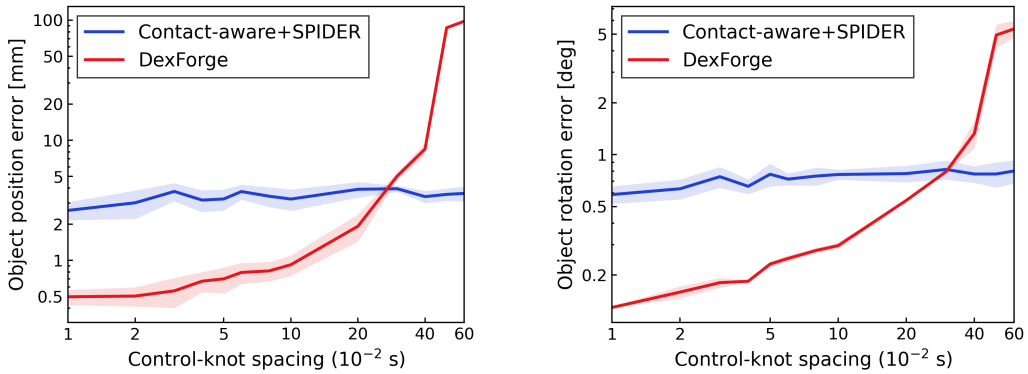
(b) Object rotation error

Figure 9: Performance–runtime comparison between DexForge and Contact-aware+SPIDER. Object position (a) and rotation (b) tracking errors are plotted against optimization time, with runtime varied through the optimization iteration limit of each method. Curves show the mean over three runs and shaded regions indicate the standard deviation. Under the same runtime budget, DexForge consistently achieves lower tracking errors.

To examine the performance–runtime trade-off, we compare DexForge and Contact-aware+SPIDER on a fixed subset of ablation sequences successfully retargeted by both methods across all tested budgets. After object drops, smaller tracking errors do not necessarily indicate better retargeting; failures are instead reflected in the success rates reported above. For each method, we vary only the optimization iteration limit and measure total optimization time, disabling SPIDER's early stopping to control the budget through the iteration limit. All other MPC settings remain at their respective defaults. Both methods use identical kinematic references, stable-contact guidance, and simulation environment (ComFree-GS), running sequentially on a single RTX 4090 D. Each setting is repeated three times. Figure 9 reports the mean and standard deviation of object position and rotation tracking errors at different runtimes. On this shared successful subset, DexForge achieves lower tracking errors under the same runtime budget, demonstrating more precise trajectory refinement.

We further examine sensitivity to control-knot spacing by comparing DexForge and Contact-aware+SPIDER with $H = 4$ control intervals, choosing method-specific iteration limits at each spacing to obtain optimization runtimes close to $200\,\mathrm{s}$. We sweep the knot spacing over $\{0.01, 0.02, \ldots, 0.10, 0.20, 0.30, 0.40, 0.60\}\,\mathrm{s}$, varying the prediction-window duration accordingly while keeping the interpolated command rate at $100\,\mathrm{Hz}$. As shown in Figure 10, on sequences completed by both methods, the tracking errors of Contact-aware+SPIDER are largely insensitive to knot spacing, remaining within approximately 2.5–$4\,\mathrm{mm}$ and 0.6–0.8° across the sweep, so its default spacing does not disadvantage its tracking accuracy. In contrast, DexForge achieves substantially lower errors at fine spacings, but its advantage diminishes and eventually reverses beyond approximately $0.3\,\mathrm{s}$, consistent with greater difficulty in gradient-based optimization over longer rollouts. This motivates the fine knot spacing used by DexForge.

This analysis, however, only covers sequences that both methods already complete and does not show whether finer spacing could reduce SPIDER's failures on HOT3D, where in-hand manipulation may involve higher-frequency motions. We therefore evaluate Contact-aware+SPIDER with $0.01\,\mathrm{s}$ knot spacing on the full HOT3D set. Its success rate increases only marginally, from 0.6% to $6.0 \pm 3.7\%$, remaining far below that of DexForge ($34.9 \pm 2.1\%$). Moreover, even with accurate physical parameters, its open-loop transfer success to MuJoCo-CPU on the ablation subset is only $3.1 \pm 2.9\%$, below the $15.4 \pm 2.6\%$ achieved with the default spacing after identification (Table 4). This suggests that the resulting high-frequency controls overfit the source simulator and reduce transfer robustness. DexForge mitigates this issue through its coarser $0.06\,\mathrm{s}$ spacing and explicit control-smoothness regularization (Eq. 12). We therefore retain each method's default temporal configuration in the main evaluation.



(a) Object position error

(b) Object rotation error

Figure 10: Sensitivity to control-knot spacing under a matched 200 s optimization budget and $H = 4$ control intervals. Errors are computed on a fixed subset of ablation sequences successfully retargeted by both methods under their default configurations; at other spacings, all runs on this subset are included regardless of success. Panels show object position (a) and rotation (b) tracking errors. Curves show mean errors over 3 runs, with shaded regions indicating ± 1 standard deviation. Contact-aware+SPIDER is largely insensitive to knot spacing, whereas DexForge achieves lower errors at finer spacings but degrades as the spacing and prediction-window duration increase, consistent with greater difficulty in gradient propagation through longer rollouts.

I Sim-to-Sim Transfer Details

We evaluate open-loop transfer from ComFree-GS to MuJoCo-CPU, where target scenes use mesh-based object models with known or estimated physical properties. In ComFree-GS, all objects are initialized with a mass of $0.2\,\mathrm{kg}$, a contact friction coefficient of 1.0, a distance offset of zero, and an isotropic inertia of $0.001\,\mathrm{kg\,m^2}$. Although the formulation in Section 3.4 supports identifying the full parameter set θ, in these experiments we identify only the mass, friction coefficient, and distance offset, keeping the inertia fixed. For each sequence, we first optimize a complete retargeted control trajectory in ComFree-GS and replay a contact-rich segment in MuJoCo-CPU to collect target observations. Parameters are identified separately for each sequence by minimizing discrepancies between the simulated and recorded target trajectories under the same fixed controls. All replays use target-position commands at $100\,\mathrm{Hz}$ with matched initial states. Gradient-only identification uses Adam for 100 iterations with a learning rate of 10^{-3} and a gradient-clipping threshold of 10. Sampling-only identification uses CEM for 100 iterations with a population of 100, an elite fraction of 10%, an initial standard deviation of 1.0 in the raw parameter space, a minimum standard deviation of 0.01, and a smoothing coefficient of 0.7. TSMC uses 64 particles, a prior standard deviation of 0.5, a temperature parameter of 1.0, an ESS ratio of 0.7, and at most eight tempering stages with systematic resampling. Each stage performs four MALA steps with a step size of 10^{-4}. Robust filtering with a threshold of $z = 3.5$ is then applied before constructing an energy-weighted parameter distribution. After identification, we update the source parameters and rerun dynamics retargeting in ComFree-GS, then replay the resulting control trajectories open-loop in MuJoCo-CPU. Evaluation uses the same subsets as the ablation study across seven robotic hands and five seeds, with the transfer success criteria specified in the main text.

J Real-World Open-Loop Replay Details and Results

We replay optimized wrist and finger references on a Unitree G1 equipped with a LEAP Hand v1. The hand is mounted on the right wrist using a custom adapter, whose wrist-to-hand transformation is incorporated into the forward-kinematic model. Before replay, we place the object at the origin of a table-mounted motion-capture rigid-body frame with a predefined orientation to establish its initial pose. We then transform the precomputed wrist trajectory accordingly, select a reachable starting point, and move the robot to the corresponding configuration. A $100\,\mathrm{Hz}$ process uses Unitree's arm

IK solver to generate arm joint-position targets and sends them together with the precomputed finger-position targets. Motion-capture torso poses compensate for body motion to preserve the intended world-frame wrist trajectory. A separate $50\,\mathrm{Hz}$ process runs NVIDIA's Decoupled WBC to maintain whole-body balance. Replay is open-loop with respect to object interaction: the precomputed manipulation references are not updated using object-state feedback, while torso compensation and balance control remain feedback-driven.

Results. We conduct 10 trials for each of the five objects, resetting the object to its predefined initial pose before each trial. Since object poses are not tracked during execution, a trial is considered successful if the object is stably grasped and held for $5\,\mathrm{s}$ without dropping. This criterion measures executability rather than tracking accuracy. The success counts are 10/10 for the gelatin box, 8/10 for the pudding box, 6/10 for the wood block, 10/10 for the power drill, and 6/10 for the tomato soup can, yielding an overall success rate of 80% (40/50). Videos are available on the project page.