

Joint Flow Trajectory Optimization For Feasible Robot Motion Generation from Video Demonstrations

Xiaoxiang Dong^{1,2}

Matthew Johnson-Roberson^{1,2}

Weiming Zhi^{3,4,1,*}

Abstract— Learning from human video demonstrations offers a scalable alternative to teleoperation or kinesthetic teaching, but poses challenges for robot manipulators due to embodiment differences and joint feasibility constraints. We address this problem by proposing the Joint Flow Trajectory Optimization (JFTO) framework for grasp pose generation and object trajectory imitation under the video-based Learning-from-Demonstration (LfD) paradigm. Rather than directly imitating human hand motions, our method treats demonstrations as object-centric guides, balancing three objectives: (i) selecting a feasible grasp pose, (ii) generating object trajectories consistent with demonstrated motions, and (iii) ensuring collision-free execution within robot kinematics. To capture the multimodal nature of demonstrations, we extend flow matching to $SE(3)$ for probabilistic modeling of object trajectories, enabling density-aware imitation that avoids mode collapse. The resulting optimization integrates grasp similarity, trajectory likelihood, and collision penalties into a unified differentiable objective. We validate our approach in both simulation and real-world experiments across diverse real-world manipulation tasks.

I. INTRODUCTION

Generating feasible motions is a key challenge in robotics. Traditional approaches address this by formulating and solving optimization problems with carefully designed costs and constraints [1], [2], [3]. An alternative line of work focuses on enabling robots to imitate humans through *Learning from Demonstration* (LfD) [4], [5]. Historically, this has required collecting demonstrations through teleoperation or precise kinesthetic teaching. More recently, researchers have attempted to leverage videos of human interaction with objects as demonstrations [6], [7]. Although promising, this paradigm introduces challenges: Human arms and robot manipulators differ significantly in morphology, and direct imitation can cause robots to violate joint constraints when tracking human motions. All demonstrations in this paper refer to such human video demonstrations.

In this paper, we tackle the problem of generating grasp poses and motion trajectories for robot manipulators under the video-based LfD paradigm, where the tracked human motion may often not be kinesthetically feasible for a robot manipulator to execute. Our goal is to identify feasible grasp configurations and ensure that grasped objects move consistently with human video demonstrations. Concretely, given videos of humans interacting with objects, we propose the Joint Flow Trajectory Optimization (JFTO) framework,

*email: Weiming.Zhi@sydney.edu.au.

¹ College of Connected Computing, Vanderbilt University, Nashville, TN, USA

² Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA

³ School of Computer Science, The University of Sydney, Australia.

⁴ Australian Centre for Robotics, The University of Sydney, Australia.

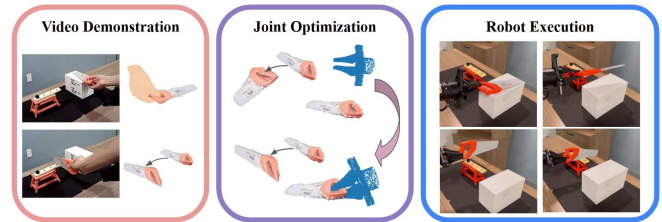


Fig. 1: We address generating feasible grasp poses and motion trajectories from video-based LfD, ensuring that robot grasps and object motions remain consistent with human demonstrations despite constraints. We propose the Joint Flow Trajectory Optimization (JFTO) framework, which jointly optimizes over grasp feasibility and trajectory imitation, producing a gripper pose and motion plan that closely match the demonstration while remaining executable.

as shown in Fig. 1, with three desiderata: (1) identifying a feasible grasp pose; (2) generating trajectories of the grasped object that align with object poses in the videos; (3) ensuring downstream robot motions remain collision-free and within joint limits. To this end, our formulation treats human video demonstrations as object-centric guides rather than strict motion references. By focusing on how objects are manipulated, rather than how the human hand is configured, we allow robots to synthesize grasps and trajectories that are consistent with observed outcomes, while respecting the robot’s own embodiment. Uncertainty is critical to robotics tasks [8], [9], [10]. We take a probabilistic approach and model the distribution of object pose trajectories extracted from videos as a generative model *Flow Matching* [11] over the Special Euclidean Group $SE(3)$. Here, as the grasps and motions of the human from the video-based demonstrations may not be valid for robot execution, our goal is not only to ensure grasp and execution feasibility, but also to mimic the motion patterns of human demonstrations in the videos.

Concretely, the technical contributions of this work are threefold:

- We introduce the Joint Flow Trajectory Optimization (JFTO) framework that integrates grasp similarity, object trajectory likelihood, and collision avoidance into a unified differentiable objective for video-based LfD.
- We extend flow matching to $SE(3)$ for probabilistic modeling of demonstrated object trajectories, enabling density-aware imitation that preserves multi-modality.
- We demonstrate the effectiveness of JFTO in both simulation and real-world experiments on diverse manipulation tasks, showing that joint optimization achieves higher fidelity to demonstrations compared to sequential baselines.

II. RELATED WORK

This paper addresses joint generation of grasp poses and motion trajectories from video-based LfD, ensuring manipulator feasibility while remaining consistent with human demonstrations, relating to work in *robot learning from videos* and *grasp pose generation*.

Robot Learning from Videos: Human videos naturally capture interaction, motion, and affordances, providing rich supervision for robot learning. Recent work leverages such data to build transferable visuomotor skills, from single demonstrations to large-scale web video distillation. For example, *Human2Sim2Robot* [12] mitigates the human–robot embodiment gap by extracting object pose trajectories and premanipulation hand configurations from a single RGB-D video, using them to define object-centric rewards and initialize reinforcement learning (RL) in simulation, enabling zero-shot sim-to-real transfer of dexterous policies. At the other end, *ZeroMimic* [13] distills manipulation skills from large-scale in-the-wild datasets such as EpicKitchens, producing image goal-conditioned policies for tasks like opening, pouring, and stirring that generalize across embodiments. Other approaches emphasize one-shot or safety-aware imitation [6], [7], [14], or synthetic demonstrations for robustness and efficiency [15], [16]. Earlier work also explored learning from unstructured human videos via scalable datasets [17], [18] or embodied dexterous platforms [19]. Overall, relative to other interfaces for demonstration collection [20], learning from videos allows for larger scale data collection.

Grasp Pose Generation: The Grasp Pose Generator (GPG), first introduced in the Grasp Pose Detection framework [21], samples candidate grasps from point cloud geometry. Candidates are filtered with heuristics (antipodal checks, collision tests) before evaluation by analytic or learned models. GPG supports large-scale datasets by labeling analytic positives (antipodal, force closure) and negatives, a paradigm popularized by Gualtieri et al. [22] and widely adopted in training pipelines [23], [24]. Methods which explore different object representations [25] upstream to grasping have been explored [26]. Benchmarks like *GraspNet-1Billion* [27] follow this approach, providing more than a billion simulated grasps. Recent refinements, such as *Graspness Discovery* [28], introduce geometric quality measures to faster pruning and higher sample quality. GPG also serves as a sampler in modern data-driven pipelines. *GtG 2.0* [29] combines GPG with graph neural networks to score 7-DoF grasps.

III. PRELIMINARIES: FLOW MATCHING

Flow matching is a generative modeling approach that estimates continuous data distributions by learning vector fields that transform samples from a known prior distribution to a target distribution by integration along an Ordinary Differential Equation (ODE). This method provides a fast and exact generative process and avoids many of the complexities associated with likelihood-based models or score-based models [30], [2], [31]. Flow matching speeds up previous normalizing flow methods [32] and their applications in robotics [33], [34].

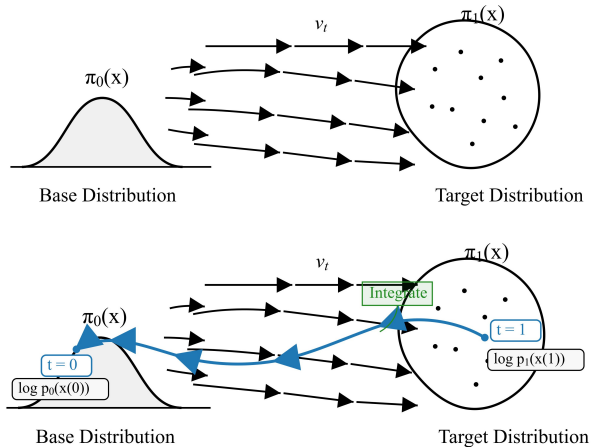


Fig. 2: Flow Matching learns a time-dependent velocity field v_t from based distribution $\pi_0(x)$ to target distribution $\pi_1(x)$, while integrating backward from a sample in $\pi_1(x)$ through v_t to $\pi_0(x)$ derives the log-density of the sample.

Let $\pi_0(x)$ denote a base distribution over $x \in \mathbf{R}$, for example standard Gaussian $\mathcal{N}(0, 1)$, and let $\pi_1(x)$ denote the target distribution. Flow matching seeks to learn a time-dependent vector field $v_t : \mathbf{R}^d \rightarrow \mathbf{R}^d$ that defines a flow via the ODE:

$$\frac{d\mathbf{x}}{dt} = v_t(\mathbf{x}), \quad \mathbf{x}_0 \sim \pi_0, \quad (1)$$

such that the solution $\mathbf{x}(1)$ approximates the distribution π_1 , as shown in Fig. 2.

To train this vector field, one sample $(\mathbf{x}_0, \mathbf{x}_1) \sim (\pi_0, \pi_1)$ and defines an interpolation:

$$\mathbf{x}_t = \gamma(t; \mathbf{x}_0, \mathbf{x}_1), \quad (2)$$

The target velocity field $\hat{v}_t(\mathbf{x}_t)$ is then defined as the time derivative of the interpolation curve:

$$\hat{v}_t(\mathbf{x}_t) = \frac{d}{dt} \gamma(t; \mathbf{x}_0, \mathbf{x}_1), \quad (3)$$

The model is trained by minimizing the expected squared error between the learned field $v_t(\mathbf{x}_t)$ and the target field $\hat{v}_t(\mathbf{x}_t)$:

$$\mathbb{E}_{t \sim \mathcal{U}(0,1)} \left[\|v_t(\mathbf{x}_t) - \hat{v}_t(\mathbf{x}_t)\|^2 \right], \quad (4)$$

Once trained, the learned vector field can be used in two ways: 1) generate new samples from π_1 by integrating the ODE from \mathbf{x}_0 to \mathbf{x}_1 , 2) evaluate the log-density of a sample $\mathbf{x}(t)$ evolving under the flow using the instantaneous change of variables formula:

$$\frac{d}{dt} \log p_t(x(t)) = -\nabla \cdot v_t(x(t)), \quad (5)$$

which, when integrated backward from $t = 1$ to $t = 0$, yields:

$$\log p_1(x(1)) = \log p_0(x(0)) - \int_0^1 \nabla \cdot v_t(x(t)) dt, \quad (6)$$

as illustrated in Fig. 2. In Section IV, we represent the probabilities in the motion trajectory distributions with flow models.

IV. JOINT FLOW TRAJECTORY OPTIMIZATION

Here, we propose Joint Flow Trajectory Optimization (JFTO), a framework that *jointly* optimizes the robot’s initial grasp and its subsequent motion so that the resulting object

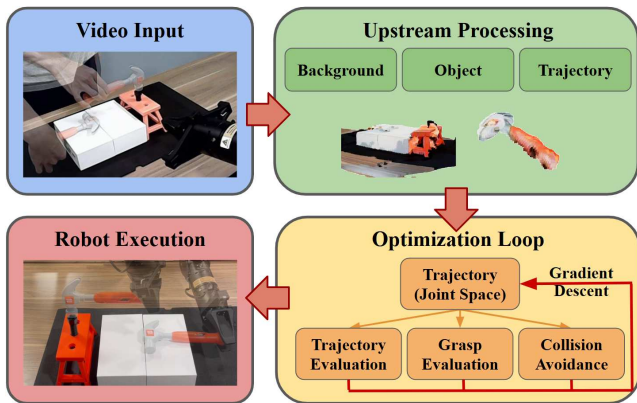


Fig. 3: We present an overview of the Joint Flow Trajectory Optimization (JFTO) framework. Human demonstration videos are processed to extract the grasp pose and object trajectory. These are then used to jointly optimize a feasible execution trajectory in the robot’s joint space, and then executed on the robot.

trajectory closely imitates a human video demonstration, while still remaining kinematically feasible and collision-free. Unlike sequential approaches that first fix a grasp and then attempt to optimize the trajectory, the JFTO formulation reasons about the grasp and downstream trajectory simultaneously. This allows the robot to choose grasps that remain feasible throughout the entire motion. An overview of JFTO is shown in Fig. 3.

A. Problem Formulation

Let $\xi = \{q_t\}_{t=0}^T$ denote the joint states of the robot in the T time steps. A rollout is scored by three components:

- $S_T(\xi)$: how well the induced object trajectory matches the demonstrated trajectory,
- $S_G(q_0)$: how well the initial grasp resembles a feasible human-like grasp, and
- $S_C(\xi)$: how safe the rollout is with respect to joint limits and collisions.

The optimization problem is formulated as:

$$q^* = \arg \max_{\xi} \left[\alpha S_T(\xi) + \beta S_G(q_0) + \gamma S_C(\xi) \right], \quad (7)$$

where α, β, γ are weights that balance the importance of trajectory imitation, grasp quality, and safety. Note that q_0 is a component of ξ .

This formulation emphasizes that the goal is not to directly replicate human hand motions, which may be kinematically infeasible, but rather to optimize for a joint trajectory that reproduces the demonstrated object motion and respects the constraints of the robot’s embodiment.

B. Data Collection and Processing

The proposed optimization framework assumes access to object poses obtained from human–object interactions. To this end, we introduce a system that extracts object trajectories directly from raw demonstration videos. While we focus on this video-based pipeline, the framework is agnostic to the source of demonstrations, and alternative methods for acquiring object pose information could be substituted without loss of generality.

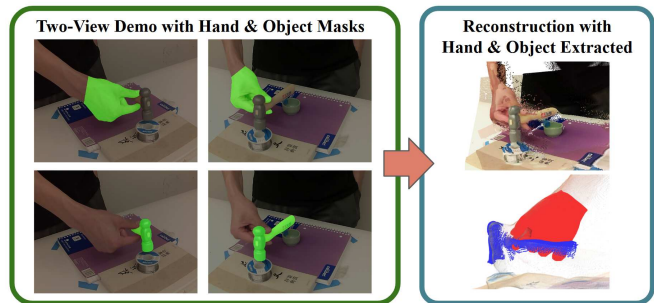


Fig. 4: From two views of images with hand and object segmentation, we reconstruct the scene with 3D foundation models and use the segmentation masks to determine points corresponding to hand and object.

For each task, we record synchronized image sequences of a human demonstration using two calibrated cameras. Each frame pair is passed to a 3D foundation model, in our case DUS_t3R [35], to reconstruct a dense 3D point cloud of the scene. Alternative models, such as MON_t3R [36] can also be used. Since DUS_t3R operates on an arbitrary internal scale, outlined in [37], [38], we recover the metric scale by triangulating the corresponding feature points between the image pairs and aligning their reconstructed 3D positions with the known physical camera baseline. This produces trajectories expressed in real-world coordinates.

Upstream Processing: To isolate the relevant elements of the scene, we apply SAM2 [39] to segment both the human hand and the manipulated object in each frame. The segmentation masks allow us to separate the object and hand point clouds from the static background, as illustrated in Fig. 4. We further estimate precise hand poses with HaMeR [40], repeating this process across key frames to obtain temporally consistent object–hand trajectories. An example visualization is shown in Fig. 7. To continuously track object motion, consecutive object point clouds are aligned using Iterative Closest Point (ICP) [41], resulting in a sequence of estimated object poses $\{\hat{x}_t\}$ that describe its trajectory during the demonstration. In parallel, the background points across all frames are fused into a single static point cloud, which is later converted into a distance field for efficient collision checking.

Connection to Downstream Optimization: From each demonstration we extract the following structured outputs: (1) the object trajectory $\{\hat{x}_t\}$, which drives trajectory imitation (S_T); (2) the initial object pose \hat{x}_0 , which anchors grasp evaluation (S_G); and (3) a static background model, which supports collision scoring (S_C). These components enable the downstream optimization of grasp and trajectory in a differentiable manner.

C. Trajectory Quality Evaluation

Flow Matching as Pose Likelihood Estimation: From n human demonstrations, each trajectory of T waypoints in $\mathbf{SE}(3)$, we collect pose samples $\mathcal{Q}_t = \{x_t^{(i)}\}_{i=1}^n$ at every time step t . We treat \mathcal{Q}_t as drawn from an underlying distribution $p_t(x)$ over $\mathbf{SE}(3)$. Our goal is to construct a

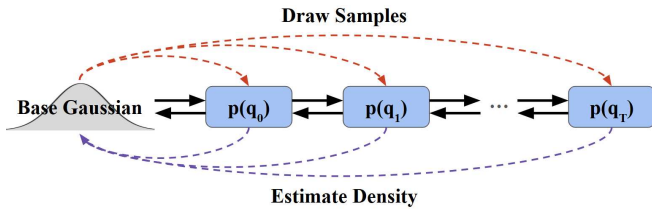


Fig. 5: We treat each time step of the demonstration as a distribution over object poses in $\mathbf{SE}(3)$. Flow matching learns velocity fields that transform a base Gaussian into these distributions. This enables both sampling of realistic poses (forward integration) and computation of their log-likelihoods via divergence.

density estimator

$$f_{den} : \mathbf{SE}(3) \times \{0, \dots, T\} \rightarrow \mathbb{R},$$

such that $f_{den}(x, t) \approx \log p_t(x)$ provides the log-likelihood of a candidate pose x at time t .

We use *flow matching* to model these time-dependent pose distributions, as illustrated in Fig. 5. Let $v_t : \mathbf{SE}(3) \rightarrow \mathbb{R}^6$ denote a learned velocity field, where the 6D output corresponds to the translational and rotational directions in the tangent space of $\mathbf{SE}(3)$. The field defines an ODE

$$\frac{dx_\tau}{d\tau} = v_\tau(x_\tau), \quad x_0 \sim \pi_0, \quad (8)$$

where π_0 is a base Gaussian distribution over poses. Integrating this ODE from $\tau = 0$ to $\tau = 1$ pushes forward π_0 to approximate the demonstration distribution $p_\tau(x)$ at each step. The associated log-density can be computed using the instantaneous change-of-variables formula,

$$\log p_\tau(x) = \log \pi_0(x_0) - \int_0^\tau \nabla \cdot v_\tau(x_\tau) d\tau. \quad (9)$$

Thus, the density estimator is defined as

$$f_{den}(x) = \exp(\log p_\tau(x)), \quad (10)$$

which is tractable because both the ODE and its divergence are differentiable. Intuitively, $f_{den}(x)$ assigns high scores to poses that align with the demonstrated motion distribution, and low scores to outliers.

Trajectory Rollout Evaluation: To evaluate how well a candidate joint trajectory imitates the demonstrations, we begin with a sequence of joint states $\xi = \{q_t\}_{t=0}^T$ in the d -DoF robot joint space $\mathcal{J} \subset \mathbb{R}^d$. Each state is restricted to be within joint limits and is mapped through forward kinematics F to obtain the corresponding pose of the end effector $r(q)_t \in \mathbf{SE}(3)$. This ensures that the trajectories are always kinematically feasible for execution. Given the initial pose of the object \hat{x}_0 recovered from the video, the relative transform between the gripper and the object at the first timestep is defined as

$$T_{go} = \hat{x}_0^{-1} r(q_0). \quad (11)$$

This transform anchors the grasp and is assumed to be fixed throughout execution, meaning that subsequent gripper motions directly induce object motions. The resulting object trajectory is then expressed as $x(q_0) = r(q_0) T_{go}^{-1}$ for $t = 0, \dots, T$. Each induced object pose is scored using the density estimator $f_{den}(x(q_t))$, and the rollout receives an overall trajectory score equal to the average log-likelihood

across all time steps:

$$S_T(q) = \frac{1}{T} \sum_{t=0}^T f_{den}(x(q_t)). \quad (12)$$

In effect, this procedure asks whether the object trajectory implied by the robot motion falls within regions of high probability according to human demonstrations. Rollouts that closely reproduce demonstrated motion patterns earn high scores, while those that deviate substantially are penalized.

Intuitively, the density estimator f_{den} models which poses are likely according to demonstrations. A rollout receives a high score if its induced object trajectory lies in regions of high demonstration density, thereby imitating not just the endpoints but the characteristic motion itself.

D. Grasp Feasibility and Similarity Evaluations

1) *Grasp Feasibility Check:* Successful execution of the object trajectory requires the gripper to establish and maintain a physically feasible grasp. To model grasp feasibility, we adopt a two-stage approach: (i) generate candidate positive grasps using a geometry-driven sampler, and (ii) train a learned classifier to distinguish these from infeasible alternatives. We first employ a Grasp Pose Generator (GPG) [21] to propose a set of analytically feasible grasps $P = \{p_i\}_{i=1}^N$ around the object point cloud. Each $p_i \in \mathbf{SE}(3)$ is guaranteed to satisfy local geometric conditions such as antipodal contact or force-closure heuristics and thus serves as a positive example.

To train a discriminative model, we construct two types of negative examples:

- *Hard negatives:* generated by perturbing valid grasps with small translational and rotational noise, e.g.,

$$p_{\text{hard}}^- = p_i \cdot \exp(\epsilon), \quad \epsilon \sim \mathcal{N}(0, \Sigma_{\text{small}}), \quad (13)$$

where $\exp(\cdot)$ maps a perturbation in the Lie algebra $\mathfrak{se}(3)$ to $\mathbf{SE}(3)$. These poses are close to feasible grasps, but violate stability constraints.

- *Soft negatives:* sampled uniformly from the vicinity of the object in $\mathbf{SE}(3)$, without regard to contact conditions. These provide a broad coverage of infeasible grasps:

$$p_{\text{soft}}^- \sim \mathcal{U}(\mathcal{N}_{\text{object}}) \subset \mathbf{SE}(3). \quad (14)$$

Each pose $p \in \mathbf{SE}(3)$ is then embedded in a feature vector using a positional encoding $E : \mathbf{SE}(3) \rightarrow \mathbb{R}^e$. Specifically, the pose is first represented in a 6D form (x, y, z, R_x, R_y, R_z) , where (x, y, z) denotes translation and (R_x, R_y, R_z) denotes the axis-angle parameterization of rotation. This vector is then passed through a Fourier encoding:

$$E(p) = [\sin(2^k \pi u), \cos(2^k \pi u)]_{k=0}^{K-1}, \quad u \in \mathbb{R}^6, \quad (15)$$

which maps each dimension of u to high-frequency features. The resulting embedding has dimension $e = 12K$.

We then train a classifier

$$f : \mathbb{R}^e \rightarrow \mathbb{R}, \quad (16)$$

using a contrastive learning objective. For a batch containing positives P^+ and negatives P^- , we minimize the binary cross-entropy loss. After training, the classifier produces a feasibility score

$$S_{\text{feas}}(p) = \sigma(f(E(p))), \quad (17)$$

which approximates the probability that pose p corresponds to a stable grasp. This learned feasibility check generalizes beyond the discrete set of grasps generated by GPG and provides differentiable gradients, making it suitable for joint optimization with trajectory imitation and collision avoidance.

2) *Grasp Distance and Evaluation:* A key challenge of the JFTO framework is balancing the feasibility and consistency of a candidate robot grasp with those observed in demonstrations. To quantify the similarity between robot and human poses, we define a weighted distance on $\mathbf{SE}(3)$ that accounts for both translation and orientation. For two poses $p_1, p_2 \in \mathbf{SE}(3)$, let $t(p) \in \mathbb{R}^3$ denote its translation and $\phi(p) \in \mathbb{R}^3$ the axis-angle representation of its rotation. The distance is

$$D_{\mathbf{SE}(3)}(p_1, p_2) = \gamma \|t(p_1) - t(p_2)\|_2 + (1 - \gamma) \|\phi(p_1) - \phi(p_2)\|_1, \quad (18)$$

where $\gamma \in [0, 1]$ balances the contributions of translation and orientation. The weight can be tuned by task; for example, orientation dominates in pouring tasks, while position is critical in pick-and-place tasks.

From the upstream pipeline, we extract a set of human grasps $H = \{h_i\}$, although not all are physically feasible due to differences between the human hand and the robot gripper. For a candidate robot grasp $p \in \mathbf{SE}(3)$, we measure similarity as the minimum distance from the demonstration set:

$$D_H(p) = \min_{h \in H} D_{\mathbf{SE}(3)}(p, h). \quad (19)$$

Given a robot joint configuration $q_0 \in \mathcal{J}$, we compute the corresponding gripper pose by forward kinematics,

$$p(q_0) = F(q_0) \in \mathbf{SE}(3). \quad (20)$$

This pose is then evaluated with two complementary criteria:

- 1) **Feasibility:** A classifier $f : \mathbb{R}^e \rightarrow \mathbb{R}$ predicts grasp validity. The input to f is a Fourier positional encoding $E : \mathbf{SE}(3) \rightarrow \mathbb{R}^e$ of the grasp pose. The output is squashed with a sigmoid σ , producing a probability-like score $\sigma(f(E(p(q_0))))$.
- 2) **Grasp Demonstration Similarity:** The distance $D_H(p(q_0))$, which is small when the grasp is close to the one observed in demonstrations.

The final grasp score is defined as

$$S_G(q_0) = \sigma(f(E(p(q_0)))) - \lambda D_H(p(q_0)), \quad (21)$$

where $\lambda > 0$ balances feasibility and similarity. High scores are assigned to grasp poses that are both executable (validated by the classifier) and close to human demonstrations (low D_H), encouraging the robot to select grasps that are robust and human-like.

E. Collision Avoidance

While grasping and trajectory imitation ensure that the robot behaves like the demonstrator, safety requires avoiding collisions with the environment. To model the static scene, we fuse the background point clouds $B \subset \mathbb{R}^3$ obtained from the video into a dense reconstruction. From this cloud, we train a distance function $dist_B : \mathbb{R}^3 \rightarrow \mathbb{R}$, which outputs the estimated distance from any query point to

Algorithm 1: Joint Flow Trajectory Optimization

- 1: Trajectory $\xi = \{q_t\}_{t=0}^T$, optimization weighting hyperparameters α, β, γ
 - 2: Set optimizer \leftarrow Adam(ξ)
 - 3: **for** $i = 1$ to N steps **do**
 - 4: Compute $S_T(\xi)$ (See Section IV-C)
 - 5: Compute $S_G(q_0)$ (See Section IV-D)
 - 6: Compute $S_C(\xi)$ (See Section IV-E)
 - 7: loss $\leftarrow -[\alpha S_T(\xi) + \beta S_G(q_0) + \gamma S_C(\xi)]$
 - 8: Minimise gradients of loss w.r.t. $\{q_t\}_{t=0}^T$
 - 9: Update ξ with optimizer
 - 10: **end for**
 - 11: **return** ξ .
-

the nearest obstacle surface. The differentiable field can be queried efficiently, making it suitable for gradient-based optimization.

For a given joint state $q_i \in \mathcal{J}$, we approximate the geometry of the robot arm by a finite set of surface points $Q(q_i) \subset \mathbb{R}^3$. These can be obtained either from mesh models sampled at fixed density or from bodypoints on the robot augmented with collision radii. A configuration q_i is considered in collision if there exists $p \in Q(q_i)$ with $dist_B(p) < \ell$, where ℓ is a small safety margin (e.g., 0.01 m). To penalize unsafe motion over a rollout, we compute a collision score

$$S_C(q) = \frac{1}{T} \sum_{t=0}^T \sum_{p \in Q(q_t)} \frac{1}{\ell} \max(0, \ell - dist_B(p)), \quad (22)$$

Here, $\max(0, \cdot)$ ensures that only points within the safety margin contribute, while scaling by $1/\ell$ normalizes the penalty. Furthermore, $S_C(q)$ remains close to zero for safe trajectories but increases as the robot approaches or intersects obstacles, creating strong gradients that steer optimization toward collision-free solutions. This formulation also allows “near-miss behaviors” (where the robot comes close to the environment) to be penalized less severely than actual interpenetration, encouraging motions that are safe yet not overly conservative. The outline of our approach is summarized in Algorithm 1. This algorithm can be parallelized on the GPU where we optimize a large batch of trajectories. This improves robustness against poor local minima. The candidate with the highest final score is selected.

V. EMPIRICAL EVALUATIONS

We evaluate the proposed *Joint Flow Trajectory Optimization* method in simulation and on a real robot. Our study asks: **(i)** Does joint optimization select grasps that are feasible and conducive to faithful object-trajectory imitation? **(ii)** How does joint optimization compare to a sequential pipeline that optimizes the grasp then trajectory? **(iii)** Does modeling demonstration *multi-modality* through flow matching improve imitation relative to distance-based objectives?

Hardware and Platform: Real-world trials are carried out on a *Agile-X Piper* 6-DoF arm with a parallel gripper. The limited DoF and small work envelope increase the likelihood of joint-limit and reachability conflicts, providing a challenging setting for grasp feasibility, collision avoidance, and faithful trajectory execution.

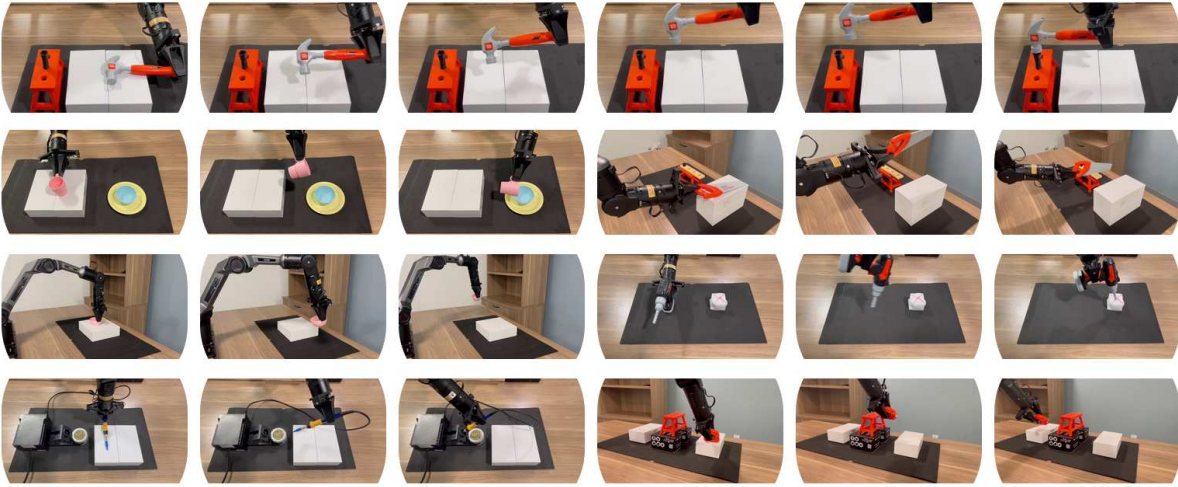


Fig. 6: We present the experimental results of the 7 real-world tasks. The first row corresponds to Task 1, second row first three corresponds to Task 2, last three corresponds to Task 3, third row first three corresponds to Task 4, last three corresponds to Task 5, last row first three corresponds to Task 6, last three corresponds to Task 7.

Tasks and Demonstrations: We construct a seven-task benchmark spanning tool use, pouring, cutting, drawing-like motions, and obstacle-aware transport:

Task 1 – Hammer Hitting Nail: pick up a hammer and strike a nail. *Task 2 – Cup Pouring Water:* grasp a cup and pour water into a bowl on the left. *Task 3 – Saw Cutting Wood:* lift a saw from the base, reorient upright and perform a cutting motion. *Task 4 – Drawing Figure 8:* trace a figure-eight with a held bowl. *Task 5 – Drilling Foam:* grasp a toy drill and drill into foam. *Task 6 – Solder Putting Back:* return a soldering tool to its station. *Task 7 – Shifting Meter Tape:* transport a tape across the workspace. We introduce additional obstacles in inference.

Human demonstrations are processed using the pipeline described in Section IV-B: stereo DUST3R [35] for 3D reconstruction with metric scaling, SAM2 [39] for hand-object segmentation, HaMeR [40] for hand pose estimation, and ICP-based tracking to recover object trajectories $\{\hat{x}_t\}_{t=0}^T$. A fused static background is additionally constructed for distance-field collision checking. Figure 7 illustrates an example trajectory from Task 2, where the reconstructed 3D motion of the cup can be visually traced. The execution of the tasks is illustrated in Figure 6. The set of evaluated tasks span a diverse set of motions, including tool use, pouring, cutting, and obstacle-aware transport.

A. Evaluation Metric

The JFTO framework produces a sequence of joint states, which we map through forward kinematics to obtain the end-effector trajectory. Using the first end-effector pose together with the initial object pose, we compute the rigid transform between the gripper and the object. This allows us to derive the executed object trajectory induced by the rollout.

To measure how closely the executed trajectory follows the human demonstration, we report both translational and rotational errors averaged across timesteps:

$$\Delta dist_{avg} = \frac{1}{T} \sum_{t=0}^T \|\mathbf{p}_t^{demo} - \mathbf{p}_t^{exec}\|_2, \quad (23)$$

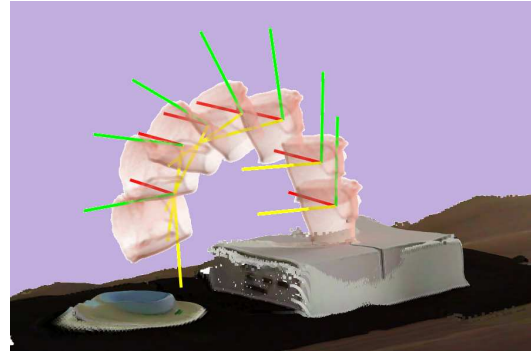


Fig. 7: We applied the upstream processing method over each key frames in the video to get a trajectory of object poses. Here, we observe the object trajectory of Task 2.

$$\Delta rot_{avg} = \frac{1}{T} \sum_{t=0}^T R(\mathbf{r}_t^{demo}, \mathbf{r}_t^{exec}), \quad (24)$$

where $\mathbf{p}_t^{demo}, \mathbf{r}_t^{demo}$ are the position and rotation of the pose of the object demonstrated in timestep t , and $\mathbf{p}_t^{exec}, \mathbf{r}_t^{exec}$ are those of the executed trajectory. The function $R(\cdot, \cdot)$ denotes the angular distance between two orientations.

In addition to these geometric errors, we also compute *average log density* of the poses of the executed object under the demonstration-informed density model introduced in Section IV-C. This provides a probabilistic measure of how well the rollout aligns with the distribution of human demonstrations.

B. Empirical Performance and Hyperparameter Choices

Our proposed JFTO framework is relatively insensitive to hyper-parameters, which trade off each of the desired loss terms. Collisions are *strictly* penalized by setting $\gamma = 1$, reflecting their critical impact on execution safety. For the trade-off between the grasp and the trajectory terms, $\alpha = \beta = 0.5$ yields balanced performance but occasionally leads to unstable grasps under some random seeds. Raising the grasp weight to $\beta = 0.8$ consistently improves reliability, securing successful grasps without significantly degrading trajectory quality.

TABLE I: Comparison of trajectory similarity between joint flow trajectory optimization and sequential optimization. Joint optimization generally yields lower rotational error and higher average log density, indicating better alignment with human demonstrations.

Method	Hammer	Cup	Saw	Figure 8	Drill	Solder	Meter Tape
Joint Optimization							
Avg Dist Diff ↓	0.0108	0.0150	0.0137	0.0116	0.0181	0.0169	0.0270
Avg Rot Diff ↓	0.2572	0.3180	0.2160	0.1787	0.4543	0.3647	0.1686
Avg Log Density ↑	7.0973	6.2451	7.2066	7.4351	7.0655	6.0329	6.8647
Sequential Optimization							
Avg Dist Diff ↓	0.0171	0.0110	0.0123	0.0080	0.0087	0.0296	0.0253
Avg Rot Diff ↓	0.5748	0.4075	0.4263	0.2758	0.3968	0.8603	0.1518
Avg Log Density ↑	1.4823	-2.7326	1.6412	6.5238	5.7580	-1.6030	5.7239

C. Joint vs. Sequential Optimization

We compare our joint flow trajectory optimization (JTFO) against a *sequential* baseline, in which the grasp is first optimized to match the demonstration, then held fixed while a second stage optimizes the object trajectory. In other words, the baseline selects a feasible, demonstration-like grasp and subsequently rolls out a trajectory that minimizes deviation from the demos. JTFO successfully executes all seven tasks (qualitative results in Fig. 6); quantitative metrics are reported in Table I. Both methods achieve similarly low *positional* errors, indicating that objects are generally delivered to the correct locations. However, the sequential baseline incurs substantially higher *rotational* errors and markedly lower log densities on tasks such as *Hammer*, *Cup*, *Saw*, and *Solder*. This suggests that, although the baseline can identify a grasp near the demonstrated one, it often fails to maintain the correct object orientation during execution. In contrast, JTFO reasons about execution feasibility *during* grasp selection. Although this can slightly reduce initial grasp similarity, it yields object poses that remain consistent over time. Consequently, JTFO produces higher-likelihood trajectories that more faithfully reproduce the demonstrated motions.

D. Multi-Modality: Why Flow Matching

Many real-world manipulation tasks admit multiple valid strategies for achieving the same goal. For instance, an object may be moved around an obstacle either to the left or to the right. In such cases, the demonstration distribution is inherently *multi-modal*. Traditional distance-based objectives, such as minimizing the Euclidean distance to all demonstrations, cannot capture this structure. They encourage trajectories to converge toward an “average” solution that lies between modes—often producing motions that are physically infeasible or semantically incorrect.

Flow matching addresses this limitation by explicitly modeling the density of the demonstration distribution. Rather than collapsing distinct behaviors into a single mean, the learned velocity field guides optimization toward one of the high-density regions in $SE(3)$ that correspond to observed demonstrations. The mode being selected depends on the initialization of the optimization, but in all cases the resulting trajectory is consistent with at least one valid

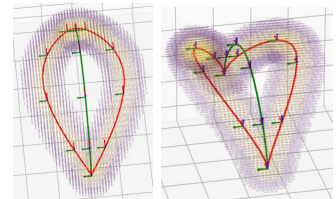


Fig. 8: Visualization of trajectory distributions. Yellow/purple indicates high/low density regions. Flow-based optimization converges to one of the demonstrated red trajectories. Distance-based optimization collapses toward the mean (in green).

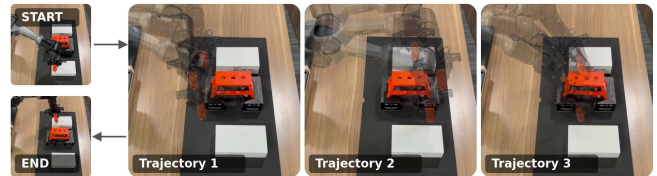


Fig. 9: Given multi-modal trajectory distributions (avoiding obstacles), the flow matching model yields either Trajectory 1 or Trajectory 2, depending on initialization, both of which follow a demonstrated strategy. In contrast, the distance-based model mode-seeks and produces Trajectory 3, cutting through the obstacle in a way not consistent with any human demonstration.

human strategy. This property is illustrated in Fig. 8. The yellow regions represent areas of high demonstration density. Gradient descent on the learned flow drives sampled waypoints into these regions, producing trajectories that align with observed demonstrations (red paths). In contrast, a Euclidean distance objective yields an intermediate solution (green path), corresponding to an unrealistic average trajectory not observed in the data. By preserving multi-modality, flow matching ensures that robots reproduce *plausible human-like strategies*, rather than collapsing behaviors into averages. Here, we extend Task 7 (moving meter tape) by collecting multiple demonstrations around an obstacle: Some routes placed the tape in front, while others carried it behind. Leveraging the flexibility of our JTFO framework with flow matching, the robot reliably reproduced one of these demonstrated strategies—passing in front or behind—depending on initialization (trajectories 1 and 2 in Fig. 9). Both outcomes correspond to faithful imitation. In contrast, applying a naive distance-based metric over the entire set of demonstrations yields an averaged path that attempts to move directly through the obstacle, bending upward to compensate (trajectory 3 in Fig. 9). This trajectory does not resemble any of the demonstrations.

VI. CONCLUSIONS

In this paper, we address generating feasible grasp poses and motion trajectories from video-based LfD, ensuring that robot grasps and object motions remain consistent with human demonstrations despite kinesthetic infeasibility. We presented the Joint Flow Trajectory Optimization (JTFO) framework that integrates grasp selection and trajectory imitation within a single optimization process. By leveraging flow matching as a density estimator, our method naturally

handles multi-modal human demonstrations, avoiding unrealistic averaging effects and guiding the robot toward feasible, human-like behaviors. Experiments in both simulation and on a low-cost 6-DoF robotic arm demonstrated that joint optimization consistently outperforms sequential baselines, achieving lower rotational error and higher trajectory likelihood.

REFERENCES

- [1] S. M. LaValle and J. James J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, 2001.
- [2] W. Zhi, I. Akinola, K. van Wyk, N. Ratliff, and F. Ramos, "Global and reactive motion generation with geometric fabric command sequences," in *IEEE International Conference on Robotics and Automation, ICRA*, 2023.
- [3] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *IEEE International Conference on Robotics and Automation*, 2009.
- [4] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual review of control, robotics, and autonomous systems*, 2020.
- [5] W. Zhi, T. Lai, L. Ott, and F. Ramos, "Diffeomorphic transforms for generalised imitation learning," in *Learning for Dynamics and Control Conference, LADC*, 2022.
- [6] H. Zhou, R. Wang, Y. Tai, Y. Deng, G. Liu, and K. Jia, "You only teach once: Learn one-shot bimanual robotic manipulation from video demonstrations," *arXiv preprint arXiv:2501.14208*, 2025.
- [7] A. Bahety, A. Balaji, B. Abbatematteo, and R. Martín-Martín, "Safemimic: Towards safe and autonomous human-to-robot imitation for mobile manipulation," in *Robotics: Science and Systems (RSS)*, 2025, 2025.
- [8] R. Senanayake, "The role of predictive uncertainty and diversity in embodied ai and robot learning," *arXiv preprint arXiv:2405.03164*, 2024.
- [9] W. Zhi, L. Ott, and F. Ramos, "Probabilistic trajectory prediction with structural constraints," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.
- [10] S. Vasudevan, S. Sagar, and R. Senanayake, "Strategic vantage selection for learning viewpoint-agnostic manipulation policies," *arXiv preprint arXiv:2506.12261*, 2025.
- [11] Y. Lipman, M. S. Albergó, B. Tzen, K. Kelley, N. Jaques, E. Vanden-Eijnden, M. Lezcano, R. T. Q. Chen, S. Fotiadis, and D. Thompson, "Flow matching for generative modeling," *arXiv preprint arXiv:2206.08791*, 2022.
- [12] T. G. W. Lum, O. Y. Lee, C. K. Liu, and J. Bohg, "Crossing the human-robot embodiment gap with sim-to-real rl using one human demonstration," 2025.
- [13] J. Shi, Z. Zhao, T. Wang, I. Pedroza, A. Luo, J. Wang, J. Ma, and D. Jayaraman, "Zeromimic: Distilling robotic manipulation skills from web videos," in *International Conference on Robotics and Automation (ICRA)*, 2025.
- [14] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu, "Egomimic: Scaling imitation learning via egocentric video," in *CoRL Workshop on Learning Robot Fine and Dexterous Manipulation: Perception and Control*, 2024.
- [15] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," *arXiv preprint arXiv:2409.01652*, 2024.
- [16] H. Chen, C. Zhu, Y. Li, and K. Driggs-Campbell, "Tool-as-interface: Learning robot policies from human tool usage through imitation learning," *arXiv preprint arXiv:2504.04612*, 2025.
- [17] S. Bahl, A. Gupta, and D. Pathak, "Human-to-robot imitation in the wild," in *Proceedings of Robotics: Science and Systems (RSS)*, 2022.
- [18] X. Liu, D. Pathak, and K. M. Kitani, "HERD: Continuous Human-to-Robot Evolution for Learning from Human Demonstration," in *The Conference on Robot Learning (CoRL)*, 2022.
- [19] K. Shaw, A. Agarwal, S. Bahl, M. K. Srirama, A. Kirchmeyer, A. Kannan, A. Sivakumar, and D. Pathak, "Demonstrating learning from humans on open-source dexterous robot hands," in *Robotics: Science and Systems (RSS), Demo Session*, 2024. Demo presentation at RSS 2024, Delft, Netherlands.
- [20] W. Zhi, T. Zhang, and M. Johnson-Roberson, "Learning from demonstration via probabilistic diagrammatic teaching," in *IEEE International Conference on Robotics and Automation*, 2024.
- [21] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *arXiv preprint arXiv:1706.09911*, 2017.
- [22] M. Gualtieri, A. T. Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 598–605, 2016.
- [23] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "Pointnetgpd: Detecting grasp configurations from point sets," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.
- [24] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3133–3139, 2021.
- [25] T. Zhang, W. Zhi, B. Meyers, N. Durrant, K. Huang, J. Mangelson, C. Barbalata, and M. Johnson-Roberson, "Recgs: Removing water caustic with recurrent gaussian splatting," *IEEE Robotics and Automation Letters*, 2025.
- [26] J. Ichnowski, Y. Avigal, J. Kerr, and K. Goldberg, "Dex-NeRF: Using a neural radiance field to grasp transparent objects," in *Conference on Robot Learning (CoRL)*, 2020.
- [27] H.-S. Fang, C. Wang, M. Gou, and C. Lu, "Graspnet-1billion: A large-scale benchmark for general object grasping," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [28] C. Wang, H.-S. Fang, M. Gou, H. Fang, J. Gao, and C. Lu, "Graspness discovery in clutters for fast and accurate grasp detection," *arXiv preprint arXiv:2406.11142*, 2024.
- [29] A. R. Moghadam, S. Rastegari, M. T. Masouleh, and A. Kalhor, "Grasp the graph (gtg) 2.0: Ensemble of gnns for high-precision grasp pose detection in clutter," *arXiv preprint arXiv:2505.02664*, 2025.
- [30] A. R. Chen, B. Tzen, Y. Lipman, M. S. Albergó, M. Lezcano, K. Kelley, and E. Vanden-Eijnden, "Optimal transport with flow matching," *arXiv preprint arXiv:2305.08891*, 2023.
- [31] M. S. Albergó, K. Kelley, G. Kanwar, P. E. Shanahan, and E. Vanden-Eijnden, "Stochastic interpolants: A unifying framework for flows and diffusions," *arXiv preprint arXiv:2304.03271*, 2023.
- [32] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *J. Mach. Learn. Res.*, 2021.
- [33] W. Zhi, H. Tang, T. Zhang, and M. Johnson-Roberson, "Teaching periodic stable robot motion generation via sketch," *IEEE Robotics and Automation Letters*, 2025.
- [34] W. Zhi, T. Lai, L. Ott, E. V. Bonilla, and F. Ramos, "Learning efficient and robust ordinary differential equations via invertible neural networks," in *International Conference on Machine Learning, ICML*, 2022.
- [35] S. Wang, V. Leroy, Y. Cabon, B. Chidlovskii, and J. Revaud, "Dust3r: Geometric 3d vision made easy," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709, June 2024.
- [36] J. Zhang, C. Herrmann, J. Hur, V. Jampani, T. Darrell, F. Cole, D. Sun, and M.-H. Yang, "Monst3r: A simple approach for estimating geometry in the presence of motion," *arXiv preprint arxiv:2410.03825*, 2024.
- [37] W. Zhi, H. Tang, T. Zhang, and M. Johnson-Roberson, "Unifying representation and calibration with 3d foundation models," *IEEE Robotics and Automation Letters*, 2024.
- [38] W. Zhi, H. Tang, T. Zhang, and M. Johnson-Roberson, "Simultaneous geometry and pose estimation of held objects via 3d foundation models," *IEEE Robotics and Automation Letters*, 2024.
- [39] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryalí, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.
- [40] G. Pavlakos, D. Shan, I. Radosavovic, A. Kanazawa, D. Fouhey, and J. Malik, "Reconstructing hands in 3D with transformers," in *CVPR*, 2024.
- [41] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1987.