2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

GIFT: Geometry-Induced Functional Transfer for Category-level Object Manipulation

Cristiana de Farias[1*] Luis Figueredo[2,3] Riddhiman Laha[2] Maxime Adjigble[1]
Brahim Tamadazte[4] Rustam Stolkin[1] Sami Haddadin[2] Naresh Marturi[1]

***Abstract*— Robotic manipulation of unfamiliar objects in new environments is challenging due to limited generalisation capabilities. We propose a new skill transfer framework, GIFT (Geometry-Induced Functional Transfer), which enables a robot to transfer complex object manipulation skills and constraints from a single human demonstration. Our approach addresses the challenge of skill acquisition and task execution by deriving geometric representations from demonstrations focusing on object-centric interactions. By leveraging the Functional Maps (FMC) framework, we efficiently map interaction functions between objects and their environments, allowing the robot to replicate task operations across objects of similar topologies or categories, even when they have significantly different shapes. Additionally, our method incorporates screw interpolation (ScLERP) for generating smooth, geometrically-aware robot paths to ensure the transferred skills adhere to the demonstrated task constraints. We validate the effectiveness and adaptability of our approach through extensive experiments, demonstrating successful skill transfer and task execution in diverse real-world environments without requiring additional training.**

I. Introduction

Robotic autonomy in human-centred settings rests on the robot's capability to perform complex tasks in diverse, unknown, or partially known environments. Success often hinges on preserving task-defining constraints, contact relationships, relative poses, and motion primitives, rather than merely reproducing motions or joint trajectories. This topic has recently gained traction thanks to advances in vision-language-action and generative policies that enable zero-/few-shot capabilities by scaling pre-trained data and model size to open-world applications. Well known examples include RT-2 and RT-X [1], [2], OpenVLA [3], and Octo [4], which co-train on large, heterogeneous robot datasets to achieve instruction-conditioned manipulation across embodiments, a practice also explored in some model-based methods. Yet these models typically lack explicit physical and geometric priors and exhibit brittle behaviour when tasks require tight constraint satisfaction or when scenes depart from the training distribution, even as they continue to improve with scale. In parallel, generative visuomotor policies based on diffusion and, more recently, flow matching have achieved strong data efficiency and multimodal action generation from less data-intensive demonstrations, but they also tend to degrade under distribution shift and provide no guarantees that contact, pose, or kinematic constraints will be upheld at execution time [5], [6]. This gap becomes critical even in routine scenarios. For instance, a robot trained to manipulate a bottle by picking it up, stirring its contents, and placing it on a table may fail if the bottle is replaced with one of a significantly different shape. Even if manually guided to a new pose, trajectory transfer can still fail (without larger training data) because of changes in the scene and reference frame. In other words, performance often collapses when the bottle's shape or topology changes unless task-consistent interactions and their invariants are represented explicitly and transferred reliably across instances. Addressing this limitation requires new ways of encoding tasks and actions that generalise across objects and scenes while respecting the inherent constraints that define successful performance. To this end, this paper explores geometry-aware approaches that facilitate the transfer of task execution capabilities, enabling robots to adapt effectively to new scenarios, objects, and conditions even under limited data.

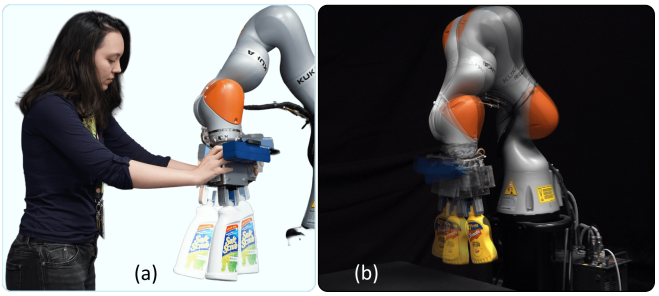

Fig. 1. (a) User demonstrates the bottle shaking operation, and (b) robot imitates it on a different bottle using the proposed skill transfer framework.

We introduce a new paradigm to address this problem, casting geometric representation not as an auxiliary cue but as the backbone of skill representation and transfer. From a single kinesthetic demonstration, we derive object-centric interaction functions that encode both task-specific grasp states and the relative constraints and object-robot and object-environment interactions. These functions are defined on the object's topology and are transported to novel, category-level instances via functional correspondences, enabling transfer to shapes with the same topology even when their geometry differs significantly. The transported interaction is then executed through a geometry-consistent path generator based on constant-screw interpolation in task space, which preserves the demonstrated relative transformations along the trajectory

This work was supported by the UK National Centre for Nuclear Robotics, and by Horizon Europe project REBELION grant 101104241.
[1]Extreme Robotics Laboratory, School of Metallurgy and Materials, University of Birmingham, Birmingham, United Kingdom.[2]Munich Institute of Robotics & Machine Intelligence, Technische Universität München (TUM), Germany. [3]School of Computer Science, University of Nottingham, UK. [4]Sorbonne Université, ISIR, Paris, France. †Corresponding Author: CXM1029@alumni.bham.ac.uk.

and blends reactively in real time. Unlike purely end-to-end pipelines, our proposed method, GIFT, Geometry-Induced Functional Transfer, is designed from the ground up around geometric structure to promote interpretability, strict constraint fidelity, and data efficiency. The skill can be transferred to many new instances without additional training; the constraint set is preserved by construction rather than inferred; and a single demonstration seeds an expandable family of actionable experiences via correspondence-driven projection to new shapes [7].

Our formulation builds on two complementary, geometry-first insights that have remained largely disconnected in prior robotics literature. First, object geometry can drive category-level transfer of grasps and task affordances when robust correspondences are available. For instance, methods such as [7]–[10] have shown that semantically anchored geometric representations enable transferring action goals across instance motivating interaction functions defined on surfaces rather than on brittle instance templates. We extend this line by explicitly encoding robot–object and object–environment relations from demonstrations and transporting them through functional correspondences, which we instantiate via functional-map-based operators [11] to handle significant shape changes while respecting topology. Second, constant-screw interpolation (ScLERP) provides a coordinate-invariant mechanism to reconstruct trajectories that conserve the implicit geometric constraints embedded in the demonstration and within the object (regardless reference-frames) [12], while also supporting reactive online blending and smooth tracking in novel scenes. The integration is smooth and ensures the geometric information transferred from within the category of objects, that is, to new unseen objects which might have different reference frames is respected. This is due to the known properties of screw representation, which is ad-invariant [12]–[14], that is, it respects changes in the reference frames in the observed objects, in contrast to decoupled methods, ubiquitously used in the literature.

Together these ideas yield GIFT, a coherent, geometry-embedded system in which (i) interaction functions formalise what must be preserved across object instances; (ii) functional correspondences instantiate where on a new shape those interactions live; and (iii) constant-screw interpolation dictates how motions evolve so that relative constraints remain invariant under changing reference frames. This design yields **explainability** (every step is anchored to surface functions and screw geometry), **predictability** (constraint satisfaction follows from representation and interpolation choices), **reactivity** (online blending under disturbances while staying on the constraint manifold), and data efficiency (one-shot transfer without policy retraining). We validate these claims in cluttered, real-world scenes using a 7-DoF platform with wrist RGB-D sensing, demonstrating consistent transfer to previously unseen instances while preserving the original geometric constraints throughout execution.

II. Related Work

Among the many challenges related to skill transfer and learning, this work mainly focuses in two areas which are often studied separately, (i) few-shot learning of complex of complex manipulation skills; and (ii) semantic or geometric correspondence for category-level transfer. In an end-to-end approach, this problem has also been recently addressed via recent advances in (iii) generative visuomotor policies for zero-/few-shot control.

Executing tasks in new environments is often approached through imitation learning methods such as behaviour cloning or video demonstration [15]–[18]. While effective in constrained settings, these approaches generally require extensive retraining and large datasets. Dynamic Movement Primitives (DMPs) [19]–[21] provided an alternative representation for encoding skills, but they are difficult to adapt to waypoint-based, object-centric interactions [22]. Probabilistic extensions such as ProMPs improved generalisation, yet typically converge only within the local region of the demonstration, limiting applicability to unseen scenarios [13], [23]. More recently, relative-constraint approaches [24] have shown that single-shot demonstrations can enable efficient, real-time deployment with object-centric constraints; however, their scope often remains tied to the specific object instance and manually defined reference frames used during demonstration, restricting transferability.

To further advance robot generalisation, recent works have relied on semantic or geometric feature correspondences that align object instances so that task goals can be transported across shapes and objects. For example, Wen et al. [25] employed dense correspondence for bolt repositioning and insertion in cluttered scenes, while Tekden et al. [26] transferred grasps across object categories by leveraging shared semantic parts such as lids or handles. For full skill transfer of category-level objects, tasks have been described using semantic keypoints extracted from object point clouds [8], [9]. Neural Descriptor Fields (NDFs) extend this paradigm by learning implicit object models that capture task-informed relationships between descriptors, enabling category-level generalisation for manipulation [27], [28]. More recently, transformer-based frameworks such as DiNoBot [29] have been proposed for imitation learning. Despite these advances, such methods often rely on resource-intensive (re)training, extensive manual labelling, or large black-box pre-trained models. Furthermore, planning task-oriented grasps and object-relative trajectories in unseen scenarios—where geometry and reference frames change—is particularly fragile when trajectory generation is decoupled from the semantic and geometric assumptions used for grasp or object information transfer. This is precisely the setting in which GIFT excels. It integrates correspondences from keypoints or descriptor fields and immediately enacts them via geometry-consistent screw interpolation with explicit constraint preservation, yielding closed-loop behaviour that respects object-relative invariants rather than isolated waypoints.

In contrast, a third strand scales foundation models and

generative visuomotor policies to achieve broad zero-/few-shot coverage across robots and tasks. RT-2 showed that co-training VLA models on web knowledge and robot trajectories unlocks semantic manipulation from language instructions [1], Open-X Embodiment and RT-X established large, standardised multi-robot datasets and policies for cross-embodiment transfer [2]. OpenVLA provided an open VLA baseline amenable to efficient fine-tuning [3] and Octo delivered an open generalist policy trained on hundreds of thousands of trajectories [4]. In parallel, diffusion policy and newer flow-matching formulations substantially improve data efficiency and multimodal action generation from limited demonstrations [5], [6], [30]. Nevertheless, these systems typically lack explicit geometric or physical priors and provide no guarantees that task-critical constraints will be upheld during execution, often degrading under shape/topology shift or camera/pose drift. Recent efforts such as ReKep, [31] bring constraints closer to the closed-loop by continuously prompting large VLMs for keypoints and code-level relations, but they still rely on repeated perception–language inference and strong open-world semantics, which increases computational load and often leads to degradation during long-horizon control. Even widely successful platforms such as ALOHA illustrate the sensitivity of zero-shot policies to viewpoint and scene distribution, with generalisation often confined to narrow camera frustums and object placements [32]. These models are improving rapidly and will continue to broaden coverage. Our aim is not to compare GIFT directly to such methods, but rather to offer a fully orthogonal and complementary approach GIFT targets guaranteed, geometry-aware execution and one-shot transfer. In future work, this geometric grounding can be layered beneath VLAs or diffusion/flow policies to harden real-world behaviour by enforcing topology-anchored functional interactions and screw-consistent trajectories. We therefore do not report head-to-head scoreboards against evolving, data-hungry generalist baselines whose training assumptions and objectives differ markedly from ours, instead, we target the regime where explicit constraint fidelity, explainability, and predictable generalisation across substantial shape variation are paramount.

III. Problem Description and Preliminaries

A. Problem definition

This work focuses on the challenge of generalising task execution to new category-level objects based on a single-user demonstration. To address this, we define a skill as:

Definition (Skill)**.** A skill is defined as the set of task operations $\mathcal{T}_i$ in which every operation is comprised of trajectories and functions defined over an object's surface that encode the geometric characteristics and task-specific requirements for performing a desired manipulation task. It can be expressed as a tuple

$$\mathcal{T} = \{\underline{\boldsymbol{\mathcal{X}}}, \mathbf{f}_{(\text{task})}\}, \tag{1}$$

where, $\underline{\boldsymbol{\mathcal{X}}} = \{\underline{\boldsymbol{x}}_0, \underline{\boldsymbol{x}}_1, \underline{\boldsymbol{x}}_2 \dots \underline{\boldsymbol{x}}_n\}$ represents the robot's end-effector trajectory from the initial pose $\underline{\boldsymbol{x}}_0$ to the final pose $\underline{\boldsymbol{x}}_n$, and $\mathbf{f}_{(\text{task})} \in \mathcal{S}$ is an interaction function defining the task-specific relationships between the robot, the environment, and other objects in the scene, with $\mathcal{S}$ being the Riemannian manifold representing an object's shape.

During deployment in new scenes, it is assumed that the initial and final poses of relevant objects adhere to implicit geometric-aware task-relevant constraints obtained during a successful demonstration. These object-centric constraints, defined by a sequence of constant screw-connected segments, are captured by robot-object and object-environment interaction functions, forming a task operation tuple as in (1). These constraints are expressed as a surface function that, when coupled with FMC matching, can be transferred to new objects of the same category. Screw transformations are applied to the new key poses and segments, with screw interpolation ensuring boundary constraints between the interpolated poses.

Problem Statement. Let a demonstration scene $\boldsymbol{\Theta}^{\text{dem}}$ composed of N objects be represented as $\boldsymbol{\Theta}^{\text{dem}} = \{\mathcal{S}_1^{\text{dem}}, \mathcal{S}_2^{\text{dem}}, \dots, \mathcal{S}_N^{\text{dem}}\}$ with a sequence of k operations kinesthetically demonstrated by the user and described by the tuple (1), i.e.,

$$\mathcal{T}_i^{\text{dem}} = e\{\underline{\boldsymbol{\mathcal{X}}}_i^{\text{dem}}, \mathbf{f}_{(\text{task},i)}\}, \quad i = 1, \dots, k. \tag{2}$$

Compute the corresponding robot operations $\mathcal{T}_i'$ that enable the robot to replicate the demonstrated skill with new objects and within the new environment, for instance, a new scene $\boldsymbol{\Theta}'$, containing M objects.

The aforementioned problem is outlined in two key steps:

(✓) **Transferability of interaction functions:** Interaction functions describing spatial and task-specific constrained relationships from the demonstrated scene $\boldsymbol{\Theta}^{\text{dem}}$ must be adaptable to new objects within the same category. This adaptation should account for differences in shape and potential deformations of the new objects.

(✓) **Maintenance of geometric constraints:** All implicit geometric constraints observed during the demonstration must be consistently maintained during task execution in new environments, regardless of changes in the object's position or the reference frame.

B. Functional map correspondence

The FMC correspondence pipeline facilitates the transfer of skill functions between objects by mapping vertices from one shape to another. For two object shapes $\mathcal{S}_1$ and $\mathcal{S}_2$, the FMC pipeline computes the map $M : \mathcal{S}_2 \to \mathcal{S}_1$. The process consists of the following steps:

i) Compute a set of orthonormal bases for each shape, storing the coefficients as columns $\boldsymbol{\Phi}_{\mathcal{S}_1}$ and $\boldsymbol{\Phi}_{\mathcal{S}_2}$. Use the first n eigenfunctions of the Laplace-Beltrami (LB) operator [11]. The LB basis decomposes the shape into harmonic elements, invariant to isometries and rigid motions, and is computed on 3D meshes [33].

ii) Calculate descriptor functions for each shape, which are expected to be approximately invariant across isometric

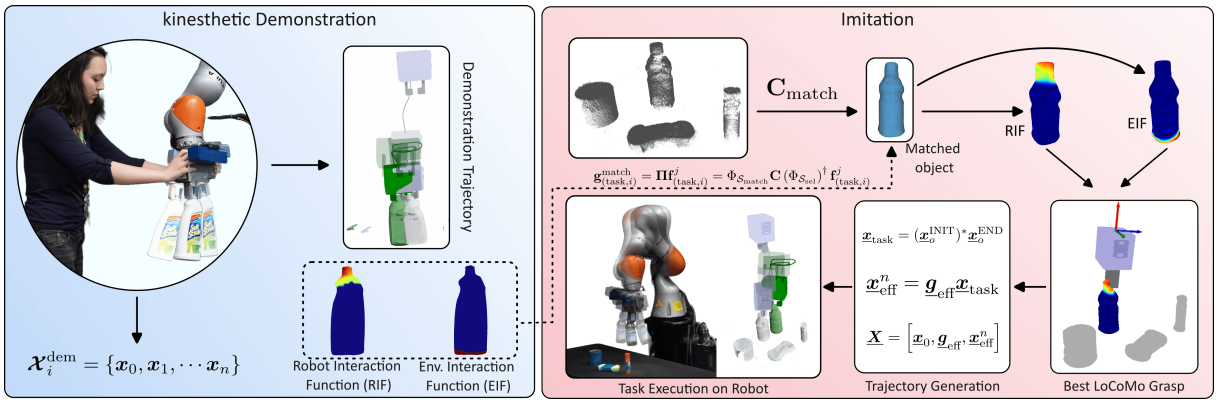

Fig. 2. Pipeline of the proposed method (GIFT), comprising the demonstration and imitation stages. The bottle stirring skill is considered here. In the demonstration stage, kinesthetic demonstrations are performed to capture the gripper's placement and object manipulation motions. GIFT then generates functions based on these demonstrations. When the robot encounters a new scene, these functions are transferred to similar objects, allowing the robot to determine corresponding gripper poses. With a new set of waypoints and the demonstrated trajectory, the tasks are generalised to novel scenes.

shapes. Represent these functions as $\mathbf{f}_i \in \mathbf{\Phi}_{\mathcal{S}_1}$ for shape $\mathcal{S}_1$ and $\mathbf{h}_i \in \mathbf{\Phi}_{\mathcal{S}_2}$ for $\mathcal{S}_2$. Store the coefficients in matrices $\mathbf{F}$ and $\mathbf{H}$, where each column corresponds to a descriptor function. The Wave Kernel Signature [34], which captures intrinsic geometric properties across multiple scales, is commonly used for this purpose.

iii) Solve the optimisation problem to obtain the optimal functional map $\mathbf{C}$ in the LB basis

$$\mathbf{C} = \arg\min_{\mathbf{C}} \left(\alpha_1 E_{\text{DP}}(\mathbf{C}) + \alpha_2 E_{\text{REG}}(\mathbf{C})\right). \quad (3)$$

where α_1 and α_2 are scalar gains. The term $E_{\text{DP}}(\mathbf{C}) = \|\mathbf{CF} - \mathbf{H}\|^2$ ensures the preservation of significant shape features, while $E_{\text{REG}}(\mathbf{C})$ adds regularisation constraints for robustness [35].

iv) Refine and convert $\mathbf{C}$ into a point-to-point correspondence vector $\mathbf{T}_p$ using the *ZoomOut* iterative upsampling method [36].

C. User-guided trajectory generation

In this work, we generates trajectories through constant-screw interpolation (ScLERP) that adhere to the geometric constraints of the demonstrated path, $\underline{\boldsymbol{\mathcal{X}}}^{\text{DEM}}$. Here, by using a screw representation, we also guarantee the ad-invariance property is transmitted. That is, even when transferring to a new object which might have a different reference frame representation, the relationship between all points in the body are maintained therefore the trajectory is consistent. In other words, it avoids the issue with decoupled methods where depending on the reference frame (observer view) the interpolated path might be different [24]. This path is represented as a sequence of unit dual quaternion poses, given as

$$\underline{\boldsymbol{\mathcal{X}}}^{\text{dem}} \triangleq \left\{ \begin{array}{cccc} \underline{\boldsymbol{x}}_0 & \underline{\boldsymbol{x}}_1 & \cdots & \underline{\boldsymbol{x}}_n \end{array} \right\}, \quad (4)$$

with $\underline{\boldsymbol{x}}_1, \underline{\boldsymbol{x}}_2, \ldots, \underline{\boldsymbol{x}}_n \in \mathbb{H} \otimes \mathbb{D}$. The relative path concerning the final end-effector pose $\underline{\boldsymbol{x}}_n$ is then computed as

$$\underline{\boldsymbol{\delta}}_i = \underline{\boldsymbol{x}}^{*}_{i-1} \underline{\boldsymbol{x}}_i, \quad i = 2, \ldots, n. \quad (5)$$

Given the desired end-effector goal pose $\underline{\boldsymbol{x}}'_n$ in the new scene $\mathbf{\Theta}'$, the imitated path $\underline{\boldsymbol{\mathcal{X}}}' \triangleq \left\{ \begin{array}{cccc} \underline{\boldsymbol{x}}'_0 & \underline{\boldsymbol{x}}'_1 & \cdots & \underline{\boldsymbol{x}}'_n \end{array} \right\}$ is obtained from (5), as

$$\underline{\boldsymbol{x}}'_{i-1} = \underline{\boldsymbol{x}}'_n \underline{\boldsymbol{\delta}}^{*}_i, \quad i = 2, \ldots, n. \quad (6)$$

This representation ensures that $\underline{\boldsymbol{\mathcal{X}}}'$ preserves the relative transformations of the demonstrated path throughout the imitated path, from the initial pose $\underline{\boldsymbol{x}}'_0$ to the new goal $\underline{\boldsymbol{x}}'_n$.

The objective is to construct a new path starting from a different initial configuration $\underline{\boldsymbol{x}}''_0$ in the task space, which does not necessarily align with the original pose $\underline{\boldsymbol{x}}'_0$. For this, intermediate points are generated in the task space such that the new path $\underline{\boldsymbol{\mathcal{X}}}''$ blends into demonstrated path $\underline{\boldsymbol{\mathcal{X}}}'$, using the ScLERP method [37]. Given the current end-effector pose $\underline{\boldsymbol{x}}_{\text{eff}}$, at any time step, and a guiding pose $\underline{\boldsymbol{x}}'_i \in \underline{\boldsymbol{\mathcal{X}}}'$ on the imitated path, the reference end-effector pose is

$$\underline{\boldsymbol{x}}_r(\tau) = \underline{\boldsymbol{x}}_{\text{eff}} \left(\underline{\boldsymbol{x}}^{*}_{\text{eff}} \underline{\boldsymbol{x}}'_i\right)^{\tau}, \quad (7)$$

with $\tau \in [0, 1]$ being the interpolation sampling time that makes a discrete path linearly scaled along the geodesic between any current end-effector pose and the guiding pose.

IV. Methodology

The proposed skill transfer methodology is presented below. As shown in Fig. 2, GIFT consists of two stages:

(i) **Demonstration**, where the demonstration scene $\mathbf{\Theta}^{\text{dem}}$ and $\mathcal{T}_i^{\text{dem}}$ are acquired; and

(ii) **Imitation**, where the robot performs the demonstrated skill in a new scene with different objects.

The demonstration stage begins with the robot acquiring a complete scene $\mathbf{\Theta}^{\text{dem}}$, by stitching point clouds from multiple scans as described in [38]. Individual scene object clusters are segmented using the DBSCAN algorithm [39] and then converted to meshes using screened-Poisson reconstruction with Dirichilet boundary constraints [40]. These are stored in the database. Next, a user kinesthetically demonstrates the skill by hand-guiding the robot to perform the desired motions. This generates a set of paths, $\{\underline{\boldsymbol{\mathcal{X}}}_{i|i=1,2\ldots}\}$, where each path contains poses, $\underline{\boldsymbol{\mathcal{X}}}_i^{\text{dem}} = \{\underline{\boldsymbol{x}}_0, \underline{\boldsymbol{x}}_1, \ldots, \underline{\boldsymbol{x}}_n\}$ computed using the robot's forward kinematic model. Interaction functions (see Sec. IV-A), $\mathbf{f}^{j}_{(\text{task,i})} \in \mathcal{S}_j^{\text{dem}}$, are then computed, defining skill-specific relationships between the robot, the

objects and the environment, which are crucial for generalising the skill to new scenes. Finally, $\mathcal{T}_i^{\text{dem}}$ can be obtained from each element in $\underline{\boldsymbol{\mathcal{X}}}_i^{\text{dem}}$ and $\mathbf{f}_{(\text{task,i})}^j$, as in (2).

In the imitation stage, the goal is to obtain an equivalent $\mathcal{T}_i'$ with new functions and trajectories for a novel scene, $\boldsymbol{\Theta}'$. After extracting shapes from the new scene (similarly to the demonstration stage), the FMC framework is used to transfer functions $\mathbf{f}_{(\text{task},i)}^j$, resulting in

$$\mathbf{g}_{(\text{task},i)}^k = \boldsymbol{\Pi}\mathbf{f}_{(\text{task},i)}^j, \tag{8}$$

where $\mathbf{g}_{(\text{task},i)}^k$ represents the transferred function applied to the new object $\mathcal{S}_k'$, which best matches the demonstrated object $\mathcal{S}_j^{\text{dem}}$. $\boldsymbol{\Pi}$ represents the mapping between these two shapes and their corresponding manifolds. In scenes with multiple candidate objects, a matching score (see Sec. IV-B) based on the functional map $\mathbf{C}_j$ is used to select the best match. After selecting the matching object and transferring functions, new end-effector poses are calculated, and the path is determined using ScLERP, which maintains the inherent screw-transformation-based constraints from the original demonstration.

A. Interaction functions

Given a scene $\boldsymbol{\Theta}^{\text{dem}}$ and the demonstrated trajectory $\underline{\boldsymbol{\mathcal{X}}}_i^{\text{dem}}$, we define interaction functions, $\mathbf{f}_{(\text{task},i)}^j$, over the object's surface to encode the interactions between the object, robot and the environment. Specifically, our functions include:

(i) Robot Interaction Functions (RIF), which define the contact regions between the robot's end-effector and the object; and
(ii) Environment Interaction Functions (EIF), which define the interactions between the object and a known part of the environment.

For demonstration scenes with multiple objects, the object of interest, $\mathcal{S}_{\text{sel}}$, is selected as the one nearest to the robot's end-effector during the last step of $\underline{\boldsymbol{\mathcal{X}}}_i^{\text{dem}}$.

1) Robot interaction function (RIF): This specifies areas on the object's surface where the robot can interact. Considering $\mathcal{S}_{\text{sel}}$ and the gripper's kinematics, we identify N_f contact points, $\mathbf{Q}^{(\text{contacts},f)} \in (\mathcal{S}_{\text{sel}})^{N_f}$, between robot's fingers and the object. The RIF is calculated as

$$\mathbf{f}_{(\text{task},i)}^{\text{RIF}} = \max\left(1 - \frac{\left\|\bar{\mathbf{Q}}^f - \mathbf{p}\right\|}{\lambda_D}, 0\right) \quad \forall \mathbf{p} \in \mathcal{S}_{\text{sel}} \tag{9}$$

where $\bar{\mathbf{Q}}^f$ is the average of $\mathbf{Q}^{(\text{contacts},f)}$ for each finger, and λ_D is a distance threshold to increase the contact region.

2) Environment interaction function (EIF): This specifies the areas of an object that interact with known elements of the environment. These elements are simplified into a set of N_p planes $\mathbf{P} = \{\mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_{N_p}\}$, where each plane $\mathcal{P}_k$ is defined by a point $\mathbf{o}_k \in \mathbb{R}^3$ and a normal vector $\mathbf{n}_k \in \mathbb{R}^3$.

Given a selected object $\mathcal{S}_{\text{sel}}$ and the relative transformation from the demonstration, $\underline{\boldsymbol{x}}_{\text{rel}} = \underline{\boldsymbol{x}}_0^* \underline{\boldsymbol{x}}_n$, with $\underline{\boldsymbol{x}}_0, \underline{\boldsymbol{x}}_n \in \underline{\boldsymbol{\mathcal{X}}}_i^{\text{dem}}$ being the initial and final trajectory points, $\mathcal{S}_{\text{sel}}$ is transformed to a new configuration, updating the scene $\boldsymbol{\Theta}^{\text{dem}}$ accordingly. Once the object's pose is updated, the EIF is defined as

$$\mathbf{f}_{(\text{task},i)}^{\text{EIF}} = \begin{cases} 1, & \text{if} \quad \|(\mathbf{o}_k + \lambda_p \mathbf{n}_k) - \mathbf{p} \cdot \mathbf{n}_k\| \leq 0 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

$\forall \mathbf{p} \in \mathcal{S}_{\text{sel}}$. Here, λ_p is a distance threshold. A separate function is defined for each plane the object contacts. Typically, three planes are sufficient to determine an object's full pose.

B. Object selection

To calculate the mapping $\mathbf{C}$, an energy function is minimised to represent features in the LB bases $\boldsymbol{\Phi}_{\mathcal{S}_1}$ and $\boldsymbol{\Phi}_{\mathcal{S}_2}$. For perfectly isometric shapes with aligned bases, $\mathbf{C}$ is a diagonal matrix. However, with relaxed isometry or misalignments, $\mathbf{C}$ becomes diagonally dominant rather than strictly diagonal. It further degrades as the shape deviates from the reference shape. Using this property, we developed a criteria for selecting and transferring functions to the correct objects in multi-object scenes.

Given an object $\mathcal{S}_{\text{sel}}$, we find matches with all objects in $\boldsymbol{\Theta}'$, resulting in a set of maps $\mathcal{C} = \{\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^j, \ldots, \mathbf{C}^M\}$, where each $\mathbf{C}^j$ maps $\mathcal{S}_{\text{sel}}$ to $\mathcal{S}_j' \in \boldsymbol{\Theta}'$. The most diagonally dominant map, $\mathbf{C}_{\text{match}} \in \mathcal{C}$ is the best match belonging to the same class. The metric $R\left(\mathbf{C}^j\right)$ is defined as

$$R\left(\mathbf{C}^j\right) = \left(\sum_{i,k}(1 - w_{ik})|c_{ik}|\right) - \left(\sum_{i,k} w_{ik}|c_{ik}|\right) \tag{11}$$

where c_{ik} and w_{ik} are the elements of $\mathbf{C}^j$ and the normalised weight matrix $\mathbf{W}$, respectively. Note that $\mathbf{W}$ has higher weights favouring the diagonal. This metric measures diagonal dominance, and the object in the scene with the highest score is selected as the most similar object to the demonstrated object, i.e., $(\mathcal{S}_{\text{match}}, \mathbf{C}_{\text{match}}) = \underset{\mathcal{S}_j' \in \boldsymbol{\Theta}'}{\arg\max}\, R\left(\mathbf{C}^j\right)$.

C. Skill execution in novel scenes

After obtaining the demonstrated operations $\mathcal{T}_i^{\text{dem}}$, the robot is deployed in a new environment with different objects. It starts by acquiring a new scene $\boldsymbol{\Theta}'$, with different and previously unseen objects. Each function in $\mathcal{T}_i^{\text{dem}}$ is then transferred to the corresponding objects, i.e., given the RIF, $\mathbf{f}_{(\text{task},i)} \in \mathcal{S}_{\text{sel}}$, a set of maps $\mathcal{C}$ is computed for every object. After identifying the best matching object $\mathcal{S}_{\text{match}}$, the corresponding $\mathbf{C}_{\text{match}}$ is used to transfer functions to the matched shape. This transfer is computed as

$$\mathbf{g}_{(\text{task},i)}^{\text{match}} = \boldsymbol{\Pi}\mathbf{f}_{(\text{task},i)}^j = \Phi_{\mathcal{S}_{\text{match}}} \mathbf{C} \left(\Phi_{\mathcal{S}_{\text{sel}}}\right)^\dagger \mathbf{f}_{(\text{task},i)}^j. \tag{12}$$

where $\Phi_{\mathcal{S}_{\text{match}}}$ and $\Phi_{\mathcal{S}_{\text{sel}}}$ are the LB bases on the matched and demonstrated objects, respectively and † is the *Moore–Penrose* pseudoinverse. This process is repeated for all functions in the demonstration.

Typically, the defined skills result in two scenarios: (i) one RIF with two trajectories, one for approach and one for post-contact movement; and (ii) two full operations, containing a trajectory with an RIF for grasping and an EIF for skill execution. In the first scenario, a simplified algorithm

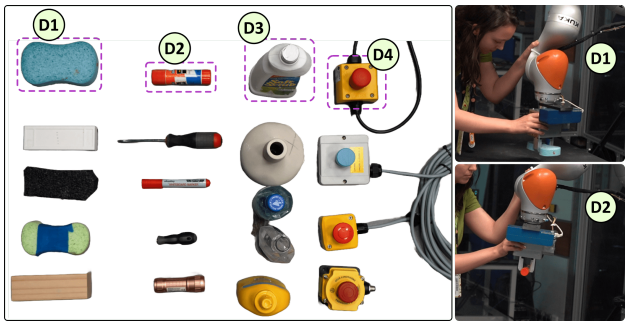

Fig. 3. Dataset of objects used for experiments, with objects marked with numbers are used for demonstration. Two user demonstrations are shown.

is applied without grasping. For skills such as pushing or button-pressing, the robot's gripper remains closed, and a desired gripper pose is set relative to the new object's RIF. This pose serves as the goal for our ScLERP , which then interpolates a new trajectory to execute the skill. For scenario (ii), both the gripper's contact regions (RIF) and the final pose with respect to the environment (EIF) are included. After the transfer, a point cloud is generated from mesh vertices where $\mathbf{g}^{\text{RIF}}_{(\text{task},i)} > \delta$ with δ serving as the cutoff value for the grasping region. This is then used with LoCoMo grasp planner [41] (or alternatively [42]) to generate a ranked set of grasp poses $\mathcal{G}$. LoCoMo was selected for its strong performance in various scenarios [43]. $\mathcal{G}$ is then filtered by removing grasps that either collide with the environment or are kinematically infeasible. Grasps are further refined by retaining only those within a cone of angle θ around the gripper's demonstrated approach vector $\mathbf{a}_o^{\text{dem}}$. The top-ranked grasp $\underline{\boldsymbol{g}}^{\text{top}}$ is then selected for execution.

Once the new task-aware grasp is generated, the next step involves finding the final object and gripper poses. First, we generate point clouds $\mathbf{S}_{\text{sel}} = \{\mathbf{p} \mid \mathbf{f}^{\text{EIF}}_{(\text{task},i)}(\mathbf{p}) = 1\}$ and $\mathbf{S}_{\text{match}} = \{\mathbf{p} \mid \mathbf{g}^{\text{EIF}}_{(\text{task},i)}(\mathbf{p}) = 1\}$, respectively for demonstrated and matched objects. Then by using point-to-point FMC, we compute the optimal rotation quaternion $\mathbf{x}_R$ to align $\mathbf{S}_{\text{match}}$ with $\mathbf{S}_{\text{sel}}$. This alignment is further refined using Iterative closest point (ICP). The $\mathbf{S}_{\text{match}}$ object's transformation from its initial pose, $\underline{\boldsymbol{x}}_o^{\text{INIT}}$ to its final pose $\underline{\boldsymbol{x}}_o^{\text{END}}$ is given by

$$\underline{\boldsymbol{x}}_{\text{task}} = (\underline{\boldsymbol{x}}_o^{\text{INIT}})^* \underline{\boldsymbol{x}}_o^{\text{END}}. \tag{13}$$

Applying this transformation to the gripper's pose in the world frame gives the final end-effector pose

$$\underline{\boldsymbol{x}}^n_{\text{eff}} = \underline{\boldsymbol{g}}_{\text{eff}} \underline{\boldsymbol{x}}_{\text{task}}. \tag{14}$$

With the new poses, ScLERP generates new trajectories based on the demonstrated ones. We create a goals list $\underline{\boldsymbol{X}} = \left[\underline{\boldsymbol{x}}_0, \underline{\boldsymbol{g}}_{\text{eff}}, \underline{\boldsymbol{x}}^n_{\text{eff}}\right]$, where $\underline{\boldsymbol{x}}_0$ represents the robot's new starting pose during imitation. For each consecutive pair in $\underline{\boldsymbol{X}}$, a novel imitated path $\underline{\boldsymbol{\mathcal{X}}}''_i$ is generated. The robot then uses these paths to execute the transferred skills.

V. Experimental Validations

A. Setup description

Our experimental setup comprises a 7-axis KUKA iiwa robot equipped with a Schunk PG 70 gripper and an Ensenso N35 3D camera mounted on the wrist. Fig. 3 illustrates the set of objects used for experiments, where objects marked with D1 – D4 are specifically designated for skill demonstration. The skills associated with these objects are: wiping with D1, drawing a box with D2, bottle stirring with D3, and button pressing with D4. The FMC framework was implemented using the packages from [36], with the number of LB bases set to 85 and upsampled to 200 during refinement. All operations using dual quaternion algebra were performed with the DQrobotics package [44]. All experiments were run on an Ubuntu 20.04 PC with an Intel i7 4-core CPU and 32 GB of RAM. The analysis began by evaluating the GIFT's ability to match shapes in a new scene with multiple objects. Following this, we assessed function transfer, focusing on RIF and EIF. Finally, we evaluated the overall skill transfer framework by examining the robot's ability to transfer and execute complete tasks in the new scene.

TABLE I

Evaluation of function transfer across various skills.

Skill	RIF		EIF		Total		Time
	MAE*	STD*	MAE*	STD*	MAE*	STD*	(s)
D1	0.175	0.200	0.201	0.330	0.188	0.265	4.16
D2	0.159	0.198	0.170	0.215	0.165	0.206	7.89
D3	0.127	0.181	0.061	0.161	0.094	0.171	5.84
D4	0.145	0.193	-	-	0.145	0.193	11.89

* Values range from 0 to 1, with 0 being the best and 1 the worst.

B. Multi-object scene

We first demonstrate GIFT's ability to handle complex scenes with multiple objects. As detailed in Sec. IV-B, we compute maps for all objects in the scene and assign scores to each of them. The one with the highest score is identified as the best match. Fig. 4 illustrates the matching and selection process for two different scenes, where the green mesh (demonstrated object) on the top left side is matched with all objects in the point cloud shown on the bottom left side. The computed maps are shown on the right with the corresponding matching scores overlaid. In both examples, GIFT correctly identified the best-matching object (marked with a red dotted line) that has the most diagonally dominant map. Specifically, in scene (a), we presented the robot with two different bottles in the scene. While both bottle maps were diagonal, the chosen bottle achieved the highest score, i.e., it was more similar in size and shape to the source mesh. This similarity better preserved the isometry constraints, resulting in a more optimal map.

C. Function transfer analysis

To assess our GIFT's effectiveness in transferring the RIF and EIF, we measured the Mean Absolute Error (MAE) and standard deviation (STD) of the function transfer relative to a manually annotated ground truth, as in [10]. Table I presents the average results from four trials for each skill, where both RIF and EIF were transferred from the objects used for demonstration to similar category-level objects (see Fig.3). Fig.5 displays the transferred RIF and EIF for all objects. We observe that the total error remains low for all skills.

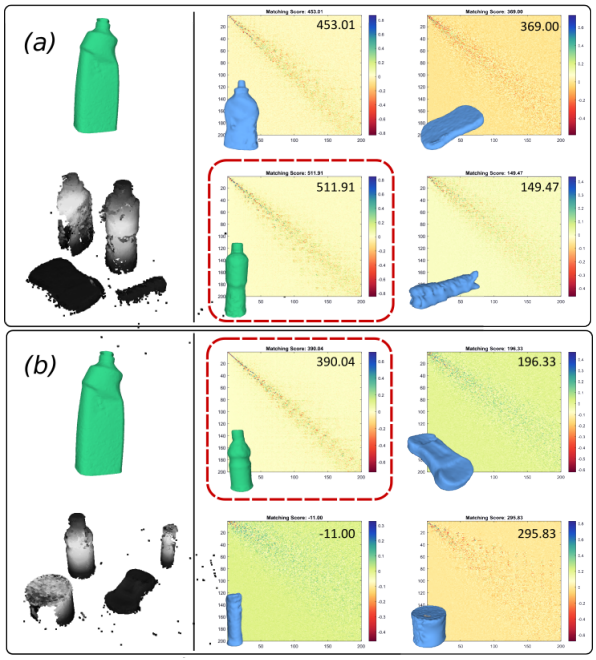

Fig. 4. Illustration of object matching in two multi-object scenes. For each scene object, a map is computed and shown on the right with scores overlaid. The object with the highest matching score is selected.

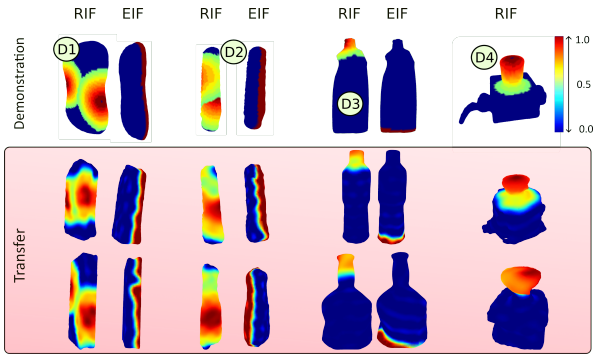

Fig. 5. Illustration of function transfer from the demonstration objects (top row) to other objects within the same category.

However, for D1, the error is slightly higher, likely due to the FMC not accounting for the object's symmetry, occasionally resulting in the rotation of the map around the object's length. Since the RIF is slightly slanted, this rotation contributes to the increased error. For less symmetric object-function pairs, such as D3 and D4, this is reduced. Nevertheless, the error remains small, and the functions are sufficiently similar to enable skill transfer. Furthermore, we analysed the time taken to transfer the functions, as FMC calculation is the most resource-intensive step in our method. Although processing time can vary based on the number of vertices and the object's geometry, the average time across all object categories is 7.4s.

D. Skill transfer analysis

This section evaluates the full skill transfer performance of our approach. Fig. 6 shows results for three skills: wiping a surface (D1), drawing a box (D2), and button press (D4). Both demonstrated and imitated trajectories are displayed,

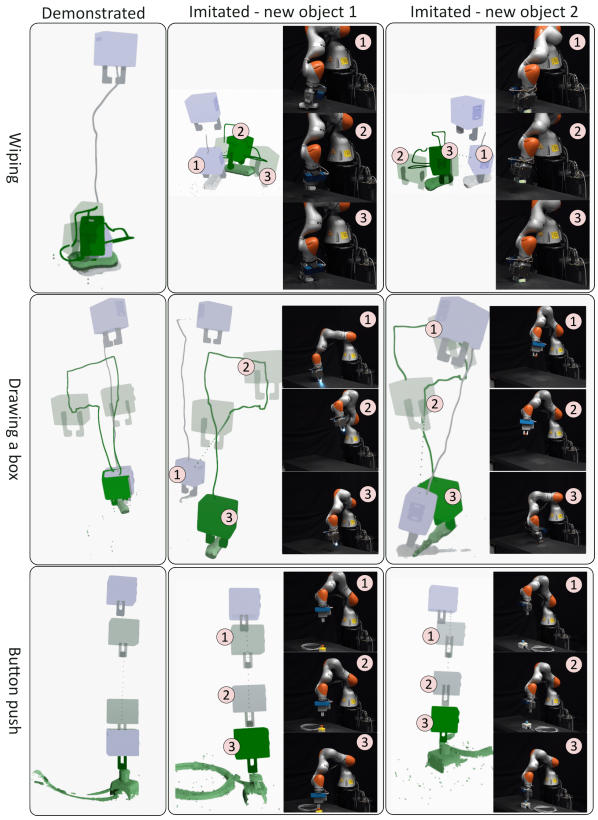

Fig. 6. Illustration of three full skill transfers: wiping, drawing a box, and pressing a button, using two test objects from the dataset. (Left) Demonstrated trajectory, (middle), and (right) robot imitations in two different scenes. See the supplementary video for detailed results.

TABLE II

PERFORMANCE ANALYSIS OF VARIOUS SKILLS.

Skill	Success rate (%)	Imitation time ± STD (s)
D1	75.0	166.75 ± 4.72
D2	100	115.00 ± 4.24
D3	100	79.00 ± 9.88
D4	100	35.22 ± 3.89

along with screenshots of the robot during imitation. It can be seen that trajectory constraints are maintained from demonstration, and even when the final pose differs from the initial configuration, the robot follows the imitated path. The robot occasionally rotates the object, as observed with the screwdriver in the drawing skill last column, due to the FMC not accounting for object symmetry allowing alignment in either direction. Note that this is not considered a failure, though future work could include symmetry analysis to improve the target end configuration. Table II presents the average success rate, which also looks for stably grasping/touching the correct region of the object, and the average skill imitation time across four trials. All tasks were successfully accomplished with every object in our dataset, except for one failure: during a wiping trial (D1), a misalignment while grasping the wooden block (last row in Fig. 3) caused the gripper to collide with the edge, rotating the block.

VI. Conclusion

In this paper, we demonstrated GIFT, a one-shot skill transfer method that effectively generalises to unseen category-level objects. Beyond enabling efficient transfer from a single demonstration, our approach highlights the applicability of embedding geometric priors directly into the representation of manipulation skills. This design choice provides stronger guarantees of constraint fidelity and interpretability compared to purely end-to-end methods. By leveraging functional map correspondences and screw-based trajectory generation, GIFT achieves reliable performance in multi-object scenes while remaining computationally efficient and data-light. For future works, we aim to integrate GIFT with large-scale visuomotor foundation models to further bridge the gap between robust geometric grounding and the semantic flexibility of data-driven policies.

References

[1] A. Brohan *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[2] O. X.-E. Team *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," *arXiv preprint arXiv:2310.08864*, 2023.

[3] M. Kim *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[4] O. M. Team, "Octo: An open-source generalist robot policy," in *Robotics: Science and Systems (RSS)*, 2024.

[5] C. Chi, J. Xu, V. Tzoumas *et al.*, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, 2024.

[6] X. Hu *et al.*, "Adaflow: Imitation learning with variance-adaptive flow-based policies," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[7] C. De Farias, B. Tamadazte, R. Stolkin *et al.*, "Grasp transfer for deformable objects by functional map correspondence," in *2022 IEEE Int. Conf. on Rob. and Auto.*, 2022, pp. 735–741.

[8] L. Manuelli, W. Gao, P. Florence *et al.*, "kpam: Keypoint affordances for category-level robotic manipulation," in *The Int. Symp. of Rob. Resear.*, 2019, pp. 132–157.

[9] W. Gao and R. Tedrake, "kpam 2.0: Feedback control for category-level robotic manipulation," *IEEE Rob and Auto. Lett.*, vol. 6, no. 2, pp. 2962–2969, 2021.

[10] C. de Farias, B. Tamadazte, M. Adjigble *et al.*, "Task-informed grasping of partially observed objects," *IEEE Rob. and Auto. Lett.*, vol. 9, no. 10, pp. 8394–8401, 2024.

[11] M. Ovsjanikov, M. Ben-Chen, J. Solomon *et al.*, "Functional maps: a flexible representation of maps between shapes," *ACM Trans. on Graph.*, vol. 31, pp. 1–11, 2012.

[12] F. Allmendinger, S. Charaf Eddine, and B. Corves, "Coordinate-invariant rigid-body interpolation on a parametric c^1 dual quaternion curve," *Mechanism and Machine Theory*, pp. 731–744, March 2018.

[13] R. Laha, R. Sun, W. Wu *et al.*, "Coordinate invariant user-guided constrained path planning with reactive rapidly expanding plane-oriented escaping trees," in *Int. Conf. on Rob. Sys.*, 2022, pp. 977–984.

[14] L. Figueredo, "Kinematic control based on dual quaternion algebra and its application to robot manipulators," Ph.D. dissertation, University of Brasilia (UnB), 2016.

[15] D.-A. Huang, Y.-W. Chao, C. Paxton *et al.*, "Motion reasoning for goal-based imitation learning," in *IEEE Int. Conf. on Rob. and Auto.*, 2020, pp. 4878–4884.

[16] A. Bonardi, S. James, and A. J. Davison, "Learning one-shot imitation from humans without humans," *IEEE Rob. and Auto. Lett.*, vol. 5, no. 2, pp. 3533–3539, 2020.

[17] F. Torabi, G. Warnell, and P. Stone, "Imitation learning from video by leveraging proprioception," *arXiv preprint arXiv:1905.09335*, 2019.

[18] Y. Liu, A. Gupta, P. Abbeel *et al.*, "Imitation from observation: Learning to imitate behaviors from raw video via context translation," in *IEEE Int. Conf. on Rob. and Auto.*, 2018, pp. 1118–1125.

[19] A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *IEEE Int. Conf. on Rob. and Auto.*, 2002.

[20] A. J. Ijspeert, J. Nakanishi, H. Hoffmann *et al.*, "Dynamical movement primitives: learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–373, 2013.

[21] S. Calinon, "Learning from demonstration (programming by demonstration)," *Encyclopedia of Rob.*, pp. 1–8, 2018.

[22] A. Paraschos, C. Daniel, J. R. Peters *et al.*, "Probabilistic movement primitives," *Adv. in Neural Info. Proc. Sys.*, vol. 26, 2013.

[23] J. Vorndamme, J. Carvalho, R. Laha *et al.*, "Integrated bi-manual motion generation and control shaped for probabilistic movement primitives," in *IEEE-RAS Int. Conf.on Humanoid Rob.*, 2022, pp. 202–209.

[24] R. Laha, A. Rao, L. F. Figueredo *et al.*, "Point-to-point path planning based on user guidance and screw linear interpolation," in *Int. Design Engin. Tech. Conf. and Comput. and Info. in Eng.*, vol. 85451, 2021, p. V08BT08A010.

[25] B. Wen, W. Lian, K. Bekris *et al.*, "Catgrasp: Learning category-level task-relevant grasping in clutter from simulation," in *2022 IEEE Int. Conf. on Rob. and Auto.*, 2022, pp. 6401–6408.

[26] A. Tekden, M. P. Deisenroth, and Y. Bekiroglu, "Grasp transfer based on self-aligning implicit representations of local surfaces," *IEEE Robot. Autom. Lett.*, vol. 8, no. 10, pp. 6315–6322, 2023.

[27] A. Simeonov, Y. Du, A. Tagliasacchi *et al.*, "Neural descriptor fields: Se(3)-equivariant object representations for manipulation," in *2022 IEEE Int. Conf. on Rob. and Auto.*, 2022.

[28] A. Simeonov, Y. Du, Y.-C. Lin *et al.*, "Se(3)-equivariant relational rearrangement with neural descriptor fields," in *Conf. on Rob Learning*, vol. 205. PMLR, 2023, pp. 835–846.

[29] N. Di Palo and E. Johns, "Dinobot: Robot manipulation via retrieval and alignment with vision foundation models," *arXiv preprint arXiv:2402.13181*, 2024.

[30] E. Chisari *et al.*, "Learning robotic manipulation policies from point clouds with conditional flow matching," in *Conference on Robot Learning (CoRL)*, 2024.

[31] W. Huang, C. Wang, Y. Li *et al.*, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," *arXiv preprint arXiv:2409.01652*, 2024.

[32] K. Zakka, Z. Xu, X. Lin *et al.*, "Aloha: A low-cost teleoperation system for general-purpose robot manipulation," *arXiv preprint arXiv:2305.03382*, 2023.

[33] M. Meyer, M. Desbrun, P. Schröder *et al.*, "Discrete differential-geometry operators for triangulated 2-manifolds," in *Visu. and Math. III*, 2003, pp. 35–57.

[34] M. Aubry, U. Schlickewei, and D. Cremers, "The wave kernel signature: A quantum mechanical approach to shape analysis," in *IEEE Int. Conf. on Comput. Vision Workshops*, 2011, pp. 1626–1633.

[35] M. Ovsjanikov, E. Corman, M. Bronstein *et al.*, "Computing and processing correspondences with functional maps," in *SIGGRAPH ASIA 2016 Courses*, 2016, pp. 1–60.

[36] S. Melzi, J. Ren, E. Rodola *et al.*, "ZoomOut: Spectral Upsampling for Efficient Shape Correspondence," *ACM Trans. on Graph.*, 2019.

[37] A. Sarker, A. Sinha, and N. Chakraborty, "On screw linear interpolation for point-to-point path planning," in *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 2020, pp. 9480–9487.

[38] N. Marturi, M. Kopicki, A. Rastegarpanah *et al.*, "Dynamic grasp and trajectory planning for moving objects," *Auto. Rob.*, vol. 43, no. 5, pp. 1241–1256, 2019.

[39] M. Ester, H.-P. Kriegel, J. Sander *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Int. Conf. on Knowledge Discovery and Data Mining*, 1996, p. 226–231.

[40] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Trans. on Graph.*, vol. 32, no. 3, pp. 1–13, 2013.

[41] M. Adjigble, N. Marturi, V. Ortenzi *et al.*, "Model-free and learning-free grasping by local contact moment matching," in *IEEE/RSJ Int. Conf. on Intell. Rob. and Sys.*, 2018, pp. 2933–2940.

[42] M. Adjigble, C. de Farias, R. Stolkin *et al.*, "Spectgrasp: Robotic grasping by spectral correlation," in *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, 2021, pp. 3987–3994.

[43] Y. Bekiroglu, N. Marturi, M. A. Roa *et al.*, "Benchmarking protocol for grasp planning algorithms," *IEEE Rob. and Auto. Lett.*, vol. 5, no. 2, pp. 315–322, 2020.

[44] B. V. Adorno and M. M. Marinho, "Dq robotics: A library for robot modeling and control," *IEEE Rob. & Auto. Mag.*, vol. 28, no. 3, pp. 102–116, Sep. 2021.