This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

K-VIL: Keypoints-Based Visual Imitation Learning

Jianfeng Gao, Zhi Tao, Noémie Jaquier, *Member, IEEE*, and Tamim Asfour, *Senior Member, IEEE*

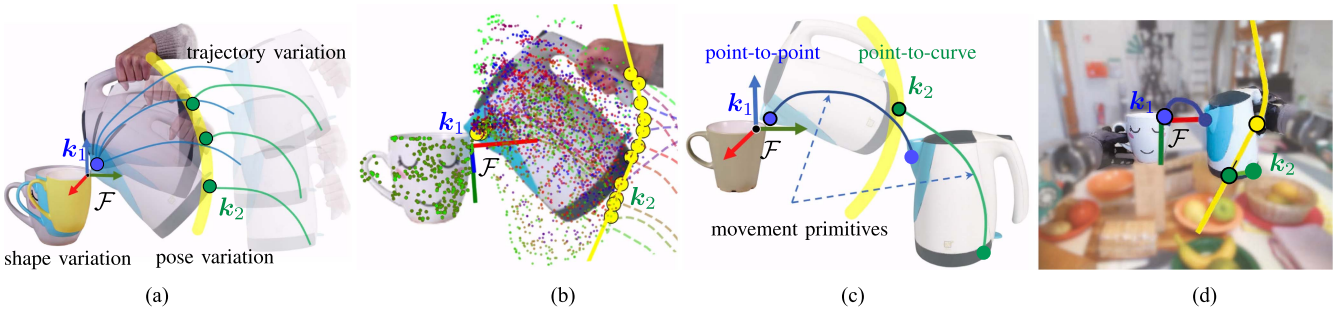


Fig. 1. Overview of the K-VIL approach. (a) Human demonstration videos of manipulation actions involving categorical objects with shape, pose, and trajectory variations. (b) Sampling of dense candidate points from the object surface. (c) Extraction of *sparse keypoints* $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ subject to certain types of *geometric constraints* (point-to-point and point-to-curve), their associated *local frames* $\mathcal{F}$, and the *movement primitives*, which represent the demonstrated keypoint motions. (d) Adaptation of the learned generalizable geometric task representation to a new scene and execution by the robot.

***Abstract*—Visual imitation learning provides efficient and intuitive solutions for robotic systems to acquire novel manipulation skills. However, simultaneously learning geometric task constraints and control policies from visual inputs alone remains a challenging problem. In this article, we propose the *keypoint-based visual imitation learning* (K-VIL) approach that automatically extracts sparse, object-centric, and embodiment-independent task representations from a small number of human demonstration videos. The task representation is composed of keypoint-based geometric constraints on principal manifolds, their associated local frames, and the movement primitives that are then needed for the task execution. Our approach is capable of extracting such task representations from a single-demonstration video and of incrementally updating them when new demonstrations are available. To reproduce manipulation skills using the learned set of prioritized geometric constraints in novel scenes, we introduce a novel keypoint-based admittance controller. We evaluate our approach in several real-world applications, showcasing its ability to deal with cluttered scenes, viewpoint mismatch, new instances of categorical objects, and large object pose and shape variations. Our evaluation demonstrates the efficiency and robustness of our approach in both one-shot and few-shot imitation learning settings.**

***Index Terms*—Learning from demonstration, learning of geometric constraints, manipulation planning, visual learning.**

Manuscript received 20 February 2023; accepted 12 May 2023. This work was supported in part by the German Federal Ministry of Education and Research (BMBF) under the Project OML and in part by Carl Zeiss Foundation under the Project JuBot. This paper was recommended for publication by Associate Editor Gionata Salvietti and Editor Sven Behnke upon evaluation of the reviewers' comments. *(Corresponding author: Jianfeng Gao.)*

The authors are with the Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany (e-mail: jianfeng.gao@kit.edu; zhitao.robotics@gmail.com; noemie.jaquier@kit.edu; asfour@kit.edu).

Videos and source code are available at https://sites.google.com/view/k-vil.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TRO.2023.3286074.

Digital Object Identifier 10.1109/TRO.2023.3286074

I. Introduction

OBSERVATIONAL learning, i.e., the ability to develop new skills from observed actions and their outcome, is an important learning mechanism in our daily lives [1], [2], [3]. For example, by watching a few videos showing people pouring water from a kettle into different teacups [as in Fig. 1(a)], we can easily learn "what" a pouring task is and "how" to perform it. From a computational point of view, the spout and bottom of the kettle can be represented by two keypoints $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$. As shown in Fig. 1(c), a pouring task then simply consists in aligning the spout $\boldsymbol{k}_1$ with a point above the rim of the cup (point-to-point constraint) and similarly aligning the bottom $\boldsymbol{k}_2$ with a curve that controls the kettle's angle of inclination (point-to-curve constraint). Such sets of keypoints and keypoint-based geometric constraints can generally be used to represent daily manipulation tasks, i.e., to parameterize the motion of their functional parts relative to some local frames of reference. Moreover, these keypoints and local frames can also be associated with local visual features of object functional parts.

In this article, we propose to exploit such task representations to teach manipulation skills to robots from video demonstrations. We additionally aim for generalizable skills, which can be reused in novel scenes (see Fig. 1(d) for an example). In this context, three main challenges arise, namely:

1) the detection and efficient extraction of task-relevant keypoints on objects;
2) the definition of generalizable and embodiment-independent task representations;
3) the reproduction of the demonstrated task and its adaptation to new scenes.

To address the first two challenges, we first *densely* sample a set of candidate points from the object mask provided by mask region-based convolutional neural network (R-CNN) [4]

1552-3098 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

and leverage the correspondence detection of dense object net (DON) [5] to track their motions in the demonstration videos and obtain their three-dimensional (3-D) positions (see Section III for a short background). Then, we jointly extract a set of *sparse keypoints* and a set of *keypoint-based geometric constraints* representing the task, as shown in Fig. 1(b) (see Section IV). To do so, we exploit principal manifold estimation (PME) algorithms [6], which are intrinsically more data- and time-efficient than approaches based on supervised learning [7], [8] and reinforcement learning (RL) [9], [10]. The resulting geometric constraints are expressed relative to local frames defined on target objects and are easily adjustable to pose and shape variations of the target object. As shown in previous works [7], [9], [11], such object-centric task representations facilitate the transfer of manipulation skills between demonstrators and imitators. It also allows our approach to deal with demonstrations provided from different viewpoints. The representation of the task in the form of keypoints and their constraints also enables the use of simpler control policies [12], such as movement primitives, for executing the given task. In this article, we exploit this property to address our third challenge. Namely, we encode the keypoint motions relative to the corresponding local frames as via-point movement primitives (VMPs) [13], which are flexible in terms of temporal scaling and trajectory adaptation while maintaining the demonstrated motion styles. The learned keypoint motions can then be executed on a robot by leveraging our novel keypoint-based admittance controller (KAC) (see Section V). We validate our approach by learning various real-world daily tasks from video demonstrations and reproducing them with a humanoid robot (see Section VI). The results show that K-VIL efficiently extracts generalizable manipulation skills, handles viewpoint mismatch, and deals with large pose and shape variations of categorical objects in cluttered scenes.

Our contributions are threefold.

1) We introduce the *keypoint-based visual imitation learning* (K-VIL) approach for automatic and incremental extraction of sparse, object-centric, viewpoint-invariant, and embodiment-independent task representations. K-VIL extracts task representations from a single-demonstration video and improves them as new demonstrations are available. The task representations consist of keypoint-based geometric constraints on principal manifolds, their associated local frames, and the movement primitives required to reproduce the task.
2) We formulate and learn a large variety of geometric constraints, which allow the proposed task representation to be flexible and efficient.
3) We propose a novel KAC that handles a set of prioritized geometric constraints and allows successful reproductions of the learned task in novel scenes.

II. Related Work

Visual imitation learning (VIL) is a class of imitation learning (IL) frameworks in which only visual sensory input is presented to the imitator. The main challenges of VIL are as follows:

1) the detection of visual correspondences between the demonstrator's and imitator's context, i.e., context translation [14], [15];
2) the fine-grained understanding of scene structures [16], along with the design of generalizable task representation;
3) the design of sample efficient and scalable control policies.

This latter challenge is often tackled along with the former ones, as described next.

A. Context Translation

Context translation has typically been addressed by training context translators in the demonstrator context to predict the observations in the imitator (e.g., the robot) context. Pixel-level translators were used in [15], [17], [18], [19] to further train RL policies by maximizing the similarity between predicted and received robot observations. Despite the performance of such models, their training is computationally expensive and time-consuming. To improve learning efficiency, Sharma et al. [14] combined a goal-level translator with a task-agnostic control policy, which was trained independently and shared among different tasks. In contrast to these works, K-VIL represents the context via a set of object-centric keypoints and their respective geometric constraints, thus facilitating the context translation between demonstrators and imitators. In addition, by leveraging Mask R-CNN and DON models—which are trained beforehand in a task-agnostic manner and shared among tasks—K-VIL's representations can be acquired from a single or few demonstrations.

B. Fine-Grained Understanding of Scene Structure

The above approaches do not scale to categorical objects as they do not explicitly extract the scene structures with respect to objects and their functional parts. In the literature, the understanding of fine-grained scene structures is mainly achieved through the following:

1) the viewpoint-invariant representation of fine-grained scene features;
2) the extraction of a hierarchy of the scene structure;
3) the definition of task constraints.

1) Viewpoint-Invariant Representation: Dense visual descriptors, such as DON [5] and neural descriptor fields (NDFs) [20], represent fine-grained scene features by detecting dense correspondences of categorical objects, thus allowing point-based representation of object functional parts. However, in [20], [21], [22], access to the robot state space was required in addition to the visual demonstrations, thus violating the purpose of visual imitation. Yang et al. [23] proposed a transporter-based representation learning model to extract keypoints from the task-agnostic human and robot play data. Building the similarity function of such a model requires robot execution videos with a similar view setup as the demonstration videos. The same requirement applies to the approaches presented in [24], [25], [26] and prevents robots from learning from human demonstrations taken from a very different viewpoint. Sermanet et al. [27] proposed time-contrastive networks to learn viewpoint-invariant latent representations of the scene. This approach requires a

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

large number of demonstration videos and robot play videos to build the correspondence between human and robot arms, which makes the approach embodiment-dependent. Similar to our article, Karnan et al. [28] proposed to leverage task-agnostic keypoint detection algorithms for vehicle navigation tasks. This approach requires storing the demonstration video and searches the closest demonstration image for reward construction. This reduces the number of demonstrations compared with the pixel-level context translations of the articles presented in [15], [17], [18], [19]. However, by overlooking the different types of geometric constraints that the keypoints are subject to, this approach suffers from averaging problem similar to the article presented in [20] (see Section II-B3 for details). In contrast, K-VIL uses dense point-based object representation and correspondence detection to align demonstrations recorded from different viewpoints. This significantly reduces the required number of demonstrations. Moreover, K-VIL explicitly extracts viewpoint- and embodiment-independent scene structure and task constraints, thus addressing the average problem and achieving better extrapolation capability in fine-grained manipulation tasks.

2) Hierarchy of Scene Structure: The variance across demonstrations was used to efficiently select appropriate local frames from some candidates in several IL frameworks as a solution to extract hierarchical scene structure [29], [30]. Representing the learned task in such local frames was shown to facilitate the transfer of skills between different embodiments and the design of control policies. However, in the absence of visual sensory input, the candidates were manually defined at object level in [29], [30]. In our work, we instead show that combining dense visual descriptors and a variance-based criterion allows for the efficient extraction of keypoints and local frames at a fine-grained level.

3) Task Constraints: Early works on visual servoing [31], [32], [33] hand-crafted task constraints are given as simple geometric constraints (e.g., point-to-point and point-to-line). To represent more complex constraints, Sieb et al. [16] proposed visual entity graphs (VEGs) based on DON to disentangle the scene structure into multiple levels, including objects, parts, and points. A path integral policy was then trained on the similarity loss between the VEGs learned from the demonstrator and the VEGs observed by the robot. The task constraints are implicitly learned in VEGs, similarly to the neural pose descriptors in [20], and are, therefore, averaged when large shape or pose variations occur in the demonstrations. To address this issue, the authors in [12], [34] introduced an explicit representation of geometric constraints using visual geometric skill kernels and graph neural networks, which generalized better to categorical objects. However, this approach requires ∼30 demonstrations to learn a generalizable representation since both task correspondences and geometric constraints need to be learned in the graph structure. In this article, we exploit the correspondence detection of DON and the variation information to jointly extract explicit sparse keypoints and endow them with geometric constraints of various types (see Section IV-B). This allows us to learn generalizable skills from only a few demonstration videos while alleviating the averaging problem of the article presented in [16]. Moreover, our task representation allows us to replace the RL policy used, e.g., in [16], with simpler movement primitives that reproduce the demonstrated keypoint motions and adapt to new goal configurations.

III. Background

In this section, we introduce the dense visual correspondence models, the PME algorithm, and the VMPs, which are essential building blocks of K-VIL.

A. Dense Visual Correspondence

DON [5] maps a color image $\boldsymbol{A} \in \mathbb{R}^{W \times H \times C}$ to a dense descriptor image $\boldsymbol{A}_{\bar{D}} \in \mathbb{R}^{W \times H \times \bar{D}}$, where W, H, and C denote the width, height, and the number of channels of the image, and $\bar{D}$ is the dimension of the descriptor space. Therefore, each pixel of the input image is represented by a $\bar{D}$-dimensional descriptor $\boldsymbol{d} \in \mathbb{R}^{\bar{D}}$. To train a DON model on an object category, multiple views of posed RGB images of multiple instances of this object category are first collected. The object meshes are then reconstructed using any state-of-the-art scene reconstruction method. The reconstructed meshes are then exploited to automatically acquire object masks and retrieve dense correspondence signals, which are used to train the model. A fully trained DON model maps similar local patches of two images of the categorical objects to patches in the descriptor space with similar descriptors. In other words, the dense visual correspondence between two pixels is detected if the distance between their descriptors is smaller than a certain threshold. For example, the spout of the kettle in different image frames is mapped to similar descriptors.

In this article, we aim at retrieving dense correspondences among different instances of the same object category. Moreover, once a set of sparse keypoints is extracted, we aim at identifying these keypoints during the task reproduction on new instances of the same object category using their descriptors. To do so, we construct a correspondence function $\boldsymbol{p} = f_c(\boldsymbol{A}, \boldsymbol{d})$ using the DON model and the camera intrinsic and extrinsic parameters, which can be used to extract the 3-D position of a keypoint represented by the descriptor $\boldsymbol{d}$. Similarly to the articles presented in [5], [35], we use 34-layer, stride-8 ResNet as the DON model and set $\bar{D} = 3$. We also train Mask R-CNN models [4] with the automatically generated object mask dataset similar to the article presented in [16]. We refer the interested readers to the articles presented in [5], [35] for additional details of DON models. For the case where the human hand is involved in the demonstrated tasks, we treat it as a special object and utilize a hand keypoint detection algorithm, e.g., MediaPipe [36], which provides more robust correspondence detection on human hands.

B. Principal Manifold Estimation

Within K-VIL, we are interested not only in extracting a set of keypoints but also in learning the constraints that they satisfy in order to represent the task. Specifically, we use geometric constraints, which allow us to restrict the keypoint target positions, e.g., as in Fig. 1(c). In a 3-D space, a simple geometric constraint

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

can be viewed as a low-dimensional manifold corresponding to a point, a line, a plane, a curve, or a surface. In this article, we leverage the PME algorithm [6] to uncover the geometric constraints as low-dimensional embedding from a set of 3-D points varying in time (obtained via DON). In PME, the principal manifold is defined as a minimum of the functional with a regularity penalty term derived on a Sobolev space. Specifically, the PME algorithm minimizes the loss

$$\mathcal{L}(f, \pi_d) = \mathbb{E}\left\|\boldsymbol{x} - f(\pi_d(\boldsymbol{x}))\right\|^2 + \lambda\|\kappa_f\|^2 \tag{1}$$

where $\pi_d : \mathbb{R}^D \to \mathbb{R}^d$ is the projection index that maps a random D-dimensional vector $\boldsymbol{x}$ onto a d-dimensional principal manifold with $d < D$, $f : \mathbb{R}^d \to \mathbb{R}^D$ is the reconstruction function, $\|\kappa_f\|^2$ represents the high-dimensional generalization of the total squared curvature of the principal manifold, and $\lambda \in [0, \infty)$ controls the model complexity. Therefore, the former term of the loss represents the reconstruction error, while the latter regularizes the model to avoid overfitting. Note that the PME reduces to linear principal component analysis (PCA) when $\lambda \to \infty$. The linearity and the dimension d of the principal manifold determine the subspace type. For example, a nonlinear principal manifold of dimension $d = 1$ is a principal curve and corresponds to a curve constraint. We refer the reader to the article presented in [6] for the details of the PME algorithm.

C. Via-Point Movement Primitive

In addition to extracting the keypoint constraints, we are interested in learning their motions from human demonstration videos. In IL, motions are often represented by movement primitives. Here, we use VMPs [13]. A VMP combines a linear elementary trajectory h_{vmp} with a nonlinear shape modulation f_{vmp} so that

$$y(x) = h_{\text{vmp}}(x) + f_{\text{vmp}}(x) = g + x(y_0 - g) + \boldsymbol{\psi}(x)^{\mathsf{T}}\boldsymbol{w}$$

where x is the canonical variable decreasing linearly from 1 to 0, y and y_0 are the current and start positions, and g is the target position. The shape modulation term is defined as a linear regression model based on N_k squared exponential (SE) kernels $\psi_i(x) = \exp(-h_i(x - c_i)^2), i \in [1, N_k]$, where h_i and c_i are the predefined constants. Similarly to probabilistic movement primitives (ProMP) [37], VMPs assume that the weight parameter $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{w}}, \boldsymbol{\Sigma}_{\boldsymbol{w}})$ follows a Gaussian distribution, and thus can be learned via maximum likelihood estimation. VMPs provide enhanced extrapolation capability compared with ProMP, as they handle via-points (including start and target positions) adaptation to points that lie out of the demonstrated distributions. In this article, we leverage VMPs to learn the demonstrated motion styles of each keypoint and to adapt the corresponding trajectories to via points identified using the dense correspondence function f_c of DON. In contrast to control policies based on RL (e.g., [16]) or on visual servoing (e.g., [12]), VMP-based control policies endow K-VIL with flexible temporal scaling and reliable via-point adaptation.

IV. Keypoint-Based VIL

In this section, we present the proposed K-VIL approach. Given N demonstration videos $\mathcal{V} = \{V_n\}_{n=1}^N$ of a task in D-dimensional task space, where $D \in \{2, 3\}$, K-VIL first preprocesses the RGB-D videos and generates the data required for learning the task. This includes densely sampled candidate points, their descriptors and trajectories, the spatial properties and roles of the objects, as well as all potential local frames (see Section IV-A). A sparse set of keypoints and their geometric constraints are then estimated via *principal constraint estimation* (PCE). As detailed in Section IV-B, our proposed PCE first extracts a set of keypoints and their geometric constraints by leveraging PME algorithms. These algorithms rely on observed distances as well as on the demonstration variability when several demonstrations are provided. For the cases where the resulting set contains redundant selections of keypoints, our PCE then leverages hierarchical agglomerative clustering (HAC) to resolve this redundancy and obtain a final set $\mathcal{P}$ of sparse keypoints. As explained in Section IV-C, $\mathcal{P}$ is then used to extract the task representation consisting of a set of keypoints defined by visual descriptors $\mathcal{D} = \{\boldsymbol{d}_l\}_{l=1}^L$, their associated geometric constraints $\mathcal{C} = \{C_l\}_{l=1}^L$, and the weights of the VMPs $\Omega = \{\boldsymbol{w}_l\}_{l=1}^L$, which are then exploited to reproduce the keypoint motions. The extracted task representation is finally used by the KAC presented in Section V to reproduce the demonstrated skill on the robot. The proposed K-VIL approach is shown in Fig. 2, and its different steps are detailed next. The main notations are listed in Table I.

A. Preprocessing

As previously mentioned, K-VIL first preprocesses the RGB-D videos $\mathcal{V}$ provided as demonstrations. This is achieved via the following five steps, also depicted in Fig. 2.

1) Sampling of Candidates: First, we query the list of objects' categories $\mathcal{O} = \{O_i\}_{i=1}^I$ involved in the task by feeding the mask R-CNN model with an image randomly sampled from $\mathcal{V}$. From the *visible region* of each object O_i, P_i candidate points are then densely and uniformly sampled and form a set $\mathcal{P}_i$. Each candidate point is a potential keypoint or a potential origin of a local frame and may later be selected as such by K-VIL. Note that we here assume that all relevant points are located in the region on the object surface that is always visible to the imitator [see Fig. 3(a)]. We denote the set of all candidate points from all objects as $\mathcal{P}_c = \bigcup_{i=1}^I \mathcal{P}_i$. Their corresponding deep visual feature descriptors are derived from DON [5] as $\mathcal{D}_c = \{\boldsymbol{d}_h\}_{h=1}^H$ with $\boldsymbol{d}_h \in \mathbb{R}^3$ and $H = |\mathcal{P}_c|$ the cardinality of $\mathcal{P}_c$.

2) Trajectories of Candidate Points: We extract the task-space trajectory of all candidate points from the videos using the DON-based correspondence function $f_c(\cdot, \cdot)$ (see Section III-A), which finds the correspondence pixel of the candidates and maps them to 3-D coordinates in the camera local frame. The obtained trajectories are then smoothed and normalized in time with T time steps. We obtain a set $\mathcal{T} = \{\boldsymbol{\tau}_h\}_{h=1}^H$ of trajectories of all candidates, where $\boldsymbol{\tau}_h \in \mathbb{R}^{N \times T \times D}$ denotes the trajectory of the hth candidate point. These trajectories are used in the

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

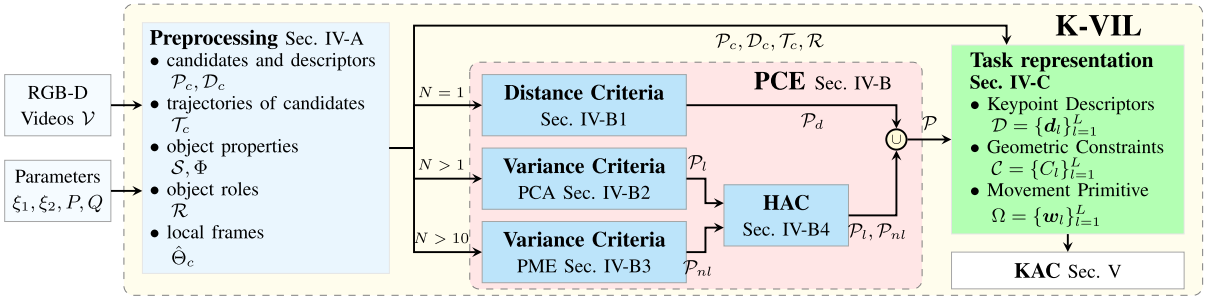


Fig. 2. Overview of K-VIL's architecture. After preprocessing the demonstration videos, K-VIL jointly extracts a set of sparse keypoints, a set of keypoint-based geometric constraints, and a set of movement primitive parameters that fully represent the task. The robot then leverages the proposed KAC to reproduce the task in novel scenes.

TABLE I
SUMMARY OF K-VIL'S NOTATIONS

Notation	Meaning	Notation	Meaning
C	geometric constraint	ξ_1, ξ_2	lower and upper thresholds of spatial variability
d	the intrinsic dimension of a principal manifold	λ	the regularization factor of PME
$D, \bar{D}$	the dimension of the task space, descriptor space	κ_f	the curvature of the principal manifold
g_1, g_2	force scaling parameters	$\boldsymbol{d}$	the descriptor vector of a keypoint / candidate
H	the total number of candidate points	$\boldsymbol{f}$	a force vector
I	the number of objects	$\boldsymbol{h}_c$	the Coriolis and gravitational force in task space
L	the number of constraints	$\boldsymbol{k}, \dot{\boldsymbol{k}}$	the position and velocity vector of a keypoint
N	the number of demonstrations	$\boldsymbol{p}$	a position vector of a point
P	the number of points sampled on an object	$\boldsymbol{w}, \boldsymbol{\Sigma}_{\boldsymbol{w}}$	a weights vector of the VMP and its covariance
Q	the number of neighboring points	$\boldsymbol{\mu}_{\boldsymbol{w}}$	the mean of $\boldsymbol{w}$
$\mathcal{C}$	the set of constraints	$\boldsymbol{\nu}$	the explained variance
$\mathcal{D}$	the set of descriptors	$\boldsymbol{\eta}$	the spatial variability
$\mathcal{F}$	a local frame	$\boldsymbol{K}$	diagonal stiffness, damping and inertia matrices
$\hat{\theta}_{\mathcal{F}}$	the configuration of a canonical local frame	$\boldsymbol{S}$	the canonical shape
$\hat{\Theta}_{\mathcal{F}}$	the set of canonical local frame configurations	$\boldsymbol{\tau}$	a trajectory
$\mathcal{M}$	a manifold	$\sigma(\cdot)$	the density force on the d-dimensional manifold
$O, \mathcal{O}$	an object category, the set of objects	$\nabla\sigma(\cdot)$	the density field
$\mathcal{P}$	the set of keypoints / candidate points	$f_c(\cdot,\cdot)$	the correspondence function
$\gamma, \mathcal{R}$	the role of the object, the set of object roles	$f(\cdot)$	the projection index of a principal manifold
$\mathcal{S}$	the set of canonical shapes of all objects	$\pi_d(\cdot)$	the reconstruction function of a principal manifold
$\mathcal{V}$	the set of demonstration videos	$\boldsymbol{\psi}(\cdot)$	the SE kernels in VMP
φ, Φ	the spatial scale, the set of spatial scales	$\psi_i(\cdot)$	the SE kernel

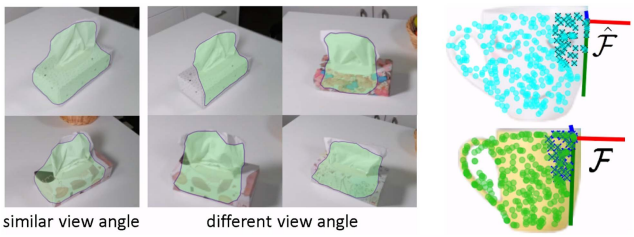


Fig. 3. (a) Illustration of the visible region of a tissue box. (b) Example of local frame matching on two cups. The canonical shape of the cup category (*top*) is defined on a white cup by the positions of its candidate points (●). The correspondence points (●) of the canonical shape are detected on a yellow cup (*bottom*). A canonical local frame $\hat{\mathcal{F}}$ is parametrized by $Q = 50$ neighboring candidates (×). These candidates are then used to find the same local frame $\mathcal{F}$ on the yellow cup.

remaining preprocessing steps and in Section IV-B to extract keypoints and geometric constraints.

3) Object Properties: We define the *canonical shape* $\boldsymbol{S}_i \in \mathbb{R}^{P_i \times D}$ of each object category O_i as the positions of all candidates on the object at the first time step of the first demonstration. This notion of canonical shape is illustrated for a cup in Fig. 3(b). Moreover, we define the *spatial scale* $\varphi_i \in \mathbb{R}$ as the maximum distance between each pair of candidates on the canonical shape, which will be used in Section IV-B2 to determine object-independent thresholds.

4) Object Roles: Geometric constraints do not suffice to entirely represent a task. For example, one of the constraints of a pouring task is the kettle–cup alignment, which could be achieved by moving the cup toward a static kettle. Instead, pouring requires a motion of the kettle. K-VIL addresses this issue by considering the role of the objects for the task at hand. Namely, we detect object motion saliency similarly to the article presented in [29] to determine the role of the objects $\mathcal{R} = \{\gamma_i\}_{i=1}^{I}$, where $\gamma_i \in \{\mathsf{master}, \mathsf{slave}\}$. The master O_m is the object with the lowest average variance of candidates' trajectory, while other objects are slaves O_s. K-VIL accounts for the objects' roles by constructing local frames only on the master and extracting keypoints only on the slaves. Therefore, we split the set $\mathcal{P}_c$ of all candidates by the object roles to $\mathcal{P}_m$ and $\mathcal{P}_s$, denoting the set of candidates on master and slave objects, respectively. Similarly, $\mathcal{D}_c$ is splitted to $\mathcal{D}_m$ and $\mathcal{D}_s$.

5) Local Frame Detection: As previously mentioned, K-VIL aims at representing the demonstrated task from an object-centric perspective. This is achieved by defining local frames on

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

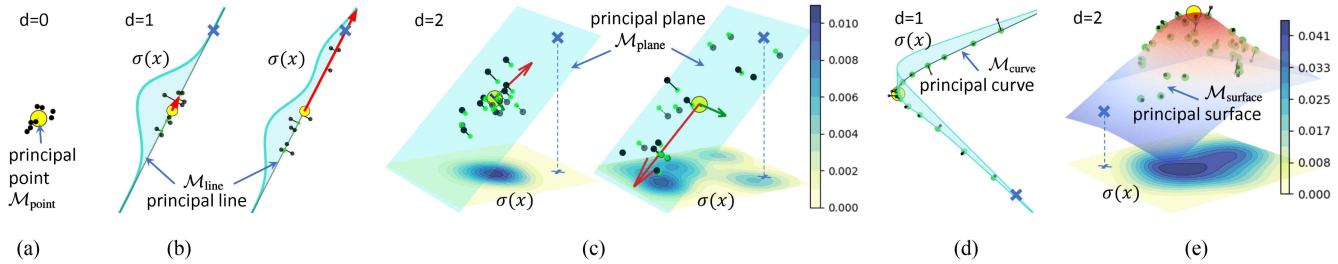


Fig. 4. Five types of geometric constraints. The constraints are obtained from candidate points (●) from N demonstrations. The density function $\sigma(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^d$ is estimated from the projections (●) of the candidate positions on the d-dimensional principal manifold. We also depict the mean $\boldsymbol{p}_m$ (●), the spatial variability (⟶, ⟶) on the principal manifold along the principal components, the stress vector $\boldsymbol{s}$ (—), and the examples of extrapolation of the keypoints (×) on the manifolds. (a) p2p. (b) p2l. (c) p2P. (d) p2c. (e) p2S.

the master object. To do so, we initially define one canonical local frame $\hat{\mathcal{F}}_j$ equal to identity for each candidate point $j \in \mathcal{P}_m$ of the master canonical shape (see Fig. 3(b)-*top*). Each local frame $\hat{\mathcal{F}}_j$ is assigned the Q closest candidates to j, whose positions in $\hat{\mathcal{F}}_j$ are denoted as the reference values $\boldsymbol{p}_q^*$. In other words, $\hat{\mathcal{F}}_j$ is parameterized by $\hat{\vartheta}_{\hat{\mathcal{F}}_j} = \{\{\boldsymbol{d}_q\}_{q=1}^Q, \{\boldsymbol{p}_q^*\}_{q=1}^Q\}$, where $\boldsymbol{d}_q$ are the descriptors of the Q neighboring candidates. These neighboring candidates are then used to detect the same local frame on another instance of the same object category at a different time t (see Fig. 3(b)-*bottom*). Namely, the local frame $\mathcal{F}_j$ is detected by minimizing the mean squared displacement of the observed coordinates $\{\boldsymbol{p}_q(t)\}_{q=1}^Q$ of the neighboring candidates at time t with their reference values $\{\boldsymbol{p}_q^*\}_{q=1}^Q$, i.e.,

$$\mathcal{F}_j(t) = \arg\min_{\mathcal{F}} \sum_{q=1}^{Q} \left\| \boldsymbol{p}_q^* - \boldsymbol{p}_q(t) \right\|^2 .$$

The set of local frames on the master object is denoted as $\hat{\Theta}_{\mathcal{F}} = \{\hat{\vartheta}_{\mathcal{F}_j} : j \in \mathcal{P}_m\}$.

In summary, the preprocessed data for all objects contain the sets $\mathcal{P}_m$ and $\mathcal{P}_s$ of the candidate points on master and slave objects, respectively, their corresponding descriptors $\mathcal{D}_m$ and $\mathcal{D}_s$ and trajectories $\mathcal{T}$, the set $\mathcal{S} = \{\boldsymbol{S}_i\}_{i=1}^I$ of the object canonical shapes, the set $\Phi = \{\varphi_i\}_{i=1}^I$ of the object spatial scales, the set $\mathcal{R}$ of the object roles, and the set $\hat{\Theta}_{\mathcal{F}}$ of all local frames.

B. Principal Constraints Estimation

Given the preprocessed data, our goal is to jointly extract a set $\mathcal{P}$ of keypoints and a set $\mathcal{C} = \{C_l\}_{l=1}^L$ of geometric constraints. As shown in Fig. 4, we consider five basic types of geometric constraints for keypoints in a 3-D Cartesian space, namely point-to-point (p2p), point-to-line (p2l), point-to-plane (p2P), point-to-curve (p2c), and point-to-surface (p2S). The p2p, p2l, and p2P constraints are linear and can, therefore, be estimated by analyzing the variance of the keypoint positions in multiple demonstrations using *PCA* [38]. In contrast, p2c and p2S are nonlinear geometric constraints, which we estimate with the iterative *PME* (see Section III-B). Note that more complex constraints, such as colinear, coplanar, parallel, and perpendicular, result from combinations of our five basic types of constraints. To ensure that the constraints are reliably estimated, the criterion of the proposed PCE is adapted to the number of demonstrations. Specifically, the single-demonstration case (i.e., one-shot IL) is considered a special case as it does not provide sufficient information to learn generalizable skills. Therefore, we propose heuristically designed distance-based criteria (see Section IV-B1). In contrast, when several demonstrations are available (i.e., few-shot IL), the keypoints and geometric constraints are learned based on the variability of the demonstrations (see Sections IV-B2 and IV-B3). Moreover, nonlinear constraints are considered only if enough demonstrations are available.

1) Distance Criteria for a Single Demonstration: A single demonstration ($N = 1$) does not provide examples of variations in the demonstrated task, and thus prevents the learning of generalizable skills. Therefore, in this case, we learn a set of constraints that fully determines the pose of the objects. To do so, we assume that the objects are rigid and extract three keypoints for each slave object in order to fully constrain their position in a 3-D space. Note that K-VIL can also be applied to 2-D cases, where two keypoints are sufficient to determine the pose of an object. We map the trajectories of all candidates on the slave objects into each of the canonical local frames $\mathcal{F}_j(t)$ on the master object at time step t, where $j \in \mathcal{P}_m$. Therefore, all demonstrations obtained from arbitrary viewpoints are aligned in a common viewpoint defined by the local frame $\mathcal{F}_j(t)$ (see Section VI-B2). Then, $\tilde{\boldsymbol{\tau}}_j^k(t) \in \mathbb{R}^{N \times D}$ with $k \in \mathcal{P}_s$ represents the positions of the kth candidate point in all demonstrations viewed from the jth common viewpoint at time step t, and we use this variable to denote the *candidate positions* in the rest of the article. We observed that, for a variety of daily manipulation tasks, the closest point k_1 on the slave object to the master object is often crucial to respect contact or avoid a collision, whereas the furthest point k_2, in combination with k_1, controls the pose of the object. Motivated by these heuristics, we propose the following procedure for each slave object. First, we choose the local frame $\mathcal{F}^*(t)$ from the canonical local frames as the closest on average to all candidates on the slave objects. The two keypoints k_1 and k_2 on the slave object then correspond to the closest and farthest candidates from the selected local frame $\mathcal{F}^*(t)$. For a 3-D task space, we select an additional keypoint k_3 as the farthest candidate from both k_1 and k_2. To fully determine the pose of the slave object, the three keypoints are subject to linear p2p constraints. For each slave object, we finally obtain a set $\mathcal{P}_d = \{k_l\}_{l=1}^D$ of keypoints and

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

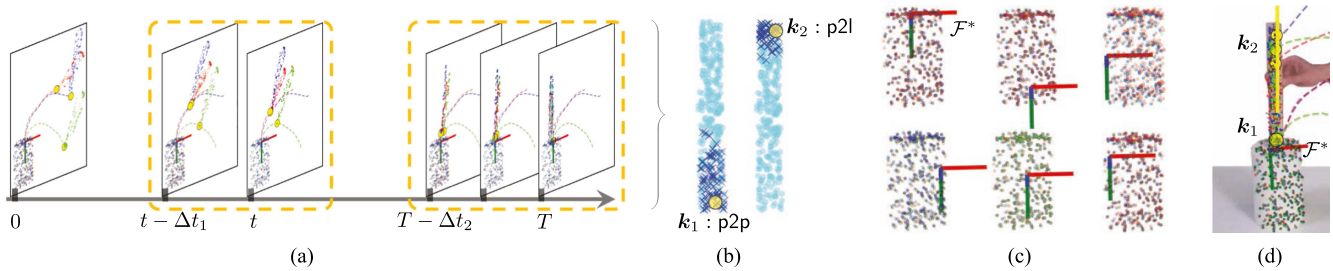


Fig. 5. HAC for inserting a stick into a paper roll (see also Section VI for the task description). (a) Selected candidates are first clustered in time (– –) to identify the adjacent time steps. (b) Candidates (×) in each time cluster (e.g., here in the last time cluster $[T-\Delta t_2, T]$) are then clustered based on their positions in the canonical shape (●) of the stick for each constraint (here p2p and p2l). For each position cluster, the keypoint (●) with the lowest variability is finally selected. (c) Since the paper roll has no shape variation, i.e., all canonical local frames are equivalent, the closest frame $\mathcal{F}^*$ to the selected keypoints is selected. (d) Final task representation (see also Fig. 13). (a) Time clustering. (b) Position clustering. (c) Equivalent local frames $\mathcal{F}$. (d) Insertion.

the corresponding geometric constraints $\mathcal{C} = \{C_l\}_{l=1}^{D}$, where $C_l = \{\mathcal{M}_{\text{point}}(k_l), t, \theta_{\mathcal{F}^*(t)}, \text{p2p}\}$ defines a p2p constraint on a 0-D principal manifold $\mathcal{M}_{\text{point}}$ on point k_l at time t in the local frame given by $\vartheta_{\mathcal{F}^*}$.

2) Variance Criteria for Linear Constraints: When several demonstrations ($N > 1$) are available, we leverage their variability to estimate linear constraints beyond p2p. To do so, we obtain the candidate positions $\tilde{\boldsymbol{\tau}}_j^k(t)$ in the canonical local frames $\mathcal{F}_j(t)$ at time t, as described in Section IV-B1, and compute the explained variance $\boldsymbol{\nu}_j^k(t) = \mathbb{V}_{\text{PCA}}[\tilde{\boldsymbol{\tau}}_j^k(t)] \in \mathbb{R}^D$ of each candidate using PCA. The *spatial variability* $\boldsymbol{\eta}_j^k(t)$ is then defined as $\boldsymbol{\eta}_j^k(t) = (\boldsymbol{\nu}_j^k(t))^{1/2}/\tilde{\varphi}_i$, with $\tilde{\varphi}_i$ the spatial scale of the slave object O_i to which the kth candidate belongs. In contrast to the explained variance, spatial variability removes dependencies on the object size. This allows us to empirically define two object-agnostic lower and upper thresholds ξ_1 and ξ_2 to identify appropriate linear geometric constraints based on the computed spatial variability. Namely, the constraints of each candidate k are determined by the following three conditions.

1) $\eta_{j,1}^k(t) < \xi_1$ implies a low spatial variability in the first components as well as across all other dimensions since all the components of the spatial variability are ranked in decreasing order, i.e., $\eta_{j,e}^k > \eta_{j,e+1}^k$, where $e = \{1, 2\}$. This also means that the position of the candidate k remains close to a fixed point $\mathcal{M}_{\text{point}}$ across all demonstrations. Therefore, k is subject to a p2p constraint.
2) $\eta_{j,2}^k(t) < \xi_1$ and $\eta_{j,1}^k(t) > \xi_2$ imply that k is constrained on a line $\mathcal{M}_{\text{line}}$ along the first component, i.e., k is subject to p2l constraint.
3) Similarly, $\eta_{j,3}^k(t) < \xi_1$ and $\eta_{j,2}^k(t) > \xi_2$ indicate that k is constrained on a plane $\mathcal{M}_{\text{plane}}$ going through the first two components, i.e., k is subject to a p2P constraint.

Any spatial variability $\boldsymbol{\eta}_j^k(t)$ satisfying the above conditions indicates the joint selection of the kth candidate, the jth local frame, and tth time step. All candidates selected as such form a set $\mathcal{P}_l$ of keypoints subject to linear geometric constraints. Note that, due to the fact that two distinct points define a line and three noncollinear points define a plane, we learn p2l constraints when $N > 2$ and of p2P constraints when $N > 3$.

3) Variance Criteria for Nonlinear Constraints: The linear constraints may not suffice to represent a given task accurately despite them being easily estimated from a few demonstrations. For instance, the pouring task of Fig. 1(c) requires a point-to-curve constraint. Therefore, we additionally estimate nonlinear (p2c and p2S) constraints with PME. In the following, a set $\mathcal{P}_s$ of all candidate points on slave objects that do not satisfy any linear constraints is considered as potential candidates for nonlinear constraints. In our case, we replace the random D-dimensional vector $\boldsymbol{x}$ in (1) with the candidate point $\tilde{\boldsymbol{\tau}}_j^k(t)$ on the demonstrated trajectory $\tilde{\boldsymbol{\tau}}_j^k$ at time step t so that the PME loss in Section III-B becomes

$$\mathcal{L}(f, \pi_d) = \mathbb{E}\left\|\tilde{\boldsymbol{\tau}}_j^k(t) - f(\pi_d(\tilde{\boldsymbol{\tau}}_j^k(t)))\right\|^2 + \lambda\|\kappa_f\|^2, \quad k \in \mathcal{P}_s.$$

After obtaining π_d from PME, we compute the projections of candidates onto the manifold, i.e., $\hat{\boldsymbol{\tau}}_j^k(t) = \pi_d(\tilde{\boldsymbol{\tau}}_j^k(t))$, where $\hat{\boldsymbol{\tau}}_j^k(t) \in \mathbb{R}^{N\times d}$. Then, analogously to Section IV-B2, we define the explained variance $\nu_{j,\parallel}^k(t)$ in the tangential direction of the principal manifold as the variance of the projections, i.e., $\nu_{j,\parallel}^k(t) = \mathbb{V}[\|\hat{\boldsymbol{\tau}}_j^k(t)\|] \in \mathbb{R}$. The explained variance $\nu_{j,\perp}^k(t)$ in the orthogonal direction corresponds to the variance of the length of the stress vectors $\boldsymbol{s} = \tilde{\boldsymbol{\tau}}_j^k(t) - f(\hat{\boldsymbol{\tau}}_j^k(t))$, i.e., $\nu_{j,\perp}^k(t) = \mathbb{V}[\|\boldsymbol{s}\|]$. Similar to the linear case, the spatial variability is defined as $\eta_{j,z}^k(t) = \sqrt{\nu_{j,z}^k(t)}/\tilde{\varphi}_i, z \in \{\perp, \|\}$. The set $\mathcal{P}_{nl}$ of keypoints subject to nonlinear geometric constraints is then selected as follows:

$$\mathcal{P}_{nl} = \left\{k \mid \nu_{j,\perp}^k(t) < \xi_1, \nu_{j,\|}^k(t) > \xi_2, k \in \mathcal{P}_s\right\}.$$

The type of the geometric constraints is determined by the intrinsic dimension d of the learned principal manifold, i.e., $d = 1$ and $d = 2$ indicate a p2c and a p2S constraint, respectively. Notice that, in order to guarantee their reliable estimation, nonlinear constraints are considered only when enough demonstrations ($N > 10$) are available.

4) HAC: As explained in Section IV-B2, each selected candidate point k in the resulting sets of linear and nonlinear constraints $\mathcal{P}_l$ and $\mathcal{P}_{nl}$ corresponds to jointly selected time step t and local frame j. Redundancy may occur due to adjacent time steps, neighboring keypoints, or equivalent local frames. To resolve this redundancy, we first cluster the keypoints in time using HAC to identify adjacent time steps. Fig. 5 shows an example of HAC for an insertion task. We then use HAC again to cluster the keypoints within each time cluster based on their positions in the canonical shape of the slave object, thus identifying neighboring

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

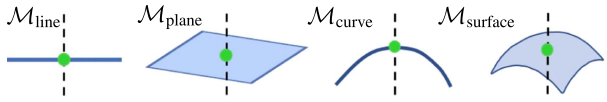


Fig. 6. Orthogonal direction (– –) to the principal manifolds.

keypoints. The redundancy is finally resolved by keeping only the keypoint with the lowest variability to represent each position cluster. This keypoint is selected for its robustness against sensor and correspondence detection noise. If a selected keypoint at a selected time step is subject to multiple constraints represented in different local frames, we select the closest local frame to the keypoint on average. In summary, the proposed PCE retrieves a sparse set of L keypoints as the union $\mathcal{P} = \mathcal{P}_d \cup \mathcal{P}_l \cup \mathcal{P}_{nl}$ and their associated (non)linear constraints $\mathcal{C} = \{C_l\}_{l=1}^{L}$, which are exploited to represent the task as explained next.

C. Extraction of K-VIL's Complete Task Representation

While the keypoints and associated constraints estimated in Section IV-B allow us to understand the demonstrated task, a control policy is additionally required for reproducing the task. Here, we propose to model the observed keypoints trajectories as VMPs [13]. For our purposes, we train the VMPs from an object-centric perspective and according to the constraints estimated via PCE. Specifically, for each keypoint subject to a p2p constraint, a VMP is trained on its observed trajectory $\tilde{\boldsymbol{\tau}}_l$ retrieved in the corresponding local frame j from time step 1 to the extracted time step t, i.e., $\tilde{\boldsymbol{\tau}}_l = (\boldsymbol{\tau}_j^k(1), \ldots, \boldsymbol{\tau}_j^k(t))^{\mathsf{T}}$, where k and l indicate that the kth candidate point in the dense set $\mathcal{P}_s$ is selected as the lth keypoint in the sparse set $\mathcal{P}$. Note that, for the case of intrinsic dimension $d > 0$ (i.e., p2l, p2P, p2c, and p2S constraints), the constraint is fulfilled if and only if the corresponding keypoint is placed at the time step t on the principal manifold that defines the constraint. Although the location of the keypoint on the manifold does not affect the fulfillment of the constraints, it may influence the similarity between the demonstrated object poses and those obtained in the reproduction. Therefore, we propose to decompose the control of such keypoint by considering the orthogonal and the tangential direction with respect to the corresponding principal manifold independently (see Fig. 6). The keypoint motion along the orthogonal direction represents the demonstrated style of approaching the principal manifold and guarantees the fulfillment of the constraints, while the motion along the tangential direction realizes the extrapolation of the keypoint target position and controls the similarity of the object pose between the demonstrations and the reproduction. Due to potential large shape variations in the objects used when reproducing the task, the final keypoints' target positions on the principal manifold may not align with the demonstrated targets. Therefore, we only train the VMP on the keypoint trajectories projected onto the orthogonal direction, i.e., $\tilde{\boldsymbol{\tau}}_{l,\perp}$. An example of the projected and reproduced trajectories obtained using the learned VMP in the case of a p2c constraint is shown in Fig. 7. At each time step during reproduction, we uncover the 3-D target position of a keypoint by adding an offset generated by the VMP in the orthogonal direction to the orthogonal projection

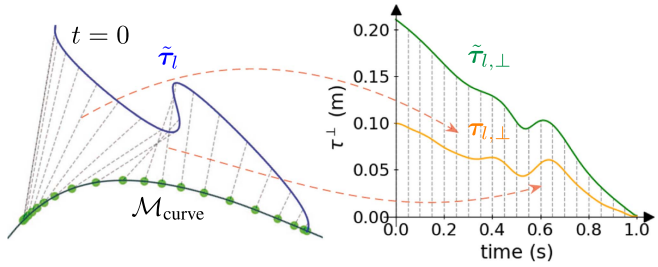


Fig. 7. Projected and reproduced trajectories using a VMP for a p2c constraint. The demonstrated trajectory $\tilde{\boldsymbol{\tau}}_l$ (—) is projected at each time step in the orthogonal direction (– –) of the principal manifold $\mathcal{M}_{\text{curve}}$. The projected trajectory $\tilde{\boldsymbol{\tau}}_{l,\perp}$ (—) is used to train movement primitives, which is then used to reproduce trajectories, e. g. , $\boldsymbol{\tau}_{l,\perp}$ (—) with a new start position at 0.1 m and goal position at 0 m. The arrows (-→) mark the corresponding projected trajectory between the 3-D and 2-D plots at two time steps.

of the current keypoint onto the principal manifold (see $\boldsymbol{k}_2^*$ in Figs. 8 and 9). Therefore, by setting the VMP goal to 0, the keypoints fulfill the corresponding geometric constraints at the end of their trajectory. The motion of the keypoints along the orthogonal and tangential directions is controlled via the KAC presented in Section V. In summary, K-VIL's final task representation is composed of a set of keypoints represented by their descriptors $\mathcal{D} = \{\boldsymbol{d}_l\}_{l=1}^{L}$, their associated geometric constraints $\mathcal{C} = \{C_l\}_{l=1}^{L}$, and their associated movement primitives encoded via the set of weights $\Omega = \{\boldsymbol{w}_l\}_{l=1}^{L}$.

V. Keypoint-Based Admittance Controller

After learning the representation of a given task from demonstrations, we aim at reproducing this task with a robot. This means that the robot should be able to interact with the objects such that their keypoints follow the learned constrained trajectories. This requires filling the gap between K-VIL's task representation and real-time robot controllers. To this end, we propose a KAC, which does the following:

1) handles variable numbers of extracted keypoints for different tasks;
2) enables the extrapolation of keypoint target positions on their learned principal manifolds;
3) resolves potential interference between different types of geometric constraints.

Note that 2) and 3) are required to handle large object shape variations in the task reproduction.

Specifically, a KAC associates each keypoint in $\mathcal{P}$ with a virtual spring–damper system, whose attractor is computed via the corresponding VMPs (see Section V-A). As detailed in Section V-D, the sum of the attraction forces of the spring–damper systems of all keypoints is then used as the task-space force command for the robot. This allows the KAC to handle a varying number of keypoints for different tasks. Regarding 2), the extrapolation of keypoint target positions subject to a p2p constraint is not allowed. For non-p2p constraints, this is achieved by decomposing the control in orthogonal and tangential directions of the learned principal manifolds (see Section IV-C). As a result, the keypoints approach the principal manifolds using the motion profiles learned from the projected

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

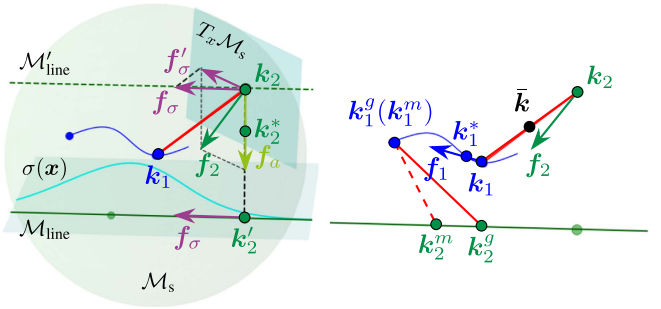


Fig. 8. Illustration of the attraction and density forces when the keypoints k_1 and k_2 are subject to p2p and p2l constraints, respectively. *Left:* The approach force f_a of k_2 is computed by the virtual spring–damper system between the attractors k_2^* and k_2. The density force f_σ is then projected onto the tangent space of the sphere at k_2. The control force of k_2 is the combination of the attraction and the projected density force. *Right:* k_1 is controlled by the attraction force f_1 following the attractor k_1^* and the VMP (—) to reach the target k_1^g, which coincides with the demonstrated target k_1^m. Note that the target k_2^g of k_2 does not coincide with k_2^m due to object shape variation, i.e., the distance (—) between k_1 and k_2 during reproduction is longer than for the demonstration (- -).

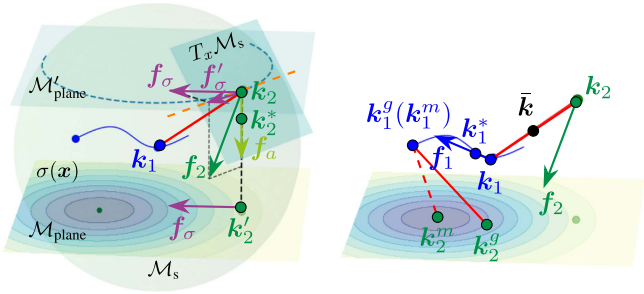


Fig. 9. Illustration of the attraction and density forces when k_2 is subject to a p2P constraint. In contrast to Fig. 8, the density force f_σ is projected onto the intersection line (- -) of the shifted principal manifold $\mathcal{M}'_{\text{plane}}$ and the tangent space. Legend as in Fig. 8.

trajectories in orthogonal directions. The control force generated by the virtual spring–damper system of each keypoint remains orthogonal to the principal manifold at each time step, and the keypoints reach the corresponding geometric constraints when the execution of the VMP finishes. While this leads to the extrapolation of keypoint target positions on the principal manifold, it does not account for the distance between the demonstrated and extrapolated targets. Therefore, we propose to balance extrapolation and regulation by estimating the density function of the demonstrated targets on the principal manifolds, as described in Section V-B. This density is then used to compute an additional force, i.e., the density force, that drives each keypoint toward the demonstrated targets. Finally, we address the interference issue 3) by assigning different priorities to different types of geometric constraints in Section V-C. The different steps of KAC are detailed next.

A. Attraction Force

Given K-VIL's task representation and a new image frame $\boldsymbol{A}$ for the task reproduction, we can identify the keypoints representing the task. Namely, their positions $\boldsymbol{k}_l \in \mathbb{R}^D$, with $l \in [1, L]$, represented in the root frame $\mathcal{F}_r$ of the robot are obtained using the visual descriptors $\boldsymbol{d}_l$ and the DON-based correspondence function $f_c(\boldsymbol{A}, \boldsymbol{d}_l)$ (see Section III-A). The attractor $\boldsymbol{k}_l^*$ of the virtual spring–damper system at each time step is computed for each keypoint by the corresponding VMPs projected onto $\mathcal{F}_r$. The attraction force generated by the virtual spring–damper system is then computed as

$$\boldsymbol{f}_{a,l} = \bar{\boldsymbol{K}}_p(\boldsymbol{k}_l^* - \boldsymbol{k}_l) + \bar{\boldsymbol{K}}_d(\dot{\boldsymbol{k}}_l^* - \dot{\boldsymbol{k}}_l)$$

where $\bar{\boldsymbol{K}}_p$ and $\bar{\boldsymbol{K}}_d$ are the diagonal stiffness and damping matrices, respectively, and $\dot{\boldsymbol{k}}_k$ and $\dot{\boldsymbol{k}}_l^*$ are the velocities of $\boldsymbol{k}_l$ and $\boldsymbol{k}_l^*$, respectively. As explained in Section IV-C, in the case of non-p2p constraints, the VMPs are trained on trajectories projected in directions that are orthogonal to the principal manifold. Therefore, the learned VMPs and the attraction forces enable the reproduction of the demonstrated motion patterns in the orthogonal direction. This implies that the final positions of the keypoints can be extrapolated anywhere on the principal manifolds, e.g., to satisfy object shape variations and other geometric constraints. However, without considering the demonstrated targets on the principal manifold, we may lose important information about successful task execution or a specific style of execution. We obtain such information using kernel density estimation and provide additional density forces driving the keypoints toward the demonstrated targets.

B. Density Force

Given a non-p2p constraint, we project the demonstrated keypoint positions $\tilde{\boldsymbol{\tau}}_l(t)$ at the extracted time step t onto the corresponding d-dimensional principal manifold using the learned projection index π_d, i.e., $\hat{\boldsymbol{k}}_l^m = \pi_d(\tilde{\boldsymbol{\tau}}_l(t)) \in \mathbb{R}^{N\times d}$. Since at time step t, the lth keypoint is supposed to fulfill the geometric constraints, and we interpret $\hat{\boldsymbol{k}}_l^m$ as its demonstrated target positions on the manifold. We then estimate the density function $\sigma(\boldsymbol{x})$ of the keypoint target positions from $\hat{\boldsymbol{k}}_l^m$ using kernel density estimation [39] with SE kernels. Examples of estimated density functions for p2l, p2P, p2c, and p2S constraints are depicted in Fig. 4(b)–(e). This density function indicates the probability of a keypoint target position on the corresponding principal manifold, given the demonstrated target positions. In other words, the density function indicates the confidence level of K-VIL when extrapolating the keypoint target positions to new locations on the principal manifold, which may occur due to object shape variations in the reproduction. Examples of extrapolated keypoint target positions (×) during reproduction are depicted in Fig. 4(b)–(e). Notice that the target position in Fig. 4(b)-*left* has a lower probability (i.e., a lower extrapolation confidence) than the one in Fig. 4(b)-*right* due to its increased distance with the demonstrated target positions. This illustrates that the control of such keypoints must not only fulfill the geometric constraints but also be as close as possible to the demonstrated targets on the constraints. Therefore, in addition to the attraction force $\boldsymbol{f}_{a,l}$ that guarantees the fulfillment of the geometric constraint, we define a *density force* $\boldsymbol{f}_{\sigma,l}$ to drive the keypoints into regions with higher probability. To do so, we first

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

define the driving force $\boldsymbol{f}_{\sigma,1}$ computed from the density field $\nabla\sigma(\boldsymbol{x})$ as

$$\boldsymbol{f}_{\sigma,1} = g_1 \cdot f(\nabla\sigma(x)). \tag{2}$$

For the regions where $\boldsymbol{f}_{\sigma,1}$ is too small to drive the keypoints, we then define a minimal driving force $\boldsymbol{f}_{\sigma,2}$ as follows:

$$\boldsymbol{f}_{\sigma,2} = g_2 \cdot f\left(\frac{\boldsymbol{k}^m - \boldsymbol{k}'_l}{\|\boldsymbol{k}^m - \boldsymbol{k}'_l\|_2}\right) \tag{3}$$

where $\boldsymbol{k}'_l = \pi_d(\boldsymbol{k}_l) \in \mathbb{R}^d$ is the projection of the keypoint onto the principal manifold, $f(\cdot)$ is the reconstruction function (see Section IV-B3), and g_1 and g_2 are the force scaling parameters. Note that $\boldsymbol{f}_{\sigma,2}$ points directly to the mean of the demonstrated targets $\hat{\boldsymbol{k}}_l^m$ on the principal manifold, i.e., $\boldsymbol{k}^m = \text{avg}(\hat{\boldsymbol{k}}_l^m)$. Then, the density force is the maximum of $\boldsymbol{f}_{\sigma,1}$ and $\boldsymbol{f}_{\sigma,2}$, i.e.,

$$\boldsymbol{f}_{\sigma,l} = \arg\max_{\boldsymbol{f}} \|\boldsymbol{f}\|_2, \quad \boldsymbol{f} \in \{\boldsymbol{f}_{\sigma,1}, \boldsymbol{f}_{\sigma,2}\}. \tag{4}$$

Examples of such density forces $\boldsymbol{f}_\sigma$ in the case of p2l and p2P constraints are shown in Figs. 8 and 9.

C. Priority

In the case of large object shape variations, controlling a p2l constraint with the same priority as a p2p constraint may lead to a violation of the latter. To reduce such interference, we propose to set a higher priority to p2p constraints compared with the other constraint types. For the sake of clarity, we use Figs. 8 and 9 to explain this concept, where Fig. 8 shows the case of two constraints, p2p for $\boldsymbol{k}_1$ and p2l for $\boldsymbol{k}_2$, while $\boldsymbol{k}_2$ is subject to a p2P constraint in Fig. 9. In both cases, we construct a sphere centered at $\boldsymbol{k}_1$ with radius $\|\boldsymbol{k}_2 - \boldsymbol{k}_1\|$ and define the tangent space of the sphere at $\boldsymbol{k}_2$ as the plane formed by all the lines tangent to the sphere at $\boldsymbol{k}_2$. For clarity, Figs. 8 and 9 also depict the corresponding principal manifolds $\mathcal{M}_{\text{line}}$ and $\mathcal{M}_{\text{plane}}$ shifted in parallel to go through $\boldsymbol{k}_2$ as $\mathcal{M}'_{\text{line}}$ and $\mathcal{M}'_{\text{plane}}$. Assuming solid connections (—) between $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$, large density forces $\boldsymbol{f}_\sigma$ generated for $\boldsymbol{k}_2$ will also drag $\boldsymbol{k}_1$ along the same direction. This may lead to the violation of the p2p constraint of $\boldsymbol{k}_1$ and cause collision if $\boldsymbol{k}_1$ is close to the master object. To reduce such interference when the principal manifold is a line $\mathcal{M}_{\text{line}}$, we project $\boldsymbol{f}_\sigma$ onto the tangent space $T_x\mathcal{M}_{\text{s}}$ of the sphere $\mathcal{M}_{\text{s}}$ so that the motion of $\boldsymbol{k}_1$ remains unaffected by the projected density force $\boldsymbol{f}'_\sigma$ (see Fig. 8). Similarly, when a principal manifold is a plane (see Fig. 9), we project $\boldsymbol{f}_\sigma$ onto the intersection between the tangent space and the shifted principal plane $\mathcal{M}'_{\text{plane}}$. This also holds for the nonlinear constraints (p2c and p2S) for which a linear approximation is considered at each time step.

In summary, on the one hand, the density force allows the reproduced task to be similar to the demonstrations on the principal manifold. On the other hand, the priority mechanism reduces the interference of the density force with p2p constraints while maintaining the extrapolation capability of K-VIL. Overall, the decomposition of the control force into attraction force (see Section V-A) and projected density force (see Sections V-B and V-C) is key to balancing the similarity of the reproduced task to the demonstration and the extrapolation capability. In practice, the stiffness and damping gains of the virtual spring–damper systems are empirically tuned for good tracking accuracy and control stability. Notice that one-shot IL is considered as a special case (see Section IV-B1). This is due to the fact that the learned task representation is composed of three keypoints subject to p2p constraints. In this case, no density force is needed, and the constraint priorities are defined as $\text{Pri}_1 > \text{Pri}_2 > \text{Pri}_3$ since $\boldsymbol{k}_1$ usually represents the contact point of two objects. Therefore, to ensure a higher control precision of $\boldsymbol{k}_1$, we set the stiffness gains of the three keypoints to $\bar{\boldsymbol{K}}_{p,1} = 5\bar{\boldsymbol{K}}_{p,2} = 10\bar{\boldsymbol{K}}_{p,3}$ and the respective damping gains to $\bar{\boldsymbol{K}}_{d,l} = 2\bar{\boldsymbol{K}}_{p,l}^{1/2}$, where $l \in [1, 3]$, which ensure a critically damped behavior for control stability.

D. Admittance Control

The goal of the KAC is to compute the control command of the robot arm from the attraction forces $\boldsymbol{f}_a$ and the projected density forces $\boldsymbol{f}'_\sigma$ of all keypoints. To do so, we first compute the control force of each keypoint as $\boldsymbol{f}_l = \boldsymbol{f}_{a,l} + \boldsymbol{f}'_{\sigma,l}$ and define a virtual tool-center-point (TCP) $\bar{\boldsymbol{k}} = \sum_{l=1}^{L} \boldsymbol{k}_l / L$ as the mean of all keypoint positions (see Fig. 8-*right* and Fig. 9-*right*). The virtual TCP is driven by a virtual force and torque

$$\boldsymbol{f}_f = \sum_{l=1}^{L} \boldsymbol{f}_l \quad \text{and} \quad \boldsymbol{f}_\tau = \sum_{l=1}^{L} (\boldsymbol{k}_l - \bar{\boldsymbol{k}}) \times \boldsymbol{f}_l$$

with $\times$ denoting the vector cross product. The total control force $\boldsymbol{f}_v = [\boldsymbol{f}_f^\mathsf{T}, \boldsymbol{f}_\tau^\mathsf{T}]^\mathsf{T}$ is applied to the robot end-effector (i.e., the humanoid hand) to calculate the virtual acceleration as follows:

$$\ddot{\boldsymbol{x}}_v = \tilde{\boldsymbol{K}}_p(\boldsymbol{x}_0 - \boldsymbol{x}_v) - \tilde{\boldsymbol{K}}_d\dot{\boldsymbol{x}}_v - \tilde{\boldsymbol{K}}_m\boldsymbol{f}_v$$

where $\boldsymbol{x}_0$ and $\boldsymbol{x}_v$ are the initial and virtual poses of the robot end-effector, $\dot{\boldsymbol{x}}_v$ is its virtual velocity, and $\tilde{\boldsymbol{K}}_m$, $\tilde{\boldsymbol{K}}_d$, and $\tilde{\boldsymbol{K}}_p$ are the inertia, damping, and stiffness factors, respectively. The robot is controlled using a task-space inverse dynamics controller, whose task-space control force $\boldsymbol{f}_m$ is calculated as follows:

$$\boldsymbol{f}_m = \boldsymbol{K}_p(\boldsymbol{x}_v - \boldsymbol{x}) + \boldsymbol{K}_d(\dot{\boldsymbol{x}}_v - \dot{\boldsymbol{x}}) + \boldsymbol{h}_c$$

where $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ are the current end-effector pose and velocity, $\boldsymbol{K}_d$ and $\boldsymbol{K}_p$ are the damping and stiffness factors of the impedance controller, respectively, and $\boldsymbol{h}_c$ represents the Coriolis and gravitational force in the task space.

VI. Evaluation

We evaluate our approach in five daily tasks involving different types of geometric constraints and various categorical objects (see Fig. 10). Namely, the considered tasks are press button (PB, Fig. 11), fetch tissue (FT, Fig. 12), insert sticks into a paper roll (IS, Fig. 13), pour water (PW, Fig. 14), hang hat on a rack (HH, Fig. 15), and clean table with a dustpan and a brush (CT, Fig. 16). As a prerequisite for our experiments, we train DON in a self-supervised and task-agnostic manner. The training dataset was collected with a handheld Azure Kinect camera moving around objects, such as tissue boxes, teacups, a rack, a hat, kettles, a paper roll, sticks, dustpans, and brushes (see Fig. 10). The collected data were then postprocessed via a 3-D

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Fig. 10. Objects used in our article include (a) tissue boxes, (b) teacups, (c) rack and a hat, (d) kettles, (e) paper roll, (f) sticks, and (g) dustpans and brushes. Note that the rack can be assembled with sticks #6–10 to have multiple shape variations.

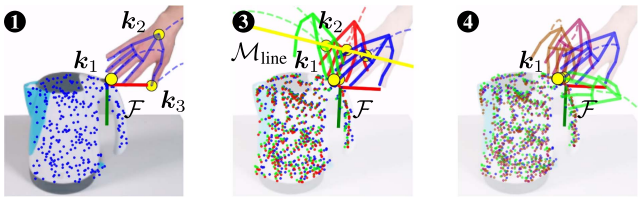


Fig. 11. Press a button on the kettle to open the lid with variations in hand orientation. The candidate points (colored points) and the skeleton of hands (colored line segments) are overlain on the objects at time step T. Different colors denote different trials. The keypoints $\boldsymbol{k}_l$ (○) are extracted from different number ⓝ of demonstrations in each subfigure. The demonstrated trajectories (– –), the local frames $\mathcal{F}$, and the estimated principal manifolds (—) are also depicted. Notice that the demonstrations were provided from different viewpoints, although they are here represented aligned to the local frame $\mathcal{F}$ for a better illustration of the extracted keypoints and constraints (for viewpoint mismatch, see Fig. 16.). (a) p2p: $\boldsymbol{k}_1, \boldsymbol{k}_2, \boldsymbol{k}_3$. (b) p2p: $\boldsymbol{k}_1$ and p2l: $\boldsymbol{k}_2$. (c) p2p: $\boldsymbol{k}_1$.

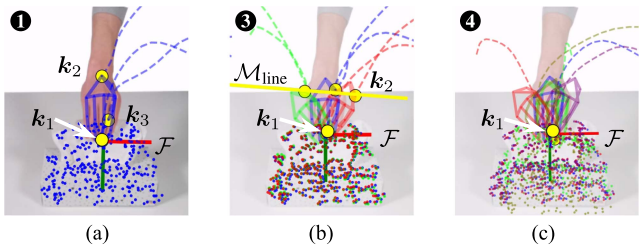


Fig. 12. Fetch tissue with variations in hand orientation. Legend as in Fig. 11. (a) p2p: $\boldsymbol{k}_1, \boldsymbol{k}_2, \boldsymbol{k}_3$. (b) p2p: $\boldsymbol{k}_1$ and p2l: $\boldsymbol{k}_2$. (c) p2p: $\boldsymbol{k}_1$.

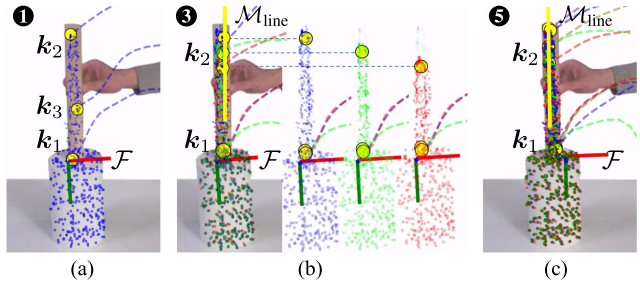


Fig. 13. Approach the insertion position to insert sticks with three length variations into a paper roll. Legend as Fig. 11. (a) p2p: $\boldsymbol{k}_1, \boldsymbol{k}_2, \boldsymbol{k}_3$. (b) p2p: $\boldsymbol{k}_1$ and p2l: $\boldsymbol{k}_2$. (c) p2p: $\boldsymbol{k}_1$ and p2l: $\boldsymbol{k}_2$.

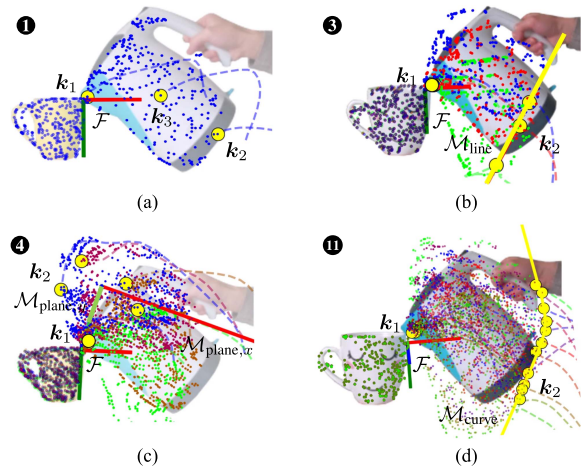


Fig. 14. Pouring task with variations in the shape of cups and the orientation of the kettle. The principal plane in (b) is represented by orthogonal vectors $\mathcal{M}_{\text{plane},x}$ (—) and $\mathcal{M}_{\text{plane},y}$ (—). Other legends as in Fig. 11. (a) p2p: $\boldsymbol{k}_1, \boldsymbol{k}_2, \boldsymbol{k}_3$. (b) p2p: $\boldsymbol{k}_1$ and p2l: $\boldsymbol{k}_2$. (c) p2p: $\boldsymbol{k}_1$ and p2P: $\boldsymbol{k}_2$. (d) p2p: $\boldsymbol{k}_1$ and p2c: $\boldsymbol{k}_2$.

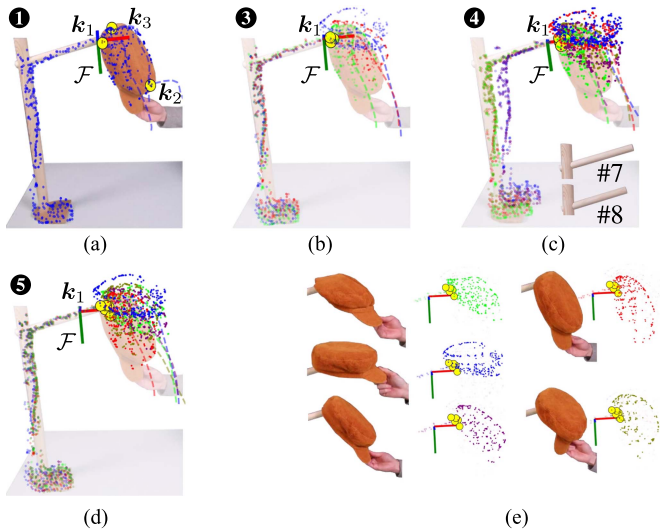


Fig. 15. Hang a hat on a rack. Note that there are only slightly deformations of the hat in (a)–(c) and relatively more obvious deformations in (d) [see (e)], while pose variations in the hats are considered in all cases. The racks in (c) have two shape variations. Legend as Fig. 11. (a) p2p: $\boldsymbol{k}_1, \boldsymbol{k}_2, \boldsymbol{k}_3$. (b) p2p: $\boldsymbol{k}_1$. (c) p2p: $\boldsymbol{k}_1$. (d) p2p: $\boldsymbol{k}_1$. (e) Shape variations of the hat.

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

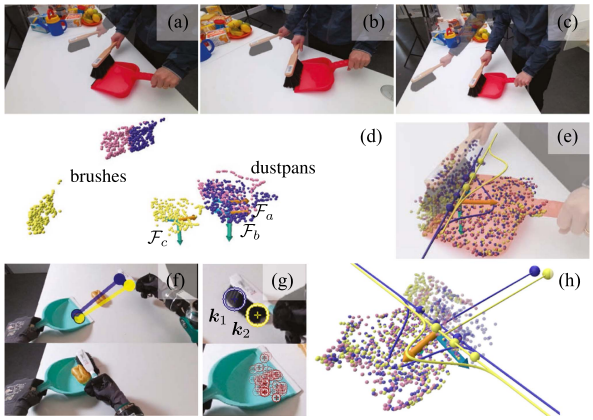


Fig. 16. K-VIL handles viewpoint mismatch in the three demonstrations (a)–(c) by aligning the corresponding local frames on the master object dustpan in (d), which results in an aligned viewpoint in (e). Two p2l constraints and their probability density functions on the principal lines are visualized. The robot reproduces the CT ❸ task from a new viewpoint with a novel brush and dustpan (f), with the keypoints (● $\boldsymbol{k}_1$, ● $\boldsymbol{k}_2$) detected on the brush hair (g). The local frame on the dustpan is determined by the $Q = 50$ neighboring points as shown in (g). (f) and (h) depict the keypoints and their movement primitives in 2-D and 3-D, respectively.

reconstruction process using Open3D [40]. We used MediaPipe to detect 21 keypoints on the human hands and treat the hands as a special type of object. It is important to emphasize that K-VIL is not limited to DON and MediaPipe, but instead can be used with any correspondence detection model, e.g., NDFs. In all experiments, we sample $P_i = 300$ (for hands $P_i = 21$) candidate points on each slave object and use $Q = 50$ (for hands $Q = 10$) neighboring candidates on the master object as references for local frame detection. We use 20 kernels for the VMPs. The empirical thresholds ξ_1 and ξ_2, and the controller gains are fine-tuned for each task. A full list of control parameters is included in the example code.

We first evaluate the ability of K-VIL to extract generalizable task representations, given different number of demonstrations (see Section VI-B1 and Section VI-B3 for a one-shot and a few-shot VIL setup, respectively, as well as Figs. 11–16). We demonstrate how variations in objects' pose and shape contribute to the efficient extraction of generalizable task representations and evaluate the ability of K-VIL to reproduce the corresponding tasks learned from a different number of demonstrations. We discuss the problems that arise when only scarce demonstrations are provided in Section VI-B4. We then show how they are resolved by providing more demonstrations and summarize the number of demonstrations required to learn a generalizable representation for each task. The proposed KAC is finally evaluated in terms of the control accuracy, precision, and success rate in Section VI-C. For more visualizations of the evaluation results on one/few-shot IL, reproduction of skills learned from a different number of demonstrations, and other types of geometric constraints, we refer the interested reader to the accompanying videos and to the https://sites.google.com/view/k-vil paper website.

A. Evaluation Protocols

For each task, we record a few demonstration videos (RGB-D) of a human performing the task using an Azure Kinect mounted on the head of the humanoid robot ARMAR-6 [41]. For tasks involving categorical objects, we distinguish between the object instances used for training of the vision models (the DON and Mask R-CNN models) for the demonstrations and for the reproductions. If we only have one instance of a specific object category, this instance is used for the training, demonstrations, and reproductions. We define a set of *extraction tasks* $\mathcal{T}_E = \{\mathsf{PB}, \mathsf{FT}, \mathsf{PW}, \mathsf{HH}, \mathsf{IS}, \mathsf{CT}\}$ for which we evaluate K-VIL's ability to extract generalizable task representations, given $N \in \{1, 3, 4, 5, 11\}$ demonstrations, respectively. For clarity, we only evaluate the representations of the last time cluster, i.e., the goal configuration of each task when $t = T$. We then define a set of *reproduction tasks* $\mathcal{T}_R = \{\mathsf{Task}\,\textcircled{N} : \mathsf{Task} \in \mathcal{T}_E\}$, each of which is the reproduction of the Task by ARMAR-6 with the task representation extracted from N demonstrations. In order to evaluate the reproduction and adaptation of the learned task representations, e.g., the geometric constraints, in new cluttered scenes, the scene is perturbed arbitrarily before each trial of execution. Specifically, the involved objects and the robot hands are placed in arbitrarily different locations within the workspace and the view of the camera. The first image frame captured by the robot is used to parameterize the task with K-VIL's representation. This includes optimizing the local frames, identifying the keypoints, configuring the geometric constraints, and generating keypoint motion trajectories using the learned VMPs. We consider a task learned from N demonstrations and from a third-person view to be *generalizable* when it can be successfully reproduced by the robot with categorical objects in new cluttered scenes. Next, we describe the specifications of each considered task in terms of the collection of demonstrations and successful reproductions by the robot. Table VIII provides the list of considered tasks.

Press Button (PB): A human demonstrates how to open the lid of a kettle by pressing the corresponding button with the tip of the middle finger of either the left or the right hand. The kettle #5 of Fig. 10(d) is considered in this task. Both hands look similar and the demonstrator approaches the button with different hand poses (e.g., see Fig. 11). To reproduce the demonstrated human motion by the robot, we design fixed maps between keypoints of human hands and keypoints on the robot hand. The reproduction of the PB task is considered successful if the robot reaches the button with its fingertip within 5 mm to the target (the button) and if the lid is opened by closing the finger with a small angle.

Fetch Tissue (FT): A human demonstrates how to fetch tissue from two tissue boxes [#3 and #4 in Fig. 10(a)] with different hand poses (see Fig. 12). The tissue boxes #1–3 are used to train the vision models, whereas #4–10 are used for reproduction. The reproduction is considered successful if the robot can successfully grasp the tissue and pull it out of the boxes with a predefined pulling action. Although the shape variations between the tissue boxes #3 and 4 in the demonstrations are not obvious, box #10 introduces large shape variations for reproduction.

Insert Stick (IS): The sticks #2–4 in Fig. 10(f) are used to train the vision models and to demonstrate the insertion task by a human (see Fig. 13). Note that we do not insert the stick into the paper roll, as otherwise, the keypoints will be occluded (we defer tasks with occlusion to future work). No pose variations are considered in this task. However, shape variations are introduced via sticks of different lengths and thicknesses. Moreover, we

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

place the sticks with an initial tilting of $\sim -30°$ to $80°$ in the reproduction, thus extrapolating the demonstration range ($\sim 0°$ to $45°$). A successful reproduction is obtained by placing the lower tip of the stick right above the hole in the center of the paper roll without collision with the paper roll during execution [see Fig. 10(e)].

Pour Water (PW): The vision models for this task are trained with the teacups #5–8 in Fig. 10(b) and the kettles #4 and #5 in Fig. 10(d). A human demonstrates the pouring task several times with the kettle #5 and teacups #1 and #3. The demonstrations incorporate teacup shape variations and kettle pose variations (see Fig. 14). The teacups #1–4 and all the kettles are used in reproduction. The reproduction of the PW task is successful if the spout of the kettle aligns above the rim of the teacup and the kettle is tilted appropriately.

Hang Hat (HH): The rack can be assembled with different lengths of sticks [#6–10 in Fig. 10(f)]. The racks assembled with sticks #7 and #8 are used for training the vision models and for the demonstrations and #6–10 are used for the reproduction. In particular, the stick #7 is used in Fig. 15(a), (b), and (d) and the sticks #7 and #8 are used in Fig. 15(c). Successful reproductions are observed if the rim of the hat is placed on top of the tip of the stick regardless of the stick length and the initial pose of the hat.

Clean Table (CT): The dustpans #2 and #3 and the brushes #3 and #4 in Fig. 10(g) are used for training the vision models and for the demonstrations, while the dustpan #1 and the brushes #1 and #2 are used for the reproduction. The CT is successful if the head of the brush aligns parallel above the edge of the dustpan.

B. Evaluation of K-VIL's Task Representation

As discussed in Section IV-B, K-VIL's task representation can be acquired from one or a few demonstration videos based on the distance and variance criteria. Therefore, the number of demonstrations and the variations in object poses and shapes play an essential role. In this evaluation, we are interested in the following questions.

1) How do task representations learned from a different number of demonstrations affect the performance of task reproduction? In other words, what are the limitations of the task representations learned from scarce demonstrations?
2) How many demonstrations are required to learn generalizable task representations?
3) How do the shape and pose variations contribute to the successful extraction of such task representations?

We evaluate K-VIL in one-shot and few shots IL setups in Section VI-B1 and Section VI-B3, respectively. We finally answer the above questions in Section VI-B4.

1) One-Shot IL: We first apply K-VIL to one-shot IL scenarios, where one demonstration ($N = 1$) is provided for each task. As previously explained, when a single demonstration is provided, K-VIL learns a task representation based on the distance criteria of Section IV-B1, resulting in a set of three linear p2p constraints.

Task extraction: The insertion task is first learned from a single demonstration consisting of inserting the stick #4 of Fig. 10(f) in a paper roll. In this task, the master object is the paper roll, and the slave object is the stick. As shown in Fig. 13(a), K-VIL extracts three keypoints subject to p2p constraints on the stick. Note that the local frame $\mathcal{F}$ and the keypoint $\boldsymbol{k}_1$ are located near the contact point, and $\boldsymbol{k}_2$ is the farthest point on the stick from the paper roll (i.e., the master object). Similarly, in Figs. 11(a), 12(a), 14(a), and 15(a), local frames are constructed on the master objects (i.e., the kettle, the tissue box, the teacup, and the rack, respectively) and three p2p constraints are extracted to fully constrain the pose of the slave objects (i.e., the hand, the hand, the kettle, and the hat, respectively).

Task reproduction: As described in Section V-C, the priorities of the three keypoints are ranked as $\mathrm{Pri}_1 > \mathrm{Pri}_2 > \mathrm{Pri}_3$. This respects the fact that $\boldsymbol{k}_1$ is usually the contact point of two objects and allows the KAC to reproduce the motion of $\boldsymbol{k}_1$ more accurately than the motions of $\boldsymbol{k}_2$ and $\boldsymbol{k}_3$. The first five columns of Table II depicts the examples of reproduction of the insertion task IS learned from a single demonstration (task N: IS ❶), as well as reproductions obtained by removing the priority from KAC as an ablation study (task O: $\mathrm{IS}_{\mathrm{np}}$❶). For illustration purposes, we display the cases with a short stick (#10), a long stick (#6, longer than the sticks used in the demonstration), and an extra long stick (the concatenation of #8 and #9). The largest length difference is $\sim 300\,\mathrm{mm}$. The images in the first row show the ability of K-VIL to correctly adapt the task representations of an insertion task in new scenes. This includes constructing the local frame $\mathcal{F}$ on the master object (the paper roll) and the three p2p geometric constraints in $\mathcal{F}$, identifying the keypoints on the slave object (the sticks) and generating the VMP trajectories. It is important to notice that, without priority in the KAC, the keypoint $\boldsymbol{k}_1$ in $\mathrm{IS}_{\mathrm{np}}$❶-*short* is not able to reach its target as accurately as in IS ❶-*short*. Moreover, as opposed to $\mathrm{IS}_{\mathrm{np}}$❶-*long*, IS ❶-*long* is successfully executed without collision between the stick and the paper roll, thanks to the priority in KAC. However, both IS ❶-*ext. long* and $\mathrm{IS}_{\mathrm{np}}$❶-*ext. long* result in a collision between the stick and the paper roll during execution due to the extended length of the stick compared with the demonstrations. The collision is more acute without priority in the KAC.

Moreover, one-shot VIL may generally fail when the learned geometric constraints are not reachable. This problem is exacerbated when demonstrations are provided from a third-person view. For example, consider Figs. 11(a) and 12(a) that show the task representations learned from a single third-person-view demonstration of the PB and FT task, respectively. The reproductions of such task representations fail (see tasks A: PB ❶ and D: FT ❶ in Table III) due to unreachable target keypoint positions. This also indicates that the task representations learned from one demonstration are not necessarily generalizable enough for motion reproduction.

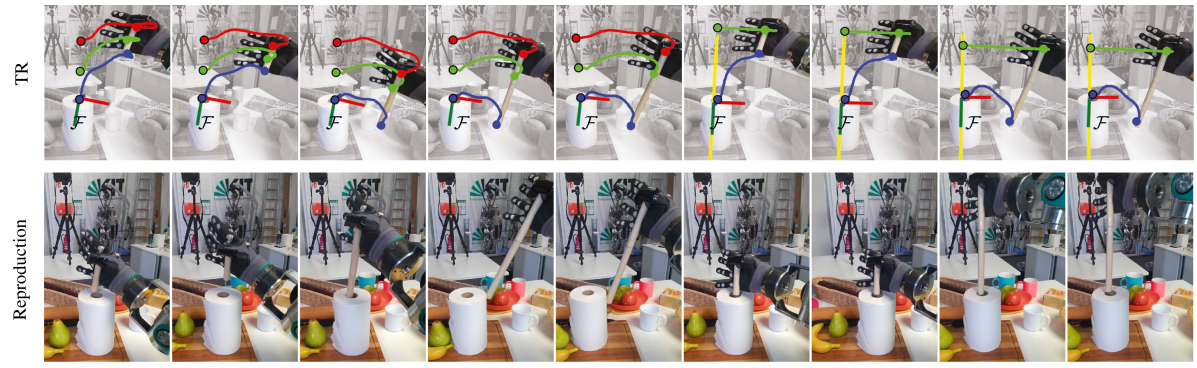
Despite a few failures in the execution, K-VIL's task representations are reliably adapted to new scenes. In other words, K-VIL is able to successfully identify the keypoint positions, locate their targets, and generate the corresponding VMPs by learning their representation, thanks to the combination of the proposed task representation with dense visual correspondence models. It is worth noting that this is already achieved by learning the corresponding representation from a single demonstration. Moreover, thanks to the prioritized KAC, K-VIL can handle

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE II
REPRODUCTION OF THE INSERTION TASKS WITH/WITHOUT PRIORITIES

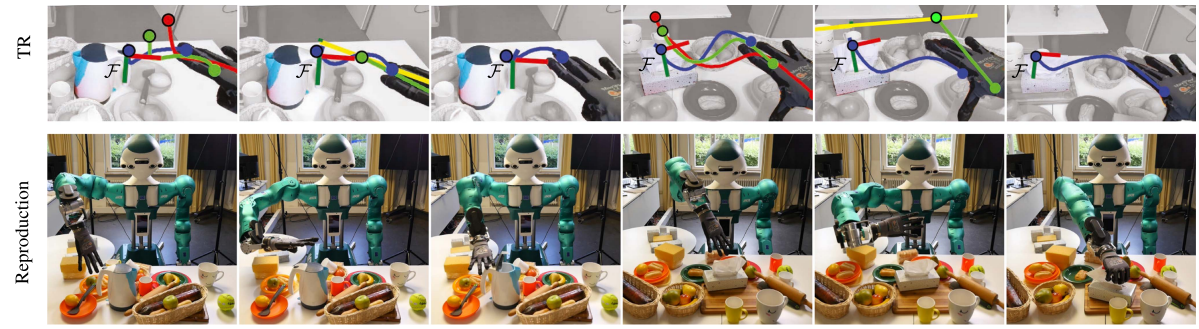
Tasks	N: IS ❶	O: IS$_{np}$ ❶	N: IS ❶	N: IS ❶	O: IS$_{np}$ ❶	P: IS ❸	Q: IS$_{np}$ ❸	P: IS ❸	Q: IS$_{np}$ ❸
stick	short	short	long	ext. long	ext. long	short	short	ext. long	ext. long
	3×p2p					p2p, p2l			



Given an image of the scene before execution, K-VIL's task representation (TR) of each task is used to identify the local frame $\mathcal{F}$, the keypoints ($\boldsymbol{k}_1$, $\boldsymbol{k}_2$, $\boldsymbol{k}_3$), their targets positions ($\boldsymbol{k}_1^g$, $\boldsymbol{k}_2^g$, $\boldsymbol{k}_3^g$), their movement primitives (—,—,—), and the line principal manifold (—) in tasks P and Q. The short, long and extremely long sticks correspond to sticks #10, #6, and the concatenation of #8 and #9. Task names and statistics are listed in Table VIII. The subscript np indicates that the task was reproduced without priority in KAC. The figures in each column are from one of the 20 trials for each task.

TABLE III
REPRODUCTIONS OF TASKS PB AND FT LEARNED FROM DIFFERENT NUMBER OF DEMONSTRATIONS

Tasks	A: PB ❶	B: PB ❸	C: PB ❹	D: FT ❶	E: FT ❸	F: FT ❹
	3×p2p	p2p, p2l	p2p	3×p2p	p2p, p2l	p2p



Reproductions of tasks PB and FT learned from a third-person view without enough demonstrations lead to failure, due to unreachable geometric constraints, see tasks A, B, D, E. Generalizable task representations of PB and FT learned from enough demonstrations can be successfully executed in C and F.

shape variations in categorical objects via the extrapolation of the keypoint target positions. However, it cannot cope with very large shape variations. In other words, providing a single demonstration limits the learning of embodiment-independent generalizable task representations. Therefore, we then evaluate the performance of K-VIL in the case where several demonstrations are available.

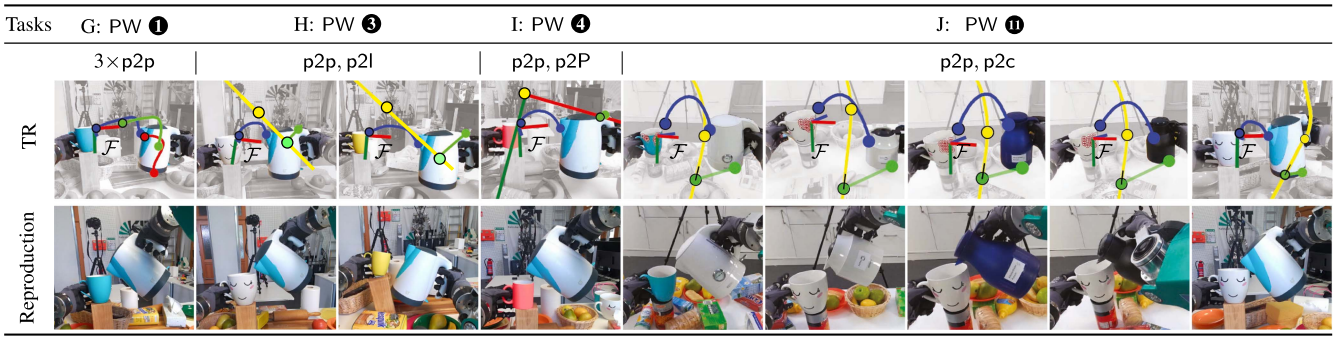
2) Handling Viewpoint Mismatch: When demonstration videos are collected from different viewpoints, e.g., in the CT ❸ task in Fig. 16, we first align the demonstrations into a common viewpoint by projecting the motions of the slave objects (e.g., the brush) into each candidate local frame $\hat{\mathcal{F}}_j$ on the master object (e.g., the dustpan) (see also Section IV-B1). In all $|\mathcal{P}_m|$ aligned common viewpoints, we apply PCE to extract the task representations. This solves the viewpoint mismatch problem of Fig. 16(d). The geometric constraints become obvious in the aligned viewpoint, as shown in Fig. 16(e). K-VIL extracts two p2l constraints for the CT ❸ task, the combination of which forms a parallel constraint. The estimated probability density functions of the two keypoints on the corresponding principal lines ensure their target positions to be above the edge of the dustpan. It is important to note that not only the demonstrations can be recorded from different viewpoints but also the reproduction of the learned skill by the robot can be performed from a viewpoint that is significantly different from any demonstration, as shown in Fig. 16(f) and (h). For the seek of clarity, we discuss the results of K-VIL in the aligned viewpoints in the remaining evaluations, although the demonstrations and reproductions happen in different viewpoints, as discussed for the CT ❸ task.

3) Updating Constraints Incrementally: Here, we apply K-VIL to few-shot IL scenarios where additional demonstrations are incrementally provided for each task. In this case, K-VIL is trained based on the variance criteria to learn more generalizable task representations based on various linear and nonlinear constraints (see Sections IV-B2 and IV-B3). Figs. 11–15 show

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE IV
REPRODUCTIONS OF TASKS PW LEARNED FROM DIFFERENT NUMBER OF DEMONSTRATIONS



For task H, I and J, we mark the learned principal manifold $\mathcal{M}_{\text{line}}$ (—), $\mathcal{M}_{\text{plane}}$ (—, —) and $\mathcal{M}_{\text{curve}}$ (—), respectively. Additionally, the point (○) indicates the mean of the demonstrated targets of keypoint $\boldsymbol{k}_2$ on the corresponding principal manifolds. Other legends as in Table II.

the task representations of each task in $\mathcal{T}_E$ learned by K-VIL from several demonstrations.

a) Task extraction and reproduction of insertion tasks IS*:* Providing several demonstrations allows us to consider variations in the demonstrated task, and thus to extract prioritized geometric constraints. For example, the keypoint $\boldsymbol{k}_1$ in Fig. 13(b) is the most invariant point on the stick across all demonstrations, while $\boldsymbol{k}_2$ is subject to a p2l constraint. Note that this contrasts with Fig. 13(a), where all keypoints were subject to p2p constraints. With such task representation, KAC successfully handles all stick lengths by fulfilling the p2p and p2l constraints, as shown in Table II for IS ❸ (P). Despite a drop in accuracy and precision (see Section VI-C, Table VIII), the KAC without priority still leads to successful task completion in this case (see IS_{np}❸ (Q) in Table II). Overall, the extrapolation abilities of K-VIL are significantly increased by providing three demonstrations instead of 1. Note that, due to the nature of the p2l constraint and the priority mechanism in KAC, sticks of arbitrary length can be handled. As the line manifold goes through both $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$, K-VIL implicitly learns a colinear constraint for the two keypoints.

b) Task extraction and reproduction of PB *and* FT *tasks:* For these two tasks, three demonstrations are not sufficient to completely represent the task, and K-VIL may coincidentally extract a superfluous p2l constraint [see $\mathcal{M}_{\text{line}}$ in Figs. 11(b) and 12(b)]. This forces the robot to place its hand similarly as demonstrated by the human. However, due to the third-person view adopted for the demonstration, this cannot be achieved by the robot, therefore resulting in failed executions of the tasks PB ❸ and FT ❸ as for PB ❶ and FT ❶ (see Table III). The unnecessary p2l constraints are removed by providing K-VIL with an additional demonstration [see Figs. 11(c) and 12(c)], allowing the tasks to be successfully reproduced (see PB ❹ and FT ❹ in Table III).

c) Task extraction and reproduction of PW *tasks:* Similarly to PB and FT tasks, K-VIL's representation obtained from three demonstrations for PW results in superfluous p2p and p2l constraints. Although the task may still be executed by the robot (see PW ❸ in the second column of Table IV), the superfluous constraints may lead to collisions between the kettle and the environment in other cases. For example, in the third column in Table IV, with some specific initial poses of the kettle, the generated VMPs and the line constraints lead to rotation of the kettle in reversed direction during the execution, thus causing reproduction failures. These restrictive task representations are alleviated by providing K-VIL with an additional demonstration. By doing so, the problematic constraints are updated to different types, e.g., the p2l constraint in Fig. 14(b) becomes a p2P constraint in Fig. 14(c). This significantly improves K-VIL's extrapolation abilities, as the p2P constraint is necessary to constrain the kettle in a vertical plane while being less restrictive than the previous p2l constraint. Notice that the density force within the plane constraint ensures that the tilting angle of the kettle is similar to the demonstrations. A successful reproduction of such task representations is shown in Table IV (PW ❹). When enough demonstrations are provided [$N = 11$ in Fig. 14(d)], K-VIL instead extracts a p2c constraint for the keypoint $\boldsymbol{k}_2$ at the bottom of the kettle. The reproduction results with different types of kettles are shown in the last five columns of Table IV (PW ⓫). Intuitively speaking, the p2c constraint aligns better with our understanding of a pouring task. Moreover, the last column in Table IV shows that, given a tilted initial pose of the kettle, KAC is able to correct the pose of the kettle in the end. The set of PW tasks in Table IV also demonstrates that the proposed approach generalizes well to categorical objects with different colors, sizes, and shapes, and is robust to change of background and viewpoints.

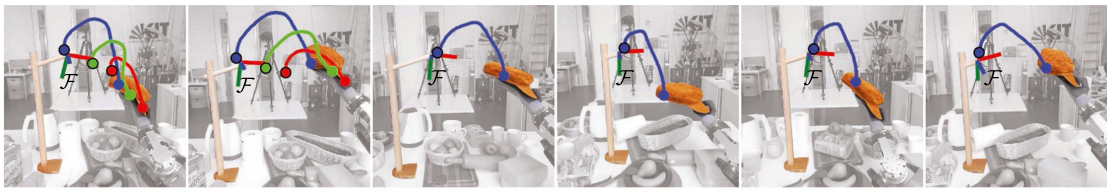
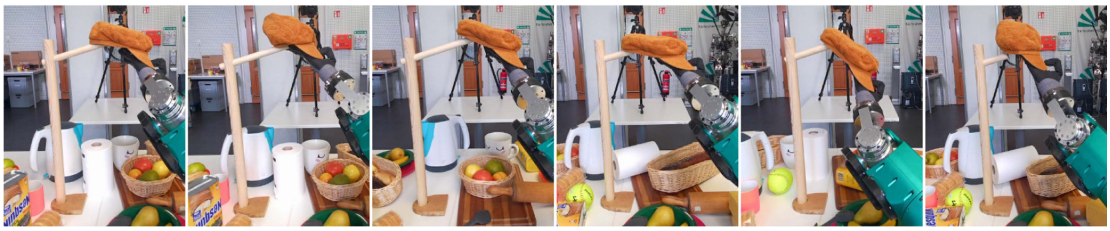
d) Task extraction and reproduction of HH *tasks:* Unlike PB, FT, and PW tasks, K-VIL's representation obtained from three demonstrations for HH does not result in superfluous p2l constraints. Instead, due to the obvious pose variations in the hat, K-VIL consistently extracts a single keypoint $\boldsymbol{k}_1$ on the backside of the hat with a p2p constraint, which encodes the demonstrated position invariances observed in the local frame near the end of the hanging stick. As shown in Table V, the target poses of the hat in HH ❸ vary according to its different initial poses, while in HH ❶, they are fully determined by the three p2p constraints. Fig. 15(c) to (e) shows the influence of the shape variations on the selection of local frames $\mathcal{F}$.

4) Evaluation Summary: As shown by our experiments, K-VIL's task representations allow the successful learning of diverse tasks and their reproduction in new cluttered scenes with large shape and pose variations in categorical objects. As

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE V
REPRODUCTIONS OF TASKS HH

Tasks	A: HH ❶	B: HH ❶	C: HH ❸	D: HH ❸	E: HH ❸	F: HH ❸
	3×p2p		p2p			
TR						
Reproduction						

Legend as in Table II.

TABLE VI
EXTRACTION TASKS AND THE GEOMETRIC CONSTRAINTS OF EACH TASK LEARNED FROM DIFFERENT NUMBERS OF DEMONSTRATIONS

		Number of demonstrations (N)				
	$\mathcal{T}_E$	1	3	4	5	11
a	PB	3 × p2p	p2p, p2l	**p2p**	**p2p**	**p2p**
b	FT	3 × p2p	p2p, p2l	**p2p**	**p2p**	**p2p**
c	PW	3 × p2p	p2p, p2l	**p2p, p2P**	**p2p, p2P**	**p2p, p2c**
d	HH	3 × p2p	**p2p**	**p2p**	**p2p**	**p2p**
e	IS	3 × p2p	**p2p, p2l**	**p2p, p2l**	**p2p, p2l**	**p2p, p2l**
f	CT	3 × p2p	**p2l, p2l**	**p2l, p2l**	**p2l, p2l**	**p2l, p2l**

We mark the cases ( ) where the learned task representations converge, and highlight the cases (in blue) where the learned task representations are generalizable.

opposed to the articles presented in [23], [24], our approach is not constrained to conserving the same viewpoint between demonstrations and reproductions, and thus is more flexible. Here, we further discuss the influence of the number and diversity of the demonstrations on such task representations.

a) Limitations of scarce demonstrations: The task representations learned from scarce demonstrations hinder the performance and the exrtrapolation ability of K-VIL in the following three ways:

1) They may be embodiment-dependent and, thus, cannot be reproduced by the robot, e.g., in Table III for PB ❶, PB ❸, FT ❶, and FT ❸.
2) They may lead to collisions during the execution, e.g., in IS ❸-*ext. long* (see Table II) and PW ❸ for improper kettle start pose (see Table IV).
3) When reproduced successfully, the control accuracy is reduced compared with the task representations learned from more demonstrations (see Section VI-C for details).

These limitations motivate us to evaluate the number of demonstrations that are required to learn a generalizable task representation according to the criteria of Section VI-A for each considered task.

b) Adequate number of demonstrations: Table VI summarizes the extracted geometric constraints of the five extraction tasks for different numbers N of demonstrations. It is interesting to notice that the learned task representations do not change anymore, i.e., converge, after a certain number of demonstrations, e.g., $N = 4$ for PB and FT, $N = 11$ for PW, and $N = 3$ for HH and IS. Moreover, the task representations become generalizable almost at the same time as they converge. As an exception, the representations of PW already generalize with a p2p and a p2P constraint learned from four demonstrations. Importantly, the learned geometric constraints do not have to include a p2p constraint, see e.g., the CT task that is represented by two p2l constraints. Overall, this demonstrates that our approach efficiently extracts generalizable task representations from considerably less demonstrations than state-of-the-art approaches.

c) Object pose and shape variations: Importantly, K-VIL's task representations rely on the variations observed in the demonstrations to extract appropriate constraints. For example, in PB, the pose variations of the demonstrator's hand [see Fig. 11(c)] allow K-VIL to distinguish the tip of the middle finger (used to press the button) from the other candidate points on the hand. In other words, these variations enable the efficient extraction of the keypoint $\boldsymbol{k}_1$ subject to a p2p constraint. The spatial distribution of keypoints across demonstrations is also obtained via pose variations. For instance, this allows K-VIL to associate the keypoint $\boldsymbol{k}_2$ with geometric constraints, such as p2l [see Fig. 14(b) and 16(e)], p2P [see Fig. 14(c)], and p2c [see Fig. 14(d)].

Shape variations also facilitate the efficient extraction of keypoints and geometric constraints. For example, the stick length variations in the insertion task IS allow K-VIL to extract the keypoints $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$ subject to p2p and p2l constraints, respectively, as intuitively shown in Fig. 13(b). In other cases, shape variations in the master objects help to remove redundancy in canonical local frames. For example, all the $J = 300$ canonical local frames on the rack in task HH of Fig. 15(b) are equivalent. This is due to the absence of variations in the rack across the provided demonstrations. In this case, K-VIL selects the local frame $\mathcal{F}$ as the closest on average to the keypoint $\boldsymbol{k}_1$ on the hat. This redundancy is removed by introducing shape

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE VII
POSE VARIATIONS (PV) AND SHAPE VARIATIONS (SV) IN THE DEMONSTRATIONS ALONG WITH THE DETECTED master AND slave OBJECTS

Index	$\mathcal{T}_E$	Role	Object	PV	SV	TR
1	PB	master slave	kettle hand	- ✓	✗ ✗	Figs. 11b and 11c
2	FT	master slave	tissue box hand	- ✓	✓ ✗	Figs. 12b and 12c
3	PW	master slave	teacups kettle	- ✓	✓ ✓	Figs. 14b to 14d
4	HH	master slave	rack hat	- ✓	✗ ✗	Fig. 15b
5	HH	master slave	rack hat	- ✓	✓ ✗	Fig. 15c
6	HH	master slave	rack hat	- ✓	✗ ✓	Figs. 15d and 15e
7	IS	master slave	paper roll stick	- ✗	✗ ✓	Figs. 13b and 13c
8	CT	master slave	dustpan brush	- ✓	✓ ✓	Fig. 16

Corresponding task representations (TR) are linked in the last column. Pose variations of the master objects are not relevant (-) as the local frames representing the object pose are constructed on the masters.

variations in the master objects. For example, by considering two variations of the rack in the task HH, as shown in Fig. 15(c), $\boldsymbol{k}_1$ is position invariant only if it is represented in the local frames near the contact point between the rack and the hat. This allows K-VIL to focus on these local frames and to filter out the others. We showed in Fig. 15(d) that K-VIL's representation converges and remains the same as Fig. 15(b) and (c) even if more demonstrations with large shape and pose variations in the hats are available. Similarly, shape variations in the tissue boxes in FT allow the selection of local frames around the grasping point, while shape variations in the teacups (PW) ensure that the local frames for the pouring task are around their rim. A summary of the effect of pose and shape variations in the considered tasks is given by Table VII. Overall, pose or/and shape variations of master and slave objects in the demonstrations are not only handled by K-VIL but also facilitate the efficient, joint extraction of local frames, keypoints, and geometric constraints.

C. Evaluation of KAC

In the previous Section VI-B, we demonstrated K-VIL's ability to extract generalizable task representations from a small number of demonstrations. In particular, we showed that these task representations successfully adapt to new cluttered scenes with categorical objects, regardless of whether the task can be reproduced by the robot. Therefore, in this section, we evaluate the proposed KAC in terms of control accuracy, control precision (i.e., repeatability), and success rate on the 18 tasks, as described in Section VI-A (see also Table VIII). Each task is reproduced $N_r = 20$ times according to the evaluation protocol in Section VI-A. For each task, we record the trajectories of all relevant keypoints during the execution period. Since we are particularly interested in the regulation behavior of KAC when the keypoints satisfy their corresponding geometric constraints, i.e., when the VMPs finish, we record the keypoint trajectories for an additional 2 s time window, denoted by $\mathcal{T}_{\text{end}}$. These trajectories are then compared with the corresponding keypoints' target trajectories, which correspond to the VMP trajectories for keypoints subject to p2p constraint and to the attractor trajectories for keypoints subject to other types of constraints. Notice that, for the latter, the 3-D position of the attractor is recovered from the corresponding 1-D VMP in the orthogonal direction of the corresponding principal manifold (see also Section V-A).

First, we evaluate the ability of KAC to satisfy the learned keypoints' geometric constraints. To do so, we compute the regulation error of each keypoint during $\mathcal{T}_{\text{end}}$ of each trial as follows:

$$e_v = \frac{1}{T_r}\sum_{t=0}^{T_r} \|\boldsymbol{k}_{l,v}^g(t) - \boldsymbol{k}_{l,v}(t)\|_2, \quad v \in [1, N_r]$$

where T_r is the total time steps recorded in $\mathcal{T}_{\text{end}}$, and $\boldsymbol{k}_{l,v}$ and $\boldsymbol{k}_{l,v}^g$ are the recorded and target positions of the considered keypoint in the vth trial, respectively. Fig. 17 displays the distribution of the keypoint's regulation errors for 20 trials of each task, where the mean values ($\circ$) correspond to the control accuracy

$$\text{Acc.} = \frac{1}{N_r}\sum_{v=0}^{N_r} e_v.$$

Second, we evaluate the control precision (i.e., repeatability) of KAC for all keypoints and all tasks. It is computed as follows:

$$\text{Prec.} = \sqrt{\frac{1}{N_r}\frac{1}{T_r}\sum\nolimits_{v=1}^{N_r}\sum\nolimits_{t=1}^{T_r} \|\boldsymbol{k}_{l,v}(t) - \boldsymbol{\mu}_{\boldsymbol{k}_{l,v}}\|_2^2} \qquad (5)$$

where we defined $\boldsymbol{\mu}_{\boldsymbol{k}_{l,v}} = \frac{1}{T_r}\sum_{t=1}^{T_r} \boldsymbol{k}_{l,v}(t)$. Finally, we also report the success rate obtained for each task according to the evaluation protocols, as described in Section VI-A.

Table VIII presents the evaluation results of KAC in terms of the three aforementioned metrics. As suggested by the qualitative evaluations in Table III, when learned from fewer than four demonstrations, the tasks PB and FT result in a 0% success rate (see A and B, D and E in Table VIII). As discussed in the previous Section VI-B3, this is due to geometric constraints that are unreachable for the robot. In contrast, the task representations learned from four demonstrations are generalizable (see Table VI in Section VI-B4), and thus, KAC reaches sub-millimeter control accuracy and precision, as well as $\geq$90% success rates (see C and F in Table VIII).

As shown in Fig. 17 and Table VIII (G–J), PW ❹ and PW ⓫ outperform PW ❸ in terms of control accuracy, precision, and success rate. This is due to the fact that, in contrast to PW ❸, PW ❹ and PW ⓫ are generalizable (see Tables IV and VI). Although PW ❶ displays a relatively high control precision and success rate, its control accuracy remains low and the pose of the kettle is fully constrained by three p2p constraints, which hinders K-VIL's extrapolation abilities. Interestingly, while $\boldsymbol{k}_1$'s control accuracy in the task PW increases with the number of demonstrations, $\boldsymbol{k}_2$'s highest control accuracy is obtained in PW ❹ (I). This is due to $\boldsymbol{k}_2$'s p2P constraint in PW ❹, which is easier to fulfill than the p2c constraint in PW ⓫. We observe a lower precision in PW ❹ than in PW ⓫ for the same reason.

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE VIII
EVALUATION OF KAC IN TERMS OF CONTROL ACCURACY (ACC.), PRECISION (PREC.), AND SUCCESS RATE (R) FOR EACH CATEGORY OF TASKS WITH TASK REPRESENTATIONS (TR) LEARNED FROM DIFFERENT NUMBERS ❾ OF DEMONSTRATIONS

			Acc. (mm)			Prec. (mm)			
	$\mathcal{T}_R$	TR	k_1	k_2	k_3	k_1	k_2	k_3	R (%)
A	PB ❶	Fig. 11a	×	×	×	×	×	×	0
B	PB ❸	Fig. 11b	×	×	×	×	×	×	0
C	PB ❹	Fig. 11c	0.67	-	-	0.34	-	-	90
D	FT ❶	Fig. 12a	×	×	×	×	×	×	0
E	FT ❸	Fig. 12b	×	×	×	×	×	×	0
F	FT ❹	Fig. 12c	0.82	-	-	0.65	-	-	100
G	PW ❶	Fig. 14a	4.81	15.01	32.09	0.75	0.78	0.72	95
H	PW ❸	Fig. 14b	3.56	3.21	-	1.20	2.12	-	87
I	PW ❹	Fig. 14c	2.01	0.88	-	1.04	5.67	-	94
J	PW ⓫	Fig. 14d	0.93	6.80	-	0.42	0.56	-	100
K	$\mathsf{PW}_{\mathsf{np}}$ ⓫	Fig. 14d	9.71	20.08	-	0.90	1.16	-	100
L	HH ❶	Fig. 15a	13.53	51.33	54.28	0.67	0.67	0.67	95
M	HH ❸	Fig. 15b	1.48	-	-	0.47	-	-	95
N	IS ❶	Fig. 13a	17.77	125.40	51.98	1.55	1.30	1.34	62
O	$\mathsf{IS}_{\mathsf{np}}$ ❶	Fig. 13a	74.55	77.05	13.40	1.03	0.97	1.02	25
P	IS ❸	Fig. 13b	1.01	0.63	-	0.36	0.36	-	95
Q	$\mathsf{IS}_{\mathsf{np}}$ ❸	Fig. 13b	2.46	1.50	-	2.32	1.39	-	90
R	CT ❸	Fig. 16e	8.89	10.22	-	2.04	2.03	-	95

Ablation studies (denoted by $\cdot_{\mathsf{np}}$) are also conducted in tasks K, O, and Q by removing the priority (see Section V-C) from KAC. The cases where data is not available due to failure execution and where a specific keypoint is not required are denoted as × and –, respectively. Gray numbers in N and O indicate inconsequential values.

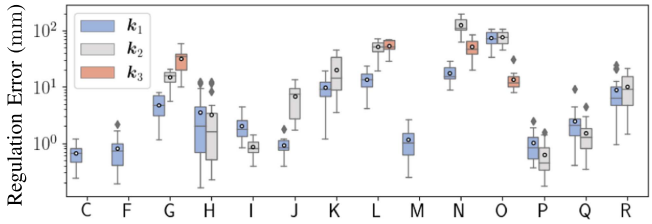


Fig. 17. Regulation errors between each keypoint and its target estimated in $\mathcal{T}_{\text{end}}$ over $N_r = 20$ trials for tasks C–R in Table VIII. The mean regulation error, i.e., control accuracy Acc., is depicted as ○. The box shows the first and third quartiles of the regulation error of each keypoint, with the bar inside it indicating the median.

Similarly as in PW, HH ❸ (M) and IS ❸ (P) are reproduced with higher control accuracy and precision than HH ❶ (L) and IS ❶ (N), respectively. Moreover, thanks to the priority introduced in KAC, HH ❶ (L) and IS ❶ (N) achieve 95% and 62% success rates, respectively, despite the single available demonstration. The lower performance of IS ❶ is explained by the fact that we consider an extremely long stick, which requires high generalization capabilities, in the reproduction (see Table II). In contrast, the hat in HH is only slightly deformable and, thus, results in low shape variations. Notice that, for HH and IS, we do not compare the control accuracy between keypoints subject to different geometric constraints (in gray in Table VIII). As these keypoints are controlled according to different priorities within KAC, their reported accuracy highly depends on the shape variations occurring in these two tasks. For example, when using sticks of various lengths in the insertion task IS ❶ (N), KAC assigns the highest priority to the keypoint $\boldsymbol{k}_1$ so that $\boldsymbol{k}_1$ is obviously controlled with higher accuracy than $\boldsymbol{k}_2$ and $\boldsymbol{k}_3$. Interestingly, $\boldsymbol{k}_2$'s highest regulation error in this task is $\sim$150 mm, which corresponds to the maximum length difference between the sticks used in the demonstration and in the reproduction. Therefore, in HH and IS, the control accuracy of $\boldsymbol{k}_2$ and $\boldsymbol{k}_3$ is rather affected by the experimental setups (e.g., stick lengths) than by KAC. Since the two p2l constraints in CT ❸ share the same priority, the two keypoints are equally controlled toward the region with high likelihood on the corresponding principal lines. The two spring–damper systems for the two keypoints result in equilibrium. Therefore, the control accuracy and the precision of the two keypoints are comparable.

Compared with IS ❶, the ablation study $\mathsf{IS}_{\mathsf{np}}$ ❶ (O), conducted without KAC's priorities, shows a drop in success rate (25%), as $\boldsymbol{k}_3$ achieves the highest accuracy at the expense of $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$. This is expected as the three identical virtual spring–damper systems for $\boldsymbol{k}_1, \boldsymbol{k}_2$, and $\boldsymbol{k}_3$ are in equilibrium, and $\boldsymbol{k}_3$ usually locates in the middle of $\boldsymbol{k}_1$ and $\boldsymbol{k}_2$. As the ablation tasks $\mathsf{PW}_{\mathsf{np}}$ ⓫ (K) and $\mathsf{IS}_{\mathsf{np}}$ ❸ (Q) are reproduced from generalizable task representations, removing KAC's priorities does not affect their success rate. However, the control accuracy and precision drop compared with PW ⓫ (J) and IS ❸ (P).

VII. DISCUSSION

In this article, we proposed the novel K-VIL approach that learns sparse, object-centric, and embodiment-independent task representations from a small set of demonstration videos. K-VIL's task representations were based on the extraction of geometric constraints by a PCE, which covers a wide range of constraints. The proposed PCE enabled one-shot and few-shot VIL and updated the learned task representations when additional demonstrations were incrementally provided, thus endowing them with enhanced extrapolation capabilities. K-VIL's task representations also included task-specific keypoint control policies encoded as VMPs, which were leveraged for task execution by a prioritized KAC. Compared with control policies based on RL or on visual servoing, VMPs allowed a flexible temporal scaling and supported via-points (including start and target position) adaptation. Therefore, they crucially contributed to K-VIL's generalization capabilities by extrapolating the keypoint target positions on the learned principal manifold.

As highlighted in our evaluation, K-VIL consistently learned generalizable task representations for six daily manipulation tasks, which involved highly cluttered scenes, new instances of categorical objects, and large variations in object poses and shapes. Importantly, we showed that the learned task representations converge and become generalizable with significantly fewer demonstrations than state-of-the-art approaches, such as the articles presented in [12], [14], [22], [27]. Interestingly, the sparse keypoint-based geometric constraints extracted by K-VIL mostly aligned with human intuition. This includes the extraction of a single p2p constraint for pressing a button of a

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

pair of p2p and p2l constraints for the insertion task, and of a p2p coupled with a p2c constraint for the pouring task, among others.

It is important to emphasize that the decomposed control and priority mechanism of the KAC allowed us to endow K-VIL with reliable extrapolation capabilities. Indeed, our quantitative evaluations demonstrated K-VIL's ability to reproduce the learned task representations with high control accuracy, control precision, and success rate. Particularly, K-VIL accurately handled very large shape variations in the considered insertion task. In contrast, previous works did not or only briefly discuss the extrapolation capabilities of their approaches [12], [16], [19], [23] . For instance, Jin and Jagersand [12] only showed extrapolation to another instance of the hammer category with very small shape variation without providing any quantitative evaluations.

It is important to note that the variations in object poses and shapes play an essential role in learning generalizable task representations. This is even more relevant when only a small number of demonstrations are provided. Without such variations, K-VIL can still generalize to categorical objects, thanks to the dense visual descriptors, but achieve lower control accuracy, precision, and success rate, and may fail in some extreme cases, e.g., in the one-shot VIL setup.

A. Limitations and Future Work

K-VIL imposes limitations in terms of visual perception models and task representations. On the one hand, we assume that the keypoints are on the surface of objects and omit transparent, reflective, and thin objects (note that this is also discussed in [20], [42]). This hinders K-VIL from being used in many real-world tasks. Furthermore, all keypoints must be visible in the demonstrations, which may not always be enforced in reality. In other words, K-VIL learns from demonstrations with and without viewpoint mismatch, as long as the keypoints of interest are not occluded. In the long run, we believe that the dense correspondence models should be combined with the state-of-the-art scene representation models (e.g., [42]) or with point generative models (e.g., [43]) for better correspondence detection and for tackling the occlusion problems. This would allow the imitator to observe objects that are visually more challenging and to learn the task from demonstrations with (self-)occlusions.

It is worth noticing that K-VIL's keypoints correspond to the subsymbolic parameters of a motion. Therefore, they do not necessarily have a clear semantic interpretation, which is also important for learning comprehensive task models. Bridging the gap between the symbolic and subsymbolic levels remains an important challenge in (visual) IL. Importantly, the symbolic representation of a task [44], [45] also has limitations, which can be alleviated by integrating subsymbolic information. For example, a contain affordance in a pouring task implies that the opening of the spout of the kettle should be placed above the contain affordance region [46]. However, this semantic representation alone cannot describe different types of pouring. For example, pouring beer requires tilting the glass and aligning the beer with the side of a glass. Instead, additional subsymbolic parameters would allow realizing specific styles of task execution.

In this sense, K-VIL deals with the subsymbolic part of the task. Namely, its ability to update the geometric constraints allows us the following:

1) to reproduce a task with a specific style;
2) to eliminate unnecessary keypoints and geometric constraints and to update the distribution of the keypoints on the extracted constraints when more demonstration styles are available.

K-VIL may then be augmented with an extraction method [47] to estimate the links between the extracted keypoints and the symbolic task representation. For instance, the probability distribution of the keypoints on their principal manifolds may be used to determine the affordance regions [46], [48], the spatial relations [49], and the grasping or effect points [10], [47]. We will investigate these aspects in our future work.

In this article, we only considered unimanual manipulation tasks that can be modeled as the combination of five basic geometric constraints in Fig. 4 in a single layer of master–slave relationship. As future work, we plan to extend K-VIL to include other types of geometric constraint and to bimanual manipulation tasks by considering bimanual coordination strategies [50] and a hierarchy of master–slave relationships. Moreover, we will extend K-VIL for periodic motions, such as stirring or wiping motions [23], as well as for handling articulated objects [51], [52].

References

[1] A. Bandura and R. H. Walters, *Social Learning Theory*, vol. 1. Englewood Cliffs, NJ, USA: Prentice Hall, 1977.

[2] D. M. Wolpert, J. Diedrichsen, and J. R. Flanagan, "Principles of sensorimotor learning," *Nature Rev. Neurosci.*, vol. 12, no. 12, pp. 739–751, 2011.

[3] C. J. Burke, P. N. Tobler, M. Baddeley, and W. Schultz, "Neural mechanisms of observational learning," *Proc. Nat. Acad. Sci.*, vol. 107, no. 32, pp. 14431–14436, 2010.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[5] P.R. Florence, L. Manuelli, and R. Tedrake, "Dense Object Nets: Learning dense visual object descriptors by and for robotic manipulation," in *Proc. 2nd Conf. Robot Learn.*, 2018, vol. 87, pp. 373–385. [Online]. Available: https://proceedings.mlr.press/v87/florence18a.html

[6] K. Meng and A. Eloyan, "Principal manifold estimation via model complexity selection," *J. Roy. Statist. Soc. B, Statist. Methodol.*, vol. 83, no. 2, pp. 369–394, 2021.

[7] W. Gao and R. Tedrake, "kPAM 2.0: Feedback control for category-level robotic manipulation," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2962–2969, Apr. 2021.

[8] R. Xu, F.-J. Chu, C. Tang, W. Liu, and P. A. Vela, "An affordance keypoint detection network for robot manipulation," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2870–2877, Apr. 2021.

[9] M. Sharma and O. Kroemer, "Generalizing object-centric task-axes controllers using keypoints," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 7548–7554.

[10] Z. Qin, K. Fang, Y. Zhu, L. Fei-Fei, and S. Savarese, "KETO: Learning keypoint representations for tool manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 7278–7285.

[11] M. Sharma, J. Liang, J. Zhao, A. Lagrassa, and O. Kroemer, "Learning to compose hierarchical object-centric controllers for robotic manipulation," in *Proc. 4th Conf. Robot Learn.*, 2021, vol. 155, pp. 822–844.

[12] J. Jin and M. Jagersand, "Generalizable task representation learning from human demonstration videos: A geometric approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2022, pp. 2504–2510. [Online]. Available: https://ieeexplore.ieee.org/document/9812195

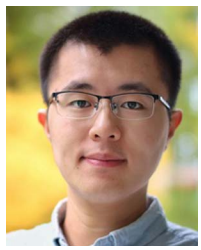
Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[13] Y. Zhou, J. Gao, and T. Asfour, "Learning via-point movement primitives with inter- and extrapolation capabilities," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4301–4308. [Online]. Available: https://ieeexplore.ieee.org/document/8968586/
[14] P. Sharma, D. Pathak, and A. K. Gupta, "Third-person visual imitation learning via decoupled hierarchical controller," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 2593–2603.
[15] L. Smith, N. Dhawan, M. Zhang, P. Abbeel, and S. Levine, "AVID: Learning multi-stage tasks via pixel-level translation of human videos," in *Proc. Robot.: Sci. Syst.*, 2020. [Online]. Available: https://www.roboticsproceedings.org/rss16/p024.html
[16] M. Sieb, Z. Xian, A. Huang, O. Kroemer, and K. Fragkiadaki, "Graph-structured visual imitation," in *Proc. Conf. Robot Learn.*, 2020, vol. 100, pp. 979–989.
[17] Y. Liu, A. Gupta, P. Abbeel, and S. Levine, "Imitation from observation: Learning to imitate behaviors from raw video via context translation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 1118–1125.
[18] D. Dwibedi, J. Tompson, C. Lynch, and P. Sermanet, "Learning actionable representations from visual observations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1577–1584.
[19] H. Karnan, F. Torabi, G. Warnell, and P. Stone, "Adversarial imitation learning from video using a state observer," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 2452–2458.
[20] A. Simeonov et al., "Neural descriptor fields: Se (3)-equivariant object representations for manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2022, pp. 6394–6400.
[21] P. Florence, L. Manuelli, and R. Tedrake, "Self-supervised correspondence in visuomotor policy learning," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 492–499, Apr. 2020.
[22] D. Pathak et al., "Zero-shot visual imitation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 2050–2053.
[23] J. Yang, J. Zhang, C. Settle, A. Rai, R. Antonova, and J. Bohg, "Learning periodic tasks from human demonstrations," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 8658–8665.
[24] J. Pari, N. M. M. Shafiullah, S. P. Arunachalam, and L. Pinto, "The surprising effectiveness of representation learning for visual imitation," in *Proc. Robot.: Sci. Syst.*, 2022, *arXiv:2112.01511*.
[25] F. Torabi, G. Warnell, and P. Stone, "Imitation learning from video by leveraging proprioception," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 3585–3591.
[26] F. Torabi, G. Warnell, and P. Stone, "Generative adversarial imitation from observation," in *Proc. ICML Workshop Imitation, Intent, Interact.*, 2019, *arXiv:1807.06158*.
[27] P. Sermanet et al., "Time-contrastive networks: Self-supervised learning from video," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 1134–1141.
[28] H. Karnan, G. Warnell, X. Xiao, and P. Stone, "Voila: Visual-observation-only imitation learning for autonomous navigation," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 2497–2503.
[29] M. Muhlig, M. Gienger, J. J. Steil, and C. Goerick, "Automatic selection of task spaces for imitation learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2009, pp. 4996–5002.
[30] A. L. P. Ureche, K. Umezawa, Y. Nakamura, and A. Billard, "Task parameterization using continuous constraints extracted from human demonstrations," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1458–1471, Dec. 2015.
[31] Z. Dodds, M. Jägersand, G. Hager, and K. Toyama, "A hierarchical vision architecture for robotic manipulation tasks," in *Proc. Int. Conf. Comput. Vis. Syst.*, 1999, pp. 312–330.
[32] J. P. Hespanha, Z. Dodds, G. D. Hager, and A. S. Morse, "What tasks can be performed with an uncalibrated stereo vision system?," *Int. J. Comput. Vis.*, vol. 35, no. 1, pp. 65–85, 1999.
[33] M. Gridseth, O. Ramirez, C. P. Quintero, and M. Jagersand, "ViTa: Visual task specification interface for manipulation with uncalibrated visual servoing," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 3434–3440.
[34] J. Jin, L. Petrich, M. Dehghan, and M. Jägersand, "A geometric perspective on visual imitation learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5194–5200.
[35] L. Manuelli, Y. Li, P. R. Florence, and R. Tedrake, "Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning," in *Proc. Conf. Robot Learn.*, 2020, pp. 693–710.
[36] C. Lugaresi et al., "MediaPipe: A Framework for Perceiving and Processing Reality," in *Proc. 3rd Workshop Comput. Vis. AR/VR IEEE Comput. Vis. Pattern Recognit.*, 2019.
[37] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Using probabilistic movement primitives in robotics," *Auton. Robots*, vol. 42, no. 3, pp. 529–551, 2018.
[38] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, 2011.
[39] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[40] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," 2018, *arXiv:1801.09847*.
[41] T. Asfour et al., "ARMAR-6: A high-performance humanoid for human-robot collaboration in real-world scenarios," *IEEE Robot. Autom. Mag.*, vol. 26, no. 4, pp. 108–121, Dec. 2019.
[42] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola, "NeRF-supervision: Learning dense object descriptors from neural radiance fields," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 6496–6503.
[43] J. Lei and K. Daniilidis, "CaDeX: Learning canonical deformation coordinate space for dynamic surface representation via neural homeomorphism," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6614–6624.
[44] C. R. G. Dreher and T. Asfour, "Learning temporal task models from human bimanual demonstrations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 7664–7671.
[45] M. Hassanin, S. Khan, and M. Tahtali, "Visual affordance and function understanding: A survey," *ACM Comput. Surv.*, vol. 54, 2018, Art. no. 47.
[46] D. Hadjivelichkov, S. Zwane, L. Agapito, M. P. Deisenroth, and D. Kanoulas, "One-shot transfer of affordance regions? AffCorrs!," in *Proc. Conf. Robot Learn.*, 2023, pp. 550–560.
[47] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies between affordance and geometry: 6-DoF grasp detection via implicit representations," in *Proc. Robot.: Sci. Syst.*, 2021. [Online]. Avaialble: https://www.roboticsproceedings.org/rss17/p024.html
[48] T.-T. Do, A. Nguyen, and I. Reid, "AffordanceNet: An end-to-end deep learning approach for object affordance detection," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 5882–5889.
[49] R. Kartmann, D. Liu, and T. Asfour, "Semantic scene manipulation based on 3D spatial object relations and language instructions," in *Proc. IEEE/RAS 20th Int. Conf. Humanoid Robots*, 2021, pp. 306–313.
[50] F. Krebs and T. Asfour, "A bimanual manipulation taxonomy," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 11031–11038, Oct. 2022.
[51] Z. Xu, Z. He, and S. Song, "Universal manipulation policy network for articulated objects," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2447–2454, Apr. 2022.
[52] R. Wu et al., "VAT-mart: Learning visual action trajectory proposals for manipulating 3D articulated objects," in *Proc. Int. Conf. Learn. Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=iEx3PiooLy

Jianfeng Gao received the bachelor's degree in mechatronics in 2015 from the Beijing Institute of Technology, Beijing, China, and the M.S. degree in mechatronics and information technology in 2018 from the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, where he is currently working toward the Ph.D. degree in computer science with High Performance Humanoid Technologies Lab.

His research focuses on visual imitation learning and compliance control for bimanual manipulation.



Zhi Tao received the bachelor's degree in mechatronics from the Beijing Institute of Technology, Beijing, China, and the M.Sc. degree in mechatronics and information technology from the Karlsruhe Institute of Technology, Karlsruhe, Germany, in 2016 and 2021, respectively.

He is currently a Robotics Algorithm Engineer with Dreame Technology, Suzhou, China. His research interests are the recognition and use of everyday tools in daily life scenes and visual imitation learning.

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Noémie Jaquier (Member, IEEE) received the B.Sc. degree in microengineering, the M.Sc. degree in robotics and autonomous systems, and the Ph.D. degree from the Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2014, 2016, and 2020, respectively.

She is currently a Postdoctoral Researcher with High Performance Humanoid Technologies Lab, Karlsruhe Institute of Technology, Karlsruhe, Germany. Her research combines robot learning, optimization, and control with Riemannian geometry.

Tamim Asfour (Senior Member, IEEE) is a Professor with the Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany, where he is the Head of the High Performance Humanoid Technologies Lab. His research interests include 24/7 humanoid robotics, specifically he studies the mechanoinformatics of humanoids as the synergetic integration of informatics, artificial intelligence, and mechatronics into complete humanoid robot systems, which are able to predict and act to perform versatile tasks in the real world. He is the developer and the leader of the development team of the ARMAR humanoid robot family.

Authorized licensed use limited to: LEHIGH UNIVERSITY. Downloaded on July 27,2023 at 13:30:20 UTC from IEEE Xplore. Restrictions apply.