Sequential Topological Representations for Predictive Models of Deformable Objects

Rika Antonova[*] RIKA.ANTONOVA@STANFORD.EDU
Stanford University, Stanford, CA, USA

Anastasia Varava[*] VARAVA@KTH.SE
Peiyang Shi PYSHI@KTH.SE
J. Frederico Carvalho JFPBDC@KTH.SE
Danica Kragic DANI@KTH.SE
KTH Royal Institute of Technology, Stockholm, Sweden

Abstract

Deformable objects present a formidable challenge for robotic manipulation due to the lack of canonical low-dimensional representations and the difficulty of capturing, predicting, and controlling such objects. We construct compact topological representations to capture the state of highly deformable objects that are topologically nontrivial. We develop an approach that tracks the evolution of this topological state through time. Under several mild assumptions, we prove that the topology of the scene and its evolution can be recovered from point clouds representing the scene. Our further contribution is a method to learn predictive models that take a sequence of past point cloud observations as input and predict a sequence of topological states, conditioned on target/future control actions. Our experiments with highly deformable objects in simulation show that the proposed multistep predictive models yield more precise results than those obtained from computational topology libraries. These models can leverage patterns inferred across various objects and offer fast multistep predictions suitable for real-time applications.

1. Introduction

Dealing with highly deformable objects in robotics entails unique challenges. Since the shape of such objects is dynamic, canonical low-dimensional representations suitable for rigid objects mostly fail to capture the information necessary for modeling, planing and control. A black-box approach of training a large neural network (NN) to solve a particular task lacks modularity. With such approaches, new policies need to be trained for each task; moreover, these do not yield interpretable representations. Furthermore, flexible NN models that excel in capturing local features useful for control are not guaranteed to capture high-level structure needed for planning advanced tasks.

Topology can capture global shape properties of objects, such as their connectivity, holes, voids, and spacial relationships between them, while ignoring unnecessary details. In robotic manipulation, notions and tools from topology can be used to efficiently represent scenes, objects, and their states (Stork et al., 2013; Pokorny et al., 2013; Varava et al., 2016; Yan et al., 2020). Topological representations are especially promising for deformable objects, since many topological properties are *invariant* under continuous deformations, and thus capture the shape and behaviour of objects in a succinct way. In this work, we tackle the problem of constructing compact topological represen-

[*] Rika and Anastasia contributed equally. Work was performed at KTH, supported in part by the Knut and Alice Wallenberg Foundation, and European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 884807).

tations for highly deformable objects. Figure 1 illustrates one of the scenarios we consider: putting an apron on a hook. To perform such task, it is crucial to identify and control the neck strap of the apron, while representing other parts of the object explicitly might not be necessary and increases the complexity of the object model. The openings of the apron can be found as *topological features* of the object point cloud without any semantic labeling; a low-dimensional topological state representation of the apron thus consists of the main openings, their location and width.

We propose a rigorous formulation for extracting topological state representations and analyzing their evolution over time. Under several assumptions about system dynamics and observation quality, we prove that it is possible to detect significant topological features (such as the straps of an apron) and observe their dynamics from point clouds without any semantic labels. We propose the *sequential persistent homology* algorithm (seqPH), building upon the persistent homology framework (Edelsbrunner and Harer, 2010) that can infer the topology of static point clouds. We validate the proposed algorithm on simulated scenarios with clothing items and flexible bags.

The proposed seqPH algorithm is directly applicable to settings that use the extracted representations in an offline manner. To make our method suitable for real-time planning and control we propose to learn predictive NN models. These models take a sequence of point cloud observations as input, and output predictions for the relevant topological features up to a horizon of k steps, conditioned on a future/desired sequence of control actions. The resulting multistep predictive models are well-suited for real-time planning and control. Since the proposed topological features are interpretable, various task-specific objective/cost functions can be obtained by employing topological tools, such as the linking number – a topological invariant that captures the 'linking' or 'entanglement' relationship between two curves (Ho and Komura, 2009; Pokorny et al., 2013; Varava et al., 2016). For instance, to hang an apron on a hook we would want to achieve a spatial relationship between the hook and the apron openings, which can be described with the linking number. These features are general and can be used for more complex manipulation scenarios, such as knot tying (Yan et al., 2020) and assistive dressing (Tamei et al., 2011).

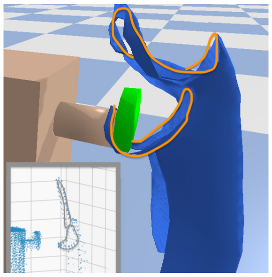
Figure 1: Hook & apron

Overall, our contributions yield a theoretically rigorous approach for tracking topological state and a scalable learning component that enables future applications to real-time planning & control.

2. Related Work in Robotic Manipulation and Learning Predictive Models

Topological representations have been used in robotic manipulation for various purposes. Stork et al. (2013); Pokorny et al. (2013) propose a method for grasping rigid objects with 'holes' (nontrivial first homology group), such as door handles and cups. In Varava et al. (2016), the concept of linking number is used to compute caging grasps for objects with narrow parts. In Yan et al. (2020), 'topological motion primitives' (based on the linking number) are used for tying knots. In this work, we consider one of the most challenging classes of objects: highly deformable objects with nontrivial topology, such as clothing items and flexible bags. The problem of sensing and manipulation with such objects is a part of the more general topic on deformable object manipulation. Recent surveys and benchmarks summarize the relevant scenarios and approaches: Sanchez et al. (2018); Garcia-Camacho et al. (2020). Significant progress has been achieved for tasks such as flattening, spreading and folding, e.g. Van Den Berg et al. (2010); Miller et al. (2012); Lakshmanan et al. (2013); Doumanoglou et al. (2016); Nair et al. (2017); McConachie et al. (2020); Li

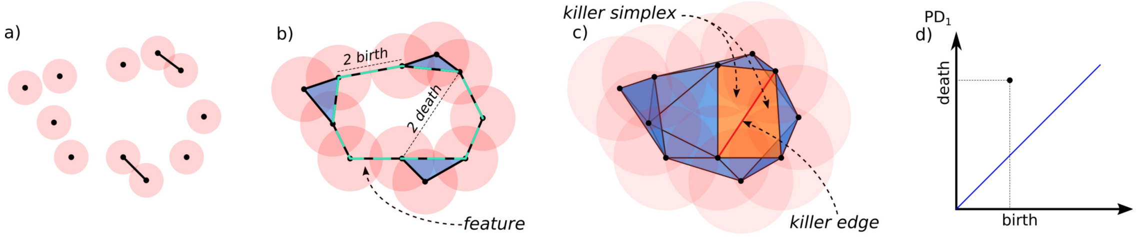

Figure 2: Construction of a Vietoris-Rips complex with growing filtration values. a) Starting from a set of points in black at each filtration level r an edge is drawn between any pair of points that is at a distance $< 2r$. Any clique of k vertices in the resulting graph gives rise to a $k-1$ simplex in the resulting complex. b) When a topological feature (a loop) is formed by adding an edge of length $2x$, we say that the feature has birth-time equal to x. c) The loop dies when it is filled in by 2-dimensional simplices, which occurs when the red edge and consequently its two adjacent orange triangles are added. d) Persistence diagram: death versus birth.

et al. (2018); Lippi et al. (2020). However, many of these works either consider topologically trivial objects (tablecloth, cloth patches, ropes) or lay the objects flat of the table and do not exploit the more complex aspects of their shapes (e.g. loops and holes, cylindrical parts). More relevant to our work are the sub-tasks considered in assistive dressing. Ho and Komura (2009) addressed the motion synthesis problem by constructing a succinct topological representation. Tamei et al. (2011) used it to create a reward/cost function for optimizing a policy of putting a T-shirt on a mannequin. However, this required an accurate motion capture system to track markers for the T-shirt neck and sleeve loops. Recent works aimed to leverage simulators, as in Clegg et al. (2018) with the task of animating character dressing, and in Yu et al. (2017) with putting on a sleeve in simulation and estimating parameters to close the sim-to-real gap. However, these works did not resolve the problem of constructing low-dimensional representations for deformable objects.

One-step forward models $p(X_{t+1}|X_t, U_t)$ have been used ubiquitously in control and model-based reinforcement learning (Deisenroth et al., 2013; Moerland et al., 2020). However, these assume the state to be fully observable. Predictive State Representations (PSRs) (Littman and Sutton, 2001) aimed to address partial observability, but either required simplifying assumptions or were computationally expensive. Nonetheless, PSRs have been used successfully in several areas of robotics (Boots et al., 2013; Stork et al., 2015). Hefny et al. (2018) proposed to encode PSR-like states into recurrent NNs, but training RNNs is non-trivial and this work has yet to be applied to large-scale settings. Recent works proposed adding an objective to reconstruct future states when learning to encode history of observations into lower-dimensional latent states (Yin et al., 2017; Zintgraf et al., 2019). However, these lack theoretical guarantees regarding what is captured in the latent states and do not support incorporating any structured domain knowledge or representations.

In this work, instead of learning lower-dimensional representations in an unsupervised black-box way, we construct predictive models for interpretable low-dimensional states. We take the middle way between partially observable approaches like PSRs and fully observable single-step forward models. Our models have a fixed size of history and prediction horizons. This allows us to employ NN architectures that offer fast and stable training.

3. Topology Background: Definitions and Notation

Here, we briefly describe the necessary definitions from computational topology (see Koplik (2019) for an informal introduction with animations and Edelsbrunner and Harer (2010) for a formal introduction). Given a point cloud observation, we want to recover the topology of the underlying scene. A *simplicial complex* consists of simplices and provides a way to discretely represent a topological space (the scene, in our case). A k-dimensional *simplex* σ can be defined as a convex hull of $k + 1$ points: a single point (or a vertex) is a 0-dimensional simplex, a segment (or an edge) is a 1-dimensional simplex, a triangle (or a face) is a 2-dimensional simplex, etc. The diameter of a simplex σ is the maximum distance between any 2 points in σ, and is 0 in case σ is a single point. We use a special kind of simplicial complexes:

Definition 1 (Vietoris-Rips complex) *Consider a finite set of points $P \subset \mathbb{R}^n$ and $r > 0$. The* Vietoris-Rips filtration $\mathrm{Rps}(\sigma)$ *is a function of a simplex that is equal to half of its diameter:* $\mathrm{Rps}(\sigma) = 1/2 \max_{p_i, p_j \in \sigma} d(p_i, p_j)$. *The Vietoris-Rips simplicial complex* $\mathrm{VR}_r(P)$ *consists of all simplicies σ such that* $\mathrm{Rps}(\sigma)$ *is less than or equal to r.*

Given $r > 0$, a Vietoris-Rips complex $\mathrm{VR}_r(P)$ can represent a union of balls $B_r(P)$ of radius r centered at the points P: a 0-dimensional simplex is any point from P, a 1-dimensional simplex is any segment between 2 points from P such that the respective balls overlap, a 2-dimensional simplex is any triangle formed by 3 points from P such that all 3 balls overlap pair-wisely, etc. $\mathrm{VR}_r(P)$ exhibits k-dimensional *topological features*, formally referred to as k-dimensional homology classes: connected components ($k = 0$), holes/loops ($k = 1$), voids ($k = 2$), etc. A Vietoris-Rips filtration can be seen as growing the radius r of the balls and considering the respective complexes $\mathrm{VR}_r(P)$. For any $r' > r$, we will have $\mathrm{VR}_r(P) \subseteq \mathrm{VR}_{r'}(P)$. As r grows, the topology of $\mathrm{VR}_r(P)$ changes: connected components merge together, holes appear and then get filled (formally, features become trivial), see Figure 2. *Persistent homology* can be used to find topological features that remain for different values of r, and thus describe the underlying topology of the space approximated by P.

Definition 2 (Topological feature) *A* topological feature *of a point set P is a non-trivial homology class of* $\mathrm{VR}_r(P)$ *for some r. A k-dimensional feature f has a* birth value $birth(f)$ *equal to the smallest filtration value r at which it appears in* $\mathrm{VR}_r(P)$. *Similarly, f has a* death value $death(f)$ *equal to the filtration value at which f is trivial in* $\mathrm{VR}_r(P)$. *The* lifetime *of a feature f is the difference between these values, $LT(f) = death(f) - birth(f)$. The lifetime information can be collected in a* persistence diagram*, denoted $PD_k(P)$ which is a set of the form* $\{(f, birth(f), death(f))\}$.

The lifetime of a feature indicates its significance: features that die soon after being born are likely to be present due to noise in the point cloud, while features with high lifetime values are likely to be present in the underlying true space. In our case, loops with high lifetime correspond to handles and openings of objects. A set of simplices comprising a feature f is called a *representative* $\hat{f}$ of f. Each feature can have multiple representatives. Two representatives of the same feature are called *homologous*. The death of a feature occurs when a certain filtration value is passed and the interior of the feature gets filled in by a simplex, or, formally, becomes trivial (for example, in dimension 1: a hole is filled in by triangles). This leads us to consider the simplices that "kill" the feature. In the Vietoris-Rips case, a simplex is added to the filtration as soon as all its edges are added. Thus, the longest edge is added together with the higher-dimensional simplices adjacent to it. Hence, this edge "kills" the feature.

Definition 3 (Killer edge) *Let f be a k-dimensional topological feature, then the* killer edge *σ of f is the edge with* $\mathrm{Rps}_{death(f)}(P)$ *that leads to the death of f. Similarly, the* killer simplices *of f are the simplices of dimension $k + 1$ that are added to* $\mathrm{VR}_{death(f)}(P)$ *and lead to the death of f.*

4. Sequential Topological State Representations

We now present our theoretical formulation for identifying topological features of the scene and their evolution over time. Let $S_t \subset \mathbb{R}^3$ be the state of the scene at time t. We cannot observe S_t directly, and instead rely on a point cloud observation – a finite set of points $O_t \subseteq S_t$. Under several assumptions about the quality of the observations and the changes between states, we prove that it is possible to recover the topological features of $O_t \subseteq S_t$. Based on this, we design an algorithm for identifying 1-dimensional topological features (loops) in a sequence of observations $O_1, ..., O_T$. To guarantee that the significant topological features of the scene can be recovered, we assume that the observed point cloud covers S_t densely enough: for each point in S_t there is at least one point in O_t that is $\alpha-$close to it (Assumption 1). Given this, Lemma 1 and Lemma 2 show that significant features (those with lifetime higher than α) of S_t can be recovered from the observation O_t.

Assumption 1 (Observation quality) *The space S_t is covered by an α-neighborhood of the sampling O_t, i.e. $S_t \subseteq B_\alpha(O_t)$, where $B_\alpha(O_t) = \bigcup_{x \in O_t} B_\alpha(x)$.*

Lemma 1 (Feature approximation) *For every k-dimensional feature f in $\mathrm{VR}_r(S_t)$ with $k > 0$, there exists some feature f^O in $\mathrm{VR}_{r+\alpha}(O_t)$ so that $f^O \sim f$ (are homologous) in $\mathrm{VR}_{r+\alpha}(S_t)$.*

Proof Note that since $S_t \subseteq B_\alpha(O_t)$, $\mathrm{VR}_r(S_t) \subseteq \mathrm{VR}_r(B_\alpha(O_t))$, which means that for every simplex σ in $\hat{f}$, a representative of f in $\mathrm{VR}_r(O_t)$, $\sigma \in \mathrm{VR}_r(B_\alpha(O_t))$. Now let us consider the case when $k = 1$. In this case we can construct f^O in a straight-forward manner, namely for each edge $\{p, q\}$, let o, o' be the nearest neighbours to p and q, respectively. The we can place the edge $\{o, o'\}$ in f^O (if $o = o'$ there is no edge added). Note that by construction, since $\{p, q\} \in \mathrm{VR}_r(S_t)$, then $\|p - q\| \leq 2r$, and since $p \in B_\alpha(o)$, and $q \in B_\alpha(o')$ $\|o - o'\| \leq \|o - p\| + \|p - q\| + \|q - o'\| \leq 2(r + \alpha)$, and therefore $\{o, o'\} \in \mathrm{VR}_{r+\alpha}(O_t)$. The same construction follows for $k > 1$ from the fact that a simplex of dimension k is in $\mathrm{VR}_{r+\alpha}(O_t)$ if all its edges are. A proof that $f^O \sim f$ in $\mathrm{VR}_{r+\alpha}(S_t)$ can be done directly using the construction, for which we provide a sketch. For each simplex $\sigma = \{p_0, \ldots, p_k\}$ of a feature f, consider the transformed simplex $\sigma^O = \{q_0, \ldots, q_k\}$. Consider the simplices $\hat{\sigma}_i = \{p_0, \ldots, p_i, q_i, \ldots, q_k\}$ for $i = 0, \ldots, k$. The union $\bigcup_{\sigma \in f} \bigcup_{i=0}^{k} \hat{\sigma}_i$ forms a subset $\mathrm{VR}_{r+\alpha}(S_t)$ whose boundary is the union of f and f^O. ■

Lemma 2 (Observability of significant features) *Let O_t be a point cloud approximation of S_t. Any topological feature f in S_t whose lifetime is higher than α has a corresponding topological feature f^O in $PD_k(O_t)$, s.t. $0 \leq birth_O(f^O) - birth_S(f) \leq \alpha, 0 \leq death_O(f^O) - death_S(f) \leq \alpha$.*

Proof First note that $O_t \subseteq S_t$ therefore $birth_O(f^O) \geq birth_S(f)$, and $death_O(f^O) \geq death_S(f)$ which implies the left-hand sides of both inequalities. By Lemma 1 given any representative $\hat{f}$ of f in $\mathrm{VR}_{birth_S(f)}(S_t)$, it can be approximated by $\hat{f}^O$ in $\mathrm{VR}_{birth_S(f)+\alpha}(O_t)$ so that $\hat{f} \sim \hat{f}^O$ in $\mathrm{VR}_{birth_S(f)+\alpha}(O_t)$. Since $LT_S(f) > \alpha$, $\hat{f}^O$ is non-trivial in $\mathrm{VR}_{birth_S(f)+\alpha}(S_t)$ and consequently in $\mathrm{VR}_{birth_S(f)+\alpha}(O_t)$. This implies that $birth_O(f^O) \leq birth_S(f) + \alpha$ which completes the first inequality. Recall that $\mathrm{VR}_{death_S(f)}(S_t) \subseteq \mathrm{VR}_{death_S(f)}(B_\alpha(O_t)) \overset{\phi}{\cong} \mathrm{VR}_{death_S(f)+\alpha}(O_t)$, where the map ϕ is given by $\sigma \mapsto \sigma^O$. Since $death_S(f) > birth_O(f^O)$, $\hat{f}^O$ is in $\mathrm{VR}_{death_S(f)}(S_t)$, where it satisfies $\hat{f}^O \sim \hat{f}$ which is trivial in $\mathrm{VR}_{death_S(f)}(S_t)$. Thus $\hat{f}^O$ is also trivial in $\mathrm{VR}_{death_S(f)}(B_\alpha(O_t))$. Since ϕ preserves trivial classes, and acts as the identity on $\hat{f}^O$, it must be that $\hat{f}^O$ is trivial in $\mathrm{VR}_{death_S(f)+\alpha}(O_t)$. Hence: $death_O(f^O) \leq death_S(f) + \alpha$, completing the second inequality. ■

Next, we analyze how the topology of the states S_t changes over time. We assume that between successive time steps the state S_t undergoes a transformation τ which is small enough, so the displacement of each point is bounded (Assumption 2). With this, we can guarantee that significant topological features are preserved between consecutive states S_t and S_{t+1}, and their lifetime does not change drastically, meaning that wide loops do not suddenly appear or collapse (Lemma 3).

Assumption 2 (Regularity of motion) *Every $x \in S_t$ satisfies $\|\tau(x) - x\| < \varepsilon$, for some $\varepsilon < \frac{1}{2}\alpha$.*

Lemma 3 (Temporal persistence of topological features) *Consider two consecutive states S_t and S_{t+1}. Any topological feature f in $PD_k(S_t)$ with lifetime higher than ε has a corresponding feature in $PD_k(S_{t+1})$, such that: $|birth_{S_t}(f) - birth_{S_{t+1}}(f)| \leq \varepsilon,\ |death_{S_t}(f) - death_{S_{t+1}}(f)| \leq \varepsilon$.*

Proof Since S_t and S_{t+1} have the same set of points, τ only affects the distance between them. From Assumption 2, we know that $\tau(\mathrm{VR}_r(S_t)) \subset \mathrm{VR}_{r+\varepsilon}(S_{t+1})$ for all $r \geq 0$ therefore, given any representative $\hat{f}$ of a topological feature in $\mathrm{VR}_r(S_t)$, it is necessarily the case that $\tau(\hat{f})$ is a representative of the same topological feature in $\mathrm{VR}_{r+\varepsilon}(S_{t+1})$. Furthermore, since the birth and death of this representative corresponds to half the length of some edge which has the property of being the longest edge in some finite set of edges, this length can change by at most 2ε and therefore we have: $|birth_{S_t}(\hat{f}) - birth_{S_{t+1}}(\tau(\hat{f}))| \leq \varepsilon,\ |death_{S_t}(\hat{f}) - death_{S_{t+1}}(\tau(\hat{f}))| \leq \varepsilon$. Since this is true for all representatives, it is true for the feature f (with $\hat{f}$ as a representative of f). ■

We have shown that, under Assumptions 1 and 2, the topology of the scene can be recovered from observations, and, moreover, does not drastically change over time. In theory, this makes it possible to track the topological features. In practice, however, we can have several topological features with similar lifetime, and to distinguish them we need additional information about their *geometric* location. For this, we will identify each feature with the corresponding killer edge, and observe how it moves over time. Thus, we assume that a killer edge representing each feature can be uniquely identified. Furthermore, we assume that different topological features in S_t are located far enough from each other, so it is possible to distinguish their respective killer edges based on Hausdorff distance $d_H(.,.)$ (Assumption 3). Then, Lemma 4 shows that killer edges from S_t can be recovered given an observation O_t. Lemma 5 shows that the motion of killer edges is limited between consecutive states S_t and S_{t+1}, and thus we can track them over time given the respective observations O_t and O_{t+1}: if two killer edges in consecutive observations are close enough to each other, then they correspond to the same topological feature in S_t and S_{t+1} (Corollary 1).

Assumption 3 (Uniqueness of killer edges and feature separation) *For each S_t and each feature f in $PD_k(S_t)$ with $LT(f) > \alpha$, there is a unique edge $\sigma_{kill}(f)$ that kills f, and any other edge σ' with filtration value satisfying $|\operatorname{Rps}(\sigma_{kill}(f)) - \operatorname{Rps}(\sigma')| \leq \alpha$ is β-close to $\sigma_{kill}(f)$: $d_H(\sigma', \sigma_{kill}(f)) \leq \beta$. Furthermore, any 2 distinct features f_1, f_2 in $PD_k(S_t)$ are separated: $d_H(\sigma_{kill}(f_1), \sigma_{kill}(f_2)) > 2\alpha + \beta + \varepsilon$.*

Lemma 4 (Recovering killer edges from observations) *Consider a feature $f^S \in PD_k(S_t)$, for any $k \geq 1$. There exists a corresponding feature $f^O \in PD_k(O_t)$ s.t. $d_H(\sigma_{kill}(f^S), \sigma_{kill}(f^O)) \leq \beta$.*

Proof First, $f^O \in PD_k(O_t)$ exists by Lemma 2, and $0 \leq death_O(f^O) - death_S(f^S) \leq \alpha$. Let $\sigma_{kill}(f^O) \in VR_{death_O(f^O)}(O_t)$ be the killer edge of f^O in O_t. Then, $\operatorname{Rps}(\sigma_{kill}(f^O)) = death_O(f^O) \leq death_S(f^S) + \alpha$. By Assumption 3, $d_H(\sigma_{kill}(f^S), \sigma_{kill}(f^O)) \leq \beta$. ■

Lemma 5 (Regularity of motion for killer edges) *Let $\sigma_{kill}(f)$ be the killer edge of a feature $f \in PD_k(S_t)$. Its Hausdorff distance to $\sigma_{kill}(\tau(f))$ that kills its image $\tau(f)$ does not exceed $\beta + \varepsilon$.*

Proof By Lemma 3, we have $|\,\mathrm{Rps}(\sigma_{kill}(f)) - \mathrm{Rps}(\sigma_{kill}(\tau(f)))| \leq \varepsilon$, where $\sigma_{kill}(\tau(f))$ is the killer edge corresponding to $\tau(f)$, which is the feature in $PD_k(S_{t+1})$ corresponding to f. Now, consider $\tau(\sigma_{kill}(f))$ – the image of $\sigma_{kill}(f)$ in S_{t+1}. By Assumption 2, the distance from each vertex of $\sigma_{kill}(f)$ to its image $\tau(\sigma_{kill}(f))$ does not exceed ε, implying $|\,\mathrm{Rps}(\sigma_{kill}(f)) - \mathrm{Rps}(\tau(\sigma_{kill}(f)))| \leq \varepsilon$, and hence $|\,\mathrm{Rps}(\sigma_{kill}(\tau(f))) - \mathrm{Rps}(\tau(\sigma_{kill}(f)))| \leq 2\varepsilon$. Furthermore, $d_H(\sigma_{kill}(f), \tau(\sigma_{kill}(f))) \leq \varepsilon$. Finally, in Assumption 2 we established that $2\varepsilon < \alpha$ whereby Assumption 3 allows us to conclude that:
$d_H(\sigma_{kill}(f), \sigma_{kill}(\tau(f))) \leq d_H(\sigma_{kill}(f), \tau(\sigma_{kill}(f))) + d_H(\tau(\sigma_{kill}(f)), \sigma_{kill}(\tau(f))) \leq \varepsilon + \beta.$ ■

Corollary 1 (Tracking killer edges) *Consider two consecutive observations, O_t and O_{t+1}, and two features $f_o \in PD_k(O_t)$ and $f'_o \in PD_k(O_{t+1})$. Let f_s and f'_s be the features in $PD_k(S_t)$ and $PD_k(S_{t+1})$, corresponding to f_o and f'_o respectively. If f_s and f'_s represent the same feature, then $d_H(\sigma_{kill}(f_o), \sigma_{kill}(f'_o)) \leq 2\alpha + \beta + \varepsilon$.*

Proof By triangle inequality we have: $d_H(\sigma_{kill}(f_o), \sigma_{kill}(f'_o)) \leq d_H^{f_o, f_s, f'_s, f'_o}$, with
$d_H^{f_o, f_s, f'_s, f'_o} = d_H(\sigma_{kill}(f_o), \sigma_{kill}(f_s)) + d_H(\sigma_{kill}(f_s), \sigma_{kill}(f'_s)) + d_H(\sigma_{kill}(f'_s), \sigma_{kill}(f'_o))$.
By Lemma 4 $d_H(\sigma_{kill}(f'_s), \sigma_{kill}(f'_o))$ and $d_H(\sigma_{kill}(f_s), \sigma_{kill}(f_o))$ are both smaller than α, and by Lemma 5 $d_H(\sigma_{kill}(f_s), \sigma_{kill}(f'_s)) \leq \beta + \varepsilon$. And so $d_H(\sigma_{kill}(f_o), \sigma_{kill}(f'_o)) \leq 2\alpha + \beta + \varepsilon$. ■

Algorithmic Procedure (Algorithm 1): Given an observation O_t, we compute a Vietoris-Rips filtration and 1D topological features (loops). PH_t represents a 1D persistent diagram of O_t, and contains 1D topological features together with their birth and death values. We filter out those whose lifetime is smaller than α, as they are likely to appear due to noise. For each remaining loop, we extract the corresponding killer edge and its immediate neighborhood – the two killer triangles and the triangles adjacent to them. Since killer triangles capture the geometric properties of the loop better than a killer edge, in practice, we use them to identify and visualize the loops. The filtration value of a killer triangle corresponds to the radius of the widest part of the loop, see Figure 2. The list $\{\zeta_l\}_{l=1}^{L_t}$ stores a representation ζ_l for each loop loop l in PH_t. ζ_l contains the killer triangles (for loop l), their filtration values and immediate neighborhoods (adjacent triangles); ID_l identifies each loop and is used for tracking (IDs are set arbitrarily in the first iteration). We use $X_t := \{\zeta_l\}_{l=1}^{L_t}$ to denote *a topological state*. The matrix $Dist$ stores Hausdorff distances between the killer edges of O_{t-1} and O_t. Since only those pairs of loops whose birth and death values are similar can correspond to the same topological feature (Corollary 1), we set the distance between them to infinity for other pairs. Given the distance matrix $Dist$, we find the best matching between the loops from O_t and O_{t-1} using the Hungarian algorithm (Kuhn, 1955). This matching, together with the IDs of the loops from the previous time step, provides a consistent labeling of features through time.

5. Predictive Models for Deformable Objects

Using the proposed sequential persistent homology algorithm we gain ability to extract compact topological representations of deformable objects. Such representations can be directly useful for any setting that requires a dataset (modeling, supervised learning, offline planning, etc). However, the best-performing computational geometry libraries still take from 100 milliseconds to several

```
Algorithm 1: seqPH
Input: Sequence of observations $\mathcal{O} = \{O_1, ..., O_T\}$, parameters $\alpha, \beta$ and $\varepsilon$
Output: Sequence $\{X_1, ..., X_T\}$ of topological states
for $O_t \in \mathcal{O}$ do
    $PH_t$ = persistent-homology-1D($O_t$) // compute loops
    $PH_t$.remove-small-features($\alpha$) // remove loops with lifetime $< \alpha$
    $L_t = |PH_t|$ // number of loops in the scene
    $\{\zeta_l\}_{l=1}^{L_t}$ = killer-triangles($PH_t$) // extract loop representation $\zeta_l$ for each loop $l$ in $PH_t$
    $X_t = \{\zeta_l\}_{l=1}^{L_t}$ // new topological state
    $Dist_{i,j} = \infty$ // initialize distances
    for $l \in \{0, ..., L_t\}$ do
        for $l' \in \{0, ..., L_{t-1}\}$ do
            if $|\zeta_{l'}.birth - \zeta_l.birth| < 2\alpha + \varepsilon$ and $|\zeta_{l'}.death - \zeta_l.death| < 2\alpha + \varepsilon$ then
                $dist$ = Hausdorff($\zeta_l$.killer, $\zeta_{l'}$.killer)
                if $dist < 2\alpha + \beta + \varepsilon$ then $Dist_{l,l'}$ = $Dist_{l,l'}$ = $dist$ // Corollary 1
        $\{ID_l\}_{l=1}^{L_t}$ = matching($X_t$, $X_{t-1}$, $Dist$) // matching killer triangles
        $X_t$.update-loop-IDs($\{ID_l\}_{l=1}^{L_t}$) // matching new loop IDs with loops from previous scene
return $\{X_1, ..., X_T\}$
```

seconds per point cloud. Hence, techniques based on computational topology alone would not be feasible for real-time planning and control. One solution could be to learn a forward model $p(O_{t+1}|O_t, U_t)$, which predicts the next high-dimensional state O_{t+1} given the current observation O_t and the vector of control actions U_t. However, predicting point clouds directly with high precision would be an extremely challenging problem. An alternative is to employ an approach similar to filtering or any other approach that infers latent space dynamics $p(X_{t+1}|O_t, U_t)$. While such approaches can successfully model rigid object dynamics, it would be challenging to train these to be highly precise for deformable objects. Hence, planning could be ineffective for longer horizons due to error accumulation when such a single-step model is used sequentially to obtain multistep predictions. Therefore, we propose to learn a multistep predictive model. Requiring the model to predict into the future has been shown to enhance ability to capture non-trivial patterns (Guo et al., 2020). Moreover, by obtaining k predictions from a single forward pass we can avoid a costlier alternative of advancing the model step-by-step for estimating rewards/costs of a multistep trajectory.

5.1. Learning Multistep Predictive Models

For dealing with deformable objects, we propose to learn multi-step predictive models that take point clouds of the scene as input and predict the evolution of the topological state up to a horizon of k steps into the future. We construct a dataset of trajectories of point clouds and the corresponding topological states, which are computed using the algorithm from Section 4 to extract 1-dimensional homologies (i.e. loops). We refer to this approach as seqPH. We then train a predictive model that takes h previous point cloud observations $O_{t\text{-}h:t}$ and future/target control actions $U_{t+1:t+k}$ as input and outputs a sequence of predicted topological states for the next k time steps: $X_{t+1:t+k}$.

Each $X_t := \{\zeta_l\}_{l=1}^{L_t}$ contains a list of topological features ζ_l for each loop l identified by seqPH. ζ_l is comprised of a killer simplex/triangle and its neighboring simplices/triangles, the ID & lifetime of the loop and the Hausdorff distance to the corresponding loop at the previous timestep (see Section 4). We train a neural network $f_{\text{NN}}(O_{t\text{-}h:t}, U_{t+1:t+k})$ to produce $X_{t+1:t+k}$ as output. The supervised training regresses directly on the topological features using an L_2 loss. To create fixed-

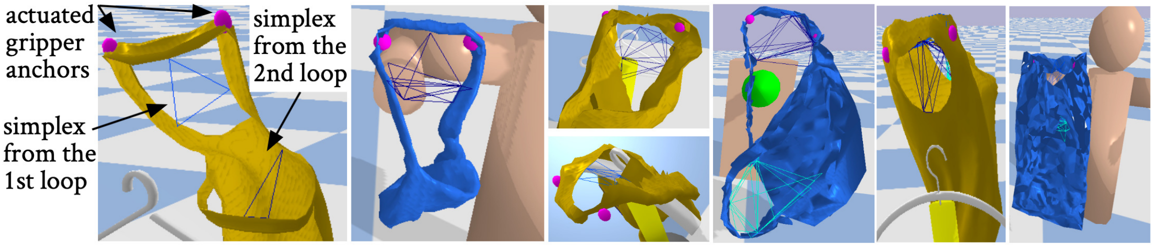

Figure 3: Examples of our scenarios with clothing and bags. Thin lines show killer simplices & the neighbors.

size NN inputs & outputs we pad (or sub-sample) the point clouds and fix the maximum number of loops L_t. For experiments with fully-connected NNs we used 4 hidden layers (512, 512, 256, 128 units). To leverage recently proposed scalable architectures for point cloud processing, we also experimented with an alternative of first passing each input O_t trough a PointNet (Qi et al., 2017).

5.2. Probabilistic Interpretation of Topological States

We construct a probabilistic interpretation of the topological state. For each loop reported by seqPH we compose a mixture of Gaussians that expresses the probability over where this loop is located. The centers of a killer simplex and the centers of the neighboring simplices comprise the means of the components of the mixture. The filtration values associated with each simplex (see Section 4) comprise the weights of the mixture. One natural choice of how to construct the covariances is to treat the 3 vertices $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3 \in \mathbb{R}^3$ of each simplex s as samples from the corresponding Gaussian component $\boldsymbol{x}_s \sim \mathcal{N}(\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)$. We can then compute the unbiased sample covariance of these 3 points: $\boldsymbol{\Sigma}_s = \frac{1}{2}\sum_{l=1}^{3}(\boldsymbol{v}_l - \boldsymbol{\mu}_s)(\boldsymbol{v}_l - \boldsymbol{\mu}_s)^T$. To ensure that $\boldsymbol{\Sigma}$ in non-singular we can place a prior on the covariance and compute posterior treating $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3$ as data. Alternatively, we can use a simpler heuristic of regularizing the covariance with a noise term: $\boldsymbol{\Sigma}_s^{reg} = \boldsymbol{\Sigma}_s + \epsilon \boldsymbol{I}$. To summarize: our probabilistic interpretation of each loop l that is described by simplices $s_0, ..., s_n$ is a Gaussian mixture:

$$p_l(\boldsymbol{x}) = \sum_{i=0}^{n} w_{s_i} \mathcal{N}(\boldsymbol{x} | \boldsymbol{\mu}_{s_i}, \boldsymbol{\Sigma}_{s_i}^{reg}) \qquad (1)$$

where s_0 is the killer simplex for the loop l and $s_1, ..., s_n$ are the n neighbors of this killer simplex; w_{s_i} is the filtration values of the simplices (the death time in the case of the killer simplex), and $\boldsymbol{\mu}_{s_i}, \boldsymbol{\Sigma}_{s_i}^{reg}$ are computed based on the vertices of each simplex s_i as explained above.

6. Experiments

We created PyBullet (Coumans and Bai, 2016–2019) simulations with objects that have non-trivial topology: clothing items and deformable bags. At the beginning of each episode two gripper anchors were attached to the deformable object in the scene. They were actuated (with a simple PD controller) to approach the target area with a hanger, a hook or a mannequin figure (Figure 3).

For training predictive models we collected 23,000 trajectories, pairing randomly the deformable & rigid objects in the scene, and randomizing trajectories of the gripper anchors. We also varied elastic and bending stiffness to emulate cloth/deformable materials with various properties.

Figure 4 illustrates topological states extracted by seqPH (from Section 4) versus those obtained using a predictive NN model (from Section 5). NN gets as input point clouds from the previous $h = 16$ states (not visualized) and the sequence of proposed actions for the steps $t{+}1, ..., t{+}60$. NN returns predicted topological states for each of the 60 steps into the future. We visualize predictions for step $t+1$ and $t+60$. Each loop is expressed by a Gaussian mixture, visualized by plotting

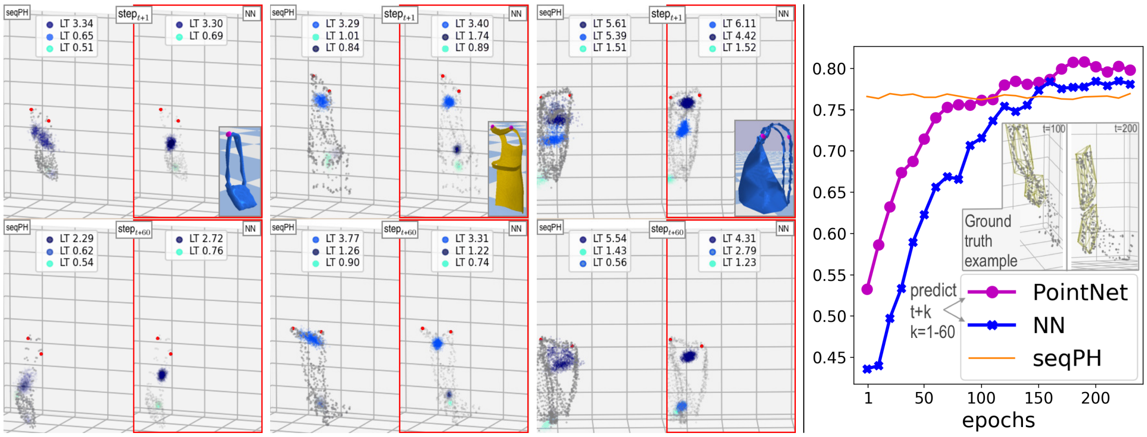

Figure 4: The left sides in each group show results from seqPH. The right sides (with red margins) show mixtures predicted by NN; NN gets point clouds from steps t-$h:t$, so the current point cloud is not given as input to NN, it is only visualized for easier interpretation. Left column: results for the small bag object; the dominant mixture/loop (dark blue) has a long lifetime (LT), indicating the loop is large. Middle column: apron (dominant mixture in light blue). Right column: backpack; mixtures for handles in dark & light blue; the other mixture (in cyan) is phantom, but can be advantageous if it consistently tracks a certain object part. Right plot shows evaluation on a set of objects for which we marked approximate ground truth loop locations.

1000 samples. Compared to topological states reported by seqPH, the predictions from NN tend to produce tighter mixtures. This is likely because NN serves as a regularizer, since it is trained on a large number varying trajectories and has to guess the future location of the loop only based on the point clouds from the previous sates and the future/target motion of the gripper anchors. In contrast, topological states extracted from seqPH are based on the point cloud at a given timestep (and loops tracked from the previous steps). Hence, seqPH could be more precise, but could be vulnerable to noise or peculiarities of the current trajectory.

In addition to qualitative evaluation above, we also conducted quantitative evaluation. The latter was highly non-trivial, since the exact ground truth for the topological state was unknown. Hence, we focused on one aspect for which it was tractable to obtain approximate ground truth as follows. We marked a subset of mesh vertices of the main loops in several objects (aprons with one and two loops). Then, we collected a test set of trajectories where these vertices were tracked. To indicate the main area of the loops we computed convex hulls of the tracked points for each loop. The right plot in Figure 4 shows results for the fraction of the test samples (out of 800) where the mean of the mixture with the longest lifetime was inside the convex hull of the true loop area (for objects with two loops: top two mixtures in two true convex hulls). Predictive model with fully connected architecture (labeled NN) matches results from seqPH. Note that seqPH does not do prediction, we report the loops it extracts from the sequence of the point clouds given to it. The plot also shows that our approach can benefit from the more advanced network architectures, such as PointNet, which outperforms NN and even seqPH results. This demonstrates the ability of NN-based learning to benefit from patterns in the whole dataset and correct the occasional mistakes that seqPH makes.

Conclusion: We proposed a topological state representation for deformable objects, provided a theoretical formulation for tracking its evolution over time, and designed a method for training a multistep predictive model to enable real-time applications. The model was tested on scenarios with highly deformable objects and offered fast multistep predictions that improved over both speed and quality of results obtained by employing only computational topology.

References

Byron Boots, Geoffrey Gordon, and Arthur Gretton. Hilbert space embeddings of predictive state representations. *arXiv preprint arXiv:1309.6819*, 2013.

Alexander Clegg, Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. Learning to dress: synthesizing human dressing motion via deep reinforcement learning. In *SIGGRAPH Asia 2018 Technical Papers*, page 179. ACM, 2018.

Erwin Coumans and Yunfei Bai. PyBullet, a Python module for physics simulation for games, robotics and machine learning. 2016–2019.

Marc Peter Deisenroth, Gerhard Neumann, and Jan Peters. *A survey on policy search for robotics*. now publishers, 2013.

Andreas Doumanoglou, Jan Stria, Georgia Peleka, Ioannis Mariolis, Vladimir Petrik, Andreas Kargakos, Libor Wagner, Vaclav Hlavac, Tae-Kyun Kim, and Sotiris Malassiotis. Folding Clothes Autonomously: A Complete Pipeline. *IEEE Trans. Robot.*, 32(6):1461–1478, 2016.

Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

Irene Garcia-Camacho, Martina Lippi, Michael C Welle, Hang Yin, Rika Antonova, Anastasiia Varava, Julia Borras, Carme Torras, Alessandro Marino, Guillem Alenya, et al. Benchmarking bimanual cloth manipulation. *IEEE Robotics and Automation Letters*, 5(2):1111–1118, 2020.

Daniel Guo, Bernardo Avila Pires, Bilal Piot, Jean-bastien Grill, Florent Altché, Rémi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask Reinforcement Learning. *arXiv preprint arXiv:2004.14646*, 2020.

Ahmed Hefny, Zita Marinho, Wen Sun, Siddhartha Srinivasa, and Geoffrey Gordon. Recurrent predictive state policy networks. *arXiv preprint arXiv:1803.01489*, 2018.

Edmond SL Ho and Taku Komura. Character motion synthesis by topology coordinates. In *Computer Graphics Forum*, volume 28, pages 299–308. Wiley Online Library, 2009.

Gary Koplik. Persistent Homology: A Non-Mathy Introduction with Examples. 2019. `https://towardsdatascience.com/persistent-homology-with-examples-1974d4b9c3d0`.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Karthik Lakshmanan, Apoorva Sachdev, Ziang Xie, Dmitry Berenson, Ken Goldberg, and Pieter Abbeel. A constraint-aware motion planning algorithm for robotic folding of clothes. In *Experimental Robotics*, pages 547–562, 2013.

Yinxiao Li, Yan Wang, Yonghao Yue, Danfei Xu, Michael Case, Shih-Fu Chang, Eitan Grinspun, and Peter K Allen. Model-driven feedforward prediction for manipulation of deformable objects. *IEEE Transactions on Automation Science and Engineering*, 15(4):1621–1638, 2018.

Martina Lippi, Petra Poklukar, Michael C Welle, Anastasiia Varava, Hang Yin, Alessandro Marino, and Danica Kragic. Latent space roadmap for visual action planning of deformable and rigid object manipulation. In *Int. Conf. on Intelligent Robotics and Applications*, 2020.

Michael Littman and Richard S Sutton. Predictive representations of state. *Advances in neural information processing systems*, 14:1555–1561, 2001.

Dale McConachie, Mengyao Ruan, and Dmitry Berenson. Interleaving planning and control for deformable object manipulation. In *Robotics Research*, pages 1019–1036. Springer, 2020.

Stephen Miller, Jur Van Den Berg, Mario Fritz, Trevor Darrell, Ken Goldberg, and Pieter Abbeel. A geometric approach to robotic laundry folding. *Int. J. Robot. Res.*, 31(2), 2012.

Thomas M Moerland, Joost Broekens, and Catholijn M Jonker. Model-based reinforcement learning: a survey. *arXiv preprint arXiv:2006.16712*, 2020.

Ashvin Nair, Dian Chen, Pulkit Agrawal, Phillip Isola, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Combining self-supervised learning and imitation for vision-based rope manipulation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2146–2153. IEEE, 2017.

Florian T Pokorny, Johannes A Stork, and Danica Kragic. Grasping objects with holes: A topological approach. In *IEEE International Conference on Robotics and Automation*. IEEE, 2013.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

Jose Sanchez, Juan-Antonio Corrales, Belhassen-Chedli Bouzgarrou, and Youcef Mezouar. Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. *Int. J. Robot. Res.*, 37(7):688–716, 2018.

Johannes A Stork, Florian T Pokorny, and Danica Kragic. A topology-based object representation for clasping, latching and hooking. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 2013.

Johannes A Stork, Carl Henrik Ek, and Danica Kragic. Learning predictive state representations for planning. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3427–3434. IEEE, 2015.

Tomoya Tamei, Takamitsu Matsubara, Akshara Rai, and Tomohiro Shibata. Reinforcement learning of clothing assistance with a dual-arm robot. In *2011 11th IEEE-RAS International Conference on Humanoid Robots*, pages 733–738. IEEE, 2011.

Jur Van Den Berg, Stephen Miller, Ken Goldberg, and Pieter Abbeel. Gravity-based robotic cloth folding. In *Algorithmic Foundations of Robotics IX*, pages 409–424. Springer, 2010.

Anastasiia Varava, Danica Kragic, and Florian T Pokorny. Caging grasps of rigid and partially deformable 3-d objects with double fork and neck features. *IEEE Transactions on Robotics*, 32 (6):1479–1497, 2016.

Mengyuan Yan, Gen Li, Yilin Zhu, and Jeannette Bohg. Learning topological motion primitives for knot planning. *IEEE International Conference on Robotics and Automation*, 2020.

Haiyan Yin, Jianda Chen, and Sinno Jialin Pan. Hashing over predicted future frames for informed exploration of deep reinforcement learning. *arXiv preprint arXiv:1707.00524*, 2017.

Wenhao Yu, Ariel Kapusta, Jie Tan, Charles C Kemp, Greg Turk, and C Karen Liu. Haptic simulation for robot-assisted dressing. In *IEEE Int. Conf. Robot. Autom.*, 2017.

Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. VariBAD: A very good method for Bayes-Adaptive Deep RL via Meta-Learning. *arXiv preprint arXiv:1910.08348*, 2019.