**Supplementary Materials for**

# DoughNet: A Visual Predictive Model for Topological Manipulation of Deformable Objects

Dominik Bauer[1], Zhenjia Xu[1,2], and Shuran Song[1,2]

[1] Columbia University
[2] Stanford University
Corresponding Author: dominik.bauer@columbia.edu
https://dough-net.github.io

## 1 Overview

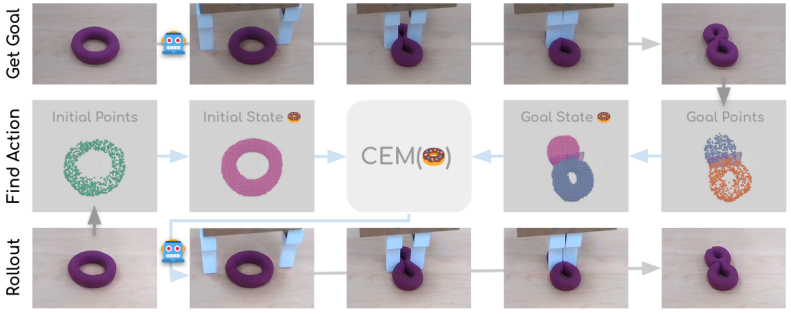
In the following, we provide experiments for **real-world planning with goals defined by both robotic and human manipulation** (i.e., using fingers or palms) in Sec. 2, generalizing from synthetic training data to real test cases. We utilize our predictive model for plan rollouts and a plan scoring that exploits our learned latent space. In addition to examples of our successful goal-directed planning, we investigate the specific challenges in this generalized setting.

In Sec. 3, we show how DoughNet may be used to **predict when topological change occurs in a sequence of observations**. To this end, we investigate the relationships between our learned latent space, the manipulated objects' geometry, their number of components and genera. We qualitatively show that topological change is salient in a t-SNE visualization of subsequent manipulation frames; and quantitatively, we validate that this may be exploited to accurately predict the exact time at which such a change occurs, using a simple classifier.

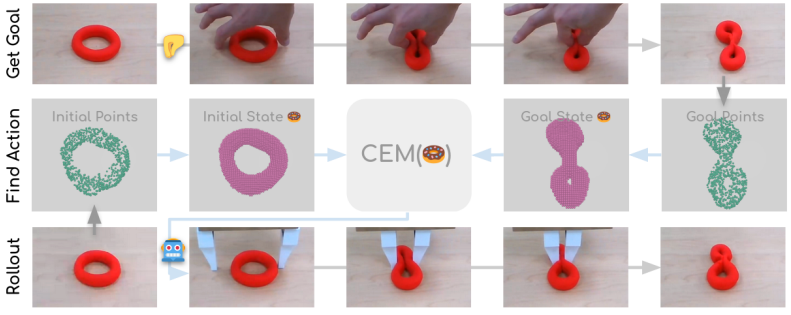
Finally, Sec. 4 further illustrates the concept of **topological manipulation** and how DoughNet may predict plans that (i) achieve different topological goals while maintaining the overall deformed geometry, and (ii) achieve a given topological goal independent of geometry, creating multiple candidate outcomes. We provide a **quantitative planning-performance comparison** with the baselines described in the main text, evaluating the accuracy of the *actual* outcome of the predicted plans with respect to the desired goal.

## 2 Real-world Planning

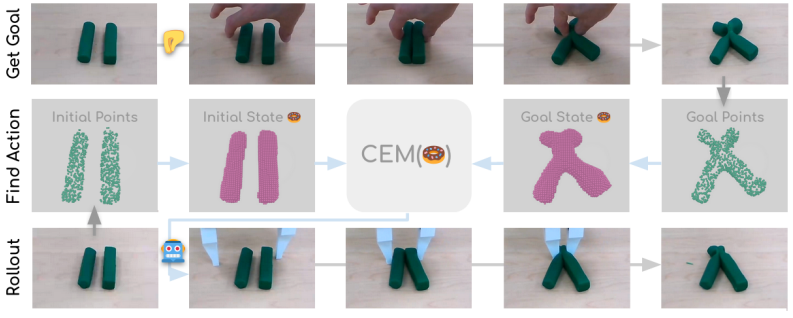
We provide real-world planning results in Fig. 1. Note that these accompany the real-world results in Sec. 5.2 and Fig. 6, as well as the planning results in Sec. 5.3 and Fig. 8 of the main paper. The latter provide complementary detail on how our approach is combined with the CEM-based planner for goal-directed actions. Please also see the videos of the real-world planning results on our project website, https://dough-net.github.io.




**(a)** Robot-defined goal: a doughnut split by the narrow EE.

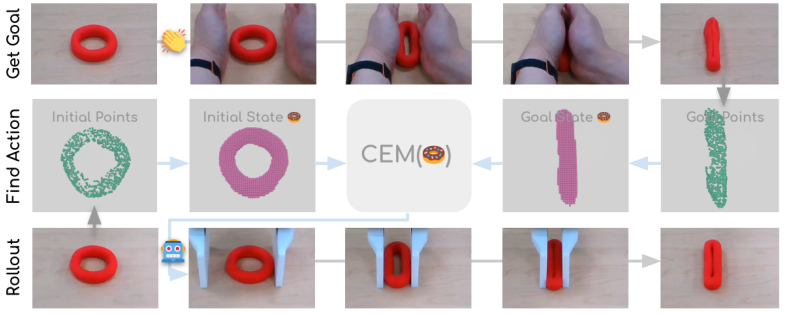



**(b)** Human-defined goal: a doughnut pinched by fingers. Our approach automatically selects the regular EE (∼ fingers) to achieve the goal.




**(c)** Human-defined goal: two rolls pinched by fingers. Our approach automatically selects the regular EE (∼ fingers) to achieve the goal.

**Fig. 1: Real-world Planning Results.** Top: Goal point clouds are created by either a robot (a), human fingers (b, c) or human palms (d); manipulating a doughnut (a, b, d) or two rolls (c). The goals are achievable by one of the held-out EE geometries, namely the narrow (a), regular (b, c) or wide EE (d). Middle: DoughNet 🍩 enables a vanilla CEM-based planner to find the best action (in-plane pose, EE, and final opening width), given the goal and a partial view of the initial state. Bottom: The found action is executed on a robot, successfully achieving the desired goal state (right-most columns).




**(d)** Human-defined goal: two rolls squeezed by palms. Our approach automatically selects the wide EE (∼ palms) to achieve the goal state.

**Fig. 1: Real-world Planning Results (cont.)**.

**From Simulated to Real Manipulation.** As shown in Fig. 1a, DoughNet successfully generalizes from synthetic robotic training data to real robotic test cases. Given a partial point cloud of the goal and initial states, we utilize our encoder to reconstruct and encode them to complete latent states. We employ DoughNet to roll-out sampled actions in a CEM-based planner. Comparing the predicted latent state to the latent goal state by cosine similarity, we are able to select the best-suited EE, in-plane pose and final opening width. By executing this plan on the robot, we accurately recreate the desired goal in terms of geometry and topology, as shown in the right-most column in Fig. 1a.

**From Robot-defined to Human-defined Goals.** Even further from the training data, DoughNet enables planning towards real human-defined goals, created by pinching Play-Doh shapes with fingers, or squeezing them with palms. As shown in Figs. 1b to 1d, our approach is able to identify the correct tool (from held-out EE geometry) to recreate the human manipulation. It is able to successfully bridge the domain gap in terms of EE and created goal shapes. However, while the achieved manipulations are close to the desired goals, we want to also discuss the specific challenges of this transfer to human input. Compared to our synthetic training data, human manipulation creates more irregular object geometries due to the varying "manipulator geometry" and its highly non-linear motion (i.e., articulated, deforming fingers and hands). In Figs. 1b and 1c, the unequal width of the index finger and thumb, combined with additional movement while pinching, creates an indentation that is too wide for the most-suited of the three held-out EE shapes. Hence, in a best effort, a plausible action that pinches part of that region is found by our approach. Similarly, in Fig. 1d, the curvature of the palms affects the created goal state, whereas our planned robotic manipulation (analogous to the training data) may only create flat contact geometries. Combined with a noisy partial observation, our reconstruction tends to represent a too wide goal state in these cases. In turn, the final opening width is slightly wide; although we note that it is still sufficient to merge the doughnut shape into a single component as desired.

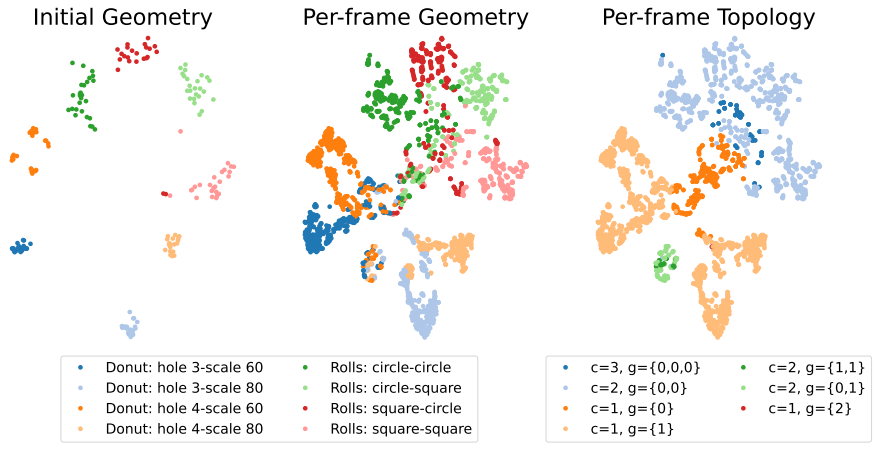


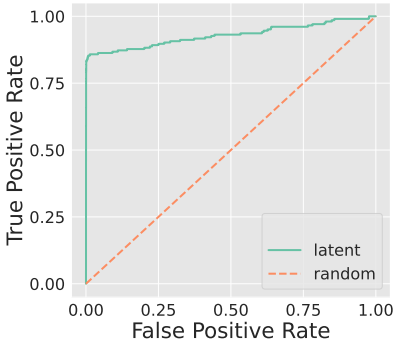

**Fig. 2: t-SNE Visualization.** We show a subset of frames, normalized and reduced to 32 features using PCA before embedding them via t-SNE. Left: The clusters correspond to the initial geometry in the first frame. Middle: Tracking these initial geometries over multiple frames, we observe a regression towards the centroid. Right: Coloring the same clusters by their topology (which may change over time), we find that the regression corresponds to the frequent topological manipulation result of merging to a single component (dark orange). To the top right of this common cluster, we find the topologies that may result from rolls (blue); to the bottom left, the topologies that may result from donuts (light orange, green). We note the equivalent super-cluster boundary for the main types of geometries (donuts and rolls in the left and middle subfigure).

## 3 Predicting Topological Change

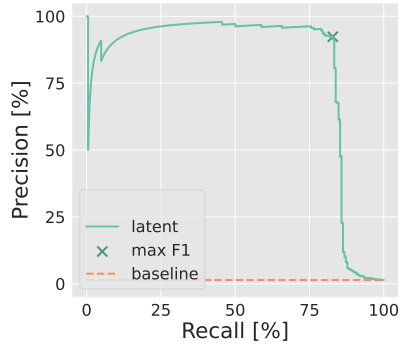
Supporting our observations in the ablation study in Sec. 5.1 and Tab. 2 of the main paper, we find our approach to jointly encode geometry and topology in a well-behaved learned latent space; thereby enabling the latent-space supervision and set-to-set prediction discussed in the ablation study.

The t-SNE visualization in Fig. 2 (left) shows the different initial geometries in (a subset of) the first frames. Each type of scene is clearly separated; we note that the mirror-equivalent configurations of a roll with circular and a roll with square profile are projected to adjacent regions. Moreover, the super-types (scenes with a donut, and scenes with two rolls) occupy distinct half-spaces in the projection. Similarly, although the scene topology changes over subsequent manipulation frames, we observe clusters in Fig. 2 (right) that are consistent with the classes of the resulting topology (in terms of the number of components $c$ and their respective genus $g$).

By tracing individual trajectories, we observe "jumps" in the latent space that seem to correlate with topological changes. Based on this, we hypothesize that the distances in the latent space are sufficient to predict whether two subsequent frames relate to different topologies – that is, if and at which frame such a topological change has occured. To this end, we define a simple classifier.




**(a)** The diagonal line indicates a random binary classifer ("change" or "no topological change" per frame).




**(b)** The *baseline* indicates the positive rate of the dataset, i.e., 1.4% of the 75 raw keyframes (before subsampling).

**Fig. 3: Predicting Topological Change.** The z-score of the Euclidean distances between subsequent frames in our learned *latent* space allows to predict when a topological change occurs. We achieve an $F_1$-score of 87.3% at a z-score threshold of 4.33.

First, we take the mean over the latent set per frame, leaving a 512 dimensional vector. We find that the Euclidean distance between two such vectors is discriminative enough of the observed jumps. To detect such "outliers" in the frame-to-frame distance, we compute the z-score using the statistic for a sequence of frames and evaluate the resulting classifier for varying thresholds on the z-score. Fig. 3 shows the resulting false-positive and true-positive rates in a ROC curve, as well as the corresponding precision-recall curve. Based on this, we determine the ideal threshold by selecting the one that maximizes the $F_1$ score, achieving an accuracy of 87.3%. This experiment illustrates the "topology awareness" of our learned latent space, sufficing a distance-based classifier to predict at which time a topological change happens in an observed manipulation sequence.

## 4 Topological Manipulation

In Fig. 4, the desired – and correctly predicted – topology resolves geometrical ambiguity. Our component-decoder design allows to, e.g., only select actions that create two separate doughnuts in Fig. 4 (bottom). This visualization moreover illustrates that slightly different actions do in fact create different outcomes (in terms of topology in the top row, in terms of geometry in the bottom row). Their prediction and topology-aware scoring, as provided by DoughNet, is thus required in the CEM framework to select the correct action for a given geometry-topology goal.

The results in Tab. 1 indicate the accuracy and runtime efficiency of our approach for planning of topological manipulation. Its prediction quality and

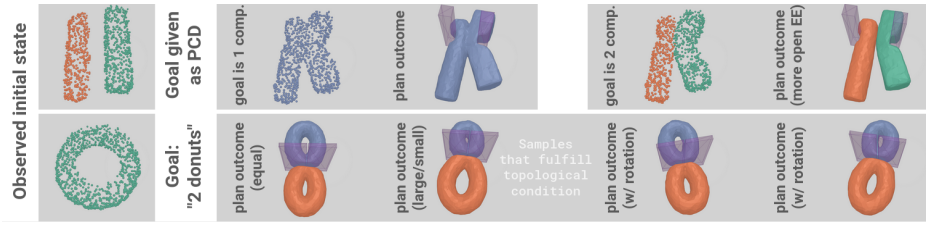



**Fig. 4: Topological Manipulation** using DoughNet under geometrically ambiguous (top) and topology-only goals (bottom).

**Table 1: Comparison of Manipulation Performance.** *CIoU* is computed between goal and actual plan outcome – tested on two *rolls* and two *donut* scenes. *Times*, split into *dynamics prediction / clustering / scoring*, for one planning step; 64 action samples, 13 prediction iterations each.

| | DoughNet | VPM | MPM | RoboCook |
|---|---|---|---|---|
| CIoU$^{\uparrow}$ | **85.0** | 81.4 | 80.7 | 64.1 |
| Times$^{\downarrow}$ | **6/−/0.005s** | 12/421/20s | 823/−/20s | 13/59/20s |

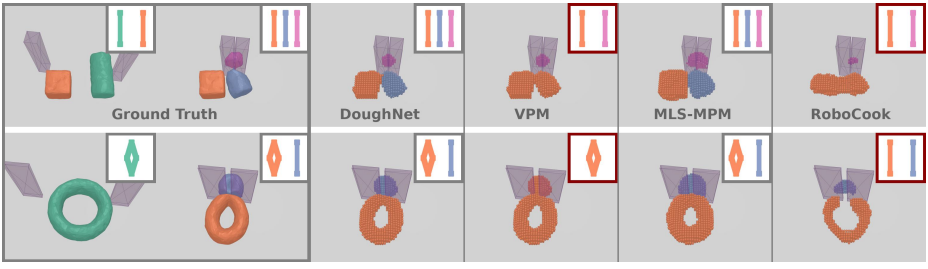



**Fig. 5: Comparison of Prediction Quality**. Initial state and comparison at final time step $t = 13$. Components are color-coded and the boxes in the top right corner of each sample indicate the scene-topology graph (its order has no meaning).

reliable scoring are critical for this. The former is demonstrated in the qualitative comparison in Fig. 5. Notably, the baselines require expensive steps like occupancy decoding (VPM), simulation (MPM), and especially simulation-based checking or clustering to get topology (all baselines). For example, while geometry-only occupancy handles varying topology implicitly, components or genera are not readily available. DoughNet, on the other hand, exploits this by design. Since we compute topology-aware dynamics and scores in latent space, we can skip the decoding and CIoU-based scoring to significantly reduce runtime, as shown in Tab. 1.