

CloSE: A Geometric Shape-Agnostic Cloth State Representation

Removed for Blind Review

Abstract— Cloth manipulation is a difficult problem mainly because of the non-rigid nature of cloth, which makes a good representation of deformation essential. We present a new representation for the deformation-state of clothes. First, we propose the *dGLI disk representation* based on topological indices computed for edge segments of the cloth border that are arranged on a circular grid. The heat-map of the dGLI disk uncovers patterns that correspond to features of the cloth state that are consistent for different shapes, sizes or orientation of the cloth. We then abstract these important features from the dGLI disk into a circle, calling it the *Cloth StatE representation* (CloSE). This representation is compact, continuous, and general for different shapes. We show that this representation is able to accurately predict the fold locations for several simulation clothing datasets. Finally, we also show the strengths of this representation in two relevant applications: semantic labeling and high- and low-level planning. The code and the dataset can be accessed from: <https://anonymousresearcher217.github.io/close-representation/>

I. INTRODUCTION

Cloth manipulation is a challenging problem in robotics mainly because of the infinite-dimensional configuration space of the cloth, i.e. the space of possible cloth positions (or states) in space. Unlike rigid or even articulated objects, where one just needs the object pose and joint configurations, clothes can deform in multiple ways making it extremely difficult to have a simplified representation. In addition, the various shapes, sizes, and mechanical properties of clothes add to the difficulty. For this reason, end-to-end learning-based methods struggle to learn to manipulate clothes even in simulation, because by sampling examples it is nearly impossible to explore all of the configuration space. Recognizing and interpreting their deformation state with such intricate shapes is challenging even in simulation. Several reviews in literature point to the need of a simplified good representations that could pave the way for having more efficient learning methods [1–3].

In recent years, many datasets with both real and simulated clothes have emerged, as summarized in [4]. Real images lack ground truth information on the configuration of cloth, and many reconstruction works [5, 6] rely on realistic simulated images where the ground truth is the full mesh. But even then, there is no real understanding of the configuration of cloth other than comparing to predefined states or recognizing which pixels are sleeves or collars. In other words, there is no definitive solution to reason on the configuration space of cloth. In addition, partial solutions are costly and inefficient because robots could work with much less information than the full mesh. For instance,

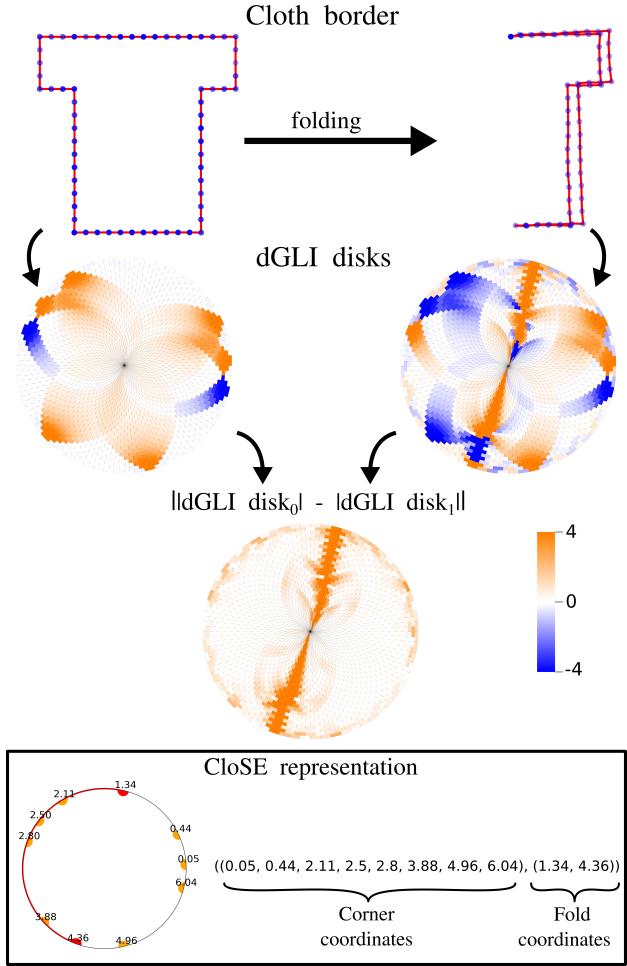
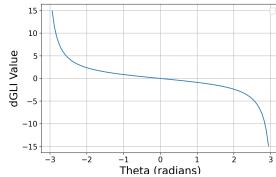


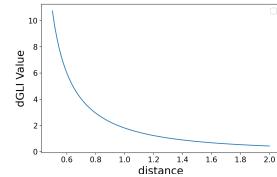
Fig. 1: Example of CloSE derivation: From the border, we compute the dGLI disk where each petal corresponds to a corner. The difference between dGLI disks for the start and end frames allows to extract the fold information, so as to obtain the final CloSE descriptor that is shown graphically and numerically at the bottom of the figure.

previous works have shown how recognizing salient features, corners or borders is enough for effective manipulation [7, 8]. Existing solutions to navigate the configuration space of cloth are based on RGB-D zenithal images of folded states [9, 10], that effectively learn observed transitions between the silhouette of observed states.

For folding tasks, cloth border is an informative indicator of the state of the cloth. Works such as [11] use the border of



(a) Variation of dGLI with angle between segments



(b) Variation of dGLI with distance between segments

Fig. 2: Behavior of dGLI in case of planar segments

the cloth to characterize its state. In particular, they apply a derivative of the Gauss Linking Integral (GLI), a topological index that measures the linking number of curves, to the cloth border to define a representation that separates different cloth states into different clusters. Our paper takes inspiration from this work, but goes beyond by representing the GLI derivative matrix in a novel way that allows to define a very compact representation of the folding state of cloth. Our compact representation, the CloSE representation, is general to different cloth shapes and is also continuous and fully analytical.

Our CloSE representation encodes:

- The shape of the cloth, i.e., the location of the corners of the silhouette of the flat unfolded state of the cloth
- The location of the folds and their orientations (which side folds up)

We propose a method departing from the work of [11], taking all edges along the border and calculating the dGLI. The first important novelty is that we arrange these values in a circular grid instead of a matrix, naming it the dGLI disk. We then abstract out the corners and fold locations from the dGLI disk onto a circle to get the CloSE representation. Fig. 1 shows the starting and final cloth borders of a folding action, their corresponding dGLI disks, the subtraction operation that allows to define the fold coordinates, and the CloSE representation obtained after abstracting out the features.

The contributions of this paper include

- 1) The dGLI disk, a new arrangement for the dGLI coordinates in a disk instead of a matrix, which unravels a hidden structure that is general to any shape, size and pose of the cloth.
- 2) A novel representation, CloSE, that captures this structure in a compact and continuous way.
- 3) Two applications of the CloSE representation: defining an automatic semantic description of the cloth state and planning manipulation sequences.

II. RELATED WORK

The configuration space of cloth is the space of possible cloth states in the Euclidean space. Different representations have been used in literature for robotic manipulation applications. Ranging from the full mesh of a simulated cloth [12] to simplified geometric representations of the silhouettes [13–15], none of these representations permit reasoning in this space in a general way, as they are highly dependent on the

cloth shape. Later learning approaches relied on directly using the RGB or RGB-D as representation of the state of cloth [16–18], usually limited to zenithal views but still allowing to encode folding states. One example is [9], where a lower dimensional latent space of different observed states was encoded as a road-map to plan actions. Since these works map directly from the RGB image space to the latent space without any knowledge of the cloth (implicit or explicit), the latent space representation is not continuous, that is, the intermediate steps cannot be visualized. In addition, these data-based methods require a lot of training data for each cloth shape. Our proposed representation is general for any shape.

The idea of using the Gauss Linking Integral (GLI) for robotics applications is not new. The GLI has been applied to representative curves of the workspace to guide path planning through holes [19, 20], for guiding caging grasps in [21–23], and for planning humanoid robot motions using the GLI to guide reinforcement learning [24]. The idea of applying the GLI to the curve that represents the border of the cloth was used in [11]. For planar curves the GLI is degenerated and clothes on a table are mostly planar, therefore, [11] proposed to apply a directional derivative of the GLI, leading to the dGLI coordinates, which enable distance reasoning in the C-space of cloth. This representation was used in [25] to learn semantic tags of a dataset of folding states of a squared piece of cloth. The dGLI coordinates rely on choosing a few segments in the border to calculate the dGLI, this makes them sensitive to noise in the chosen segments and may not be indicative of how the cloth behaves as a whole. Our work mitigates this by choosing all border edge segments for the dGLI calculation instead of a few. This reveals an underlying structure leading to the low-dimensional CloSE representation.

Our proposed method relies on knowing the curve of the silhouette border. For simulated clothes, we computed the border of the initial flat state since the full mesh is available. For real garments, recent advances have made this increasingly feasible: some works can reconstruct the full cloth mesh [26], while others can directly extract the border from images [8, 27]. However, these approaches are still in their early stages and their accuracy remains limited, which makes them less suitable for robust evaluation at this point. As these methods mature, we expect our representation to naturally extend to real-world cloth perception. However, we want to emphasize that our representation is useful in simulation to automatically label datasets and improve learning methods based on simulation data.

To the best of our knowledge, this is the first time a continuous representation of the folding state of cloth is proposed that is general across different cloth shapes.

III. BACKGROUND - dGLI

The Gauss Linking Integral (GLI) Eq.1 measures the linking number between two closed curves.

$$\mathcal{G}(\gamma_1, \gamma_2) = \frac{1}{4\pi} \int \int \frac{(\gamma_2 - \gamma_1) \cdot [\gamma'_2 \times \gamma'_1]}{\|\gamma_2 - \gamma_1\|^3} \quad (1)$$

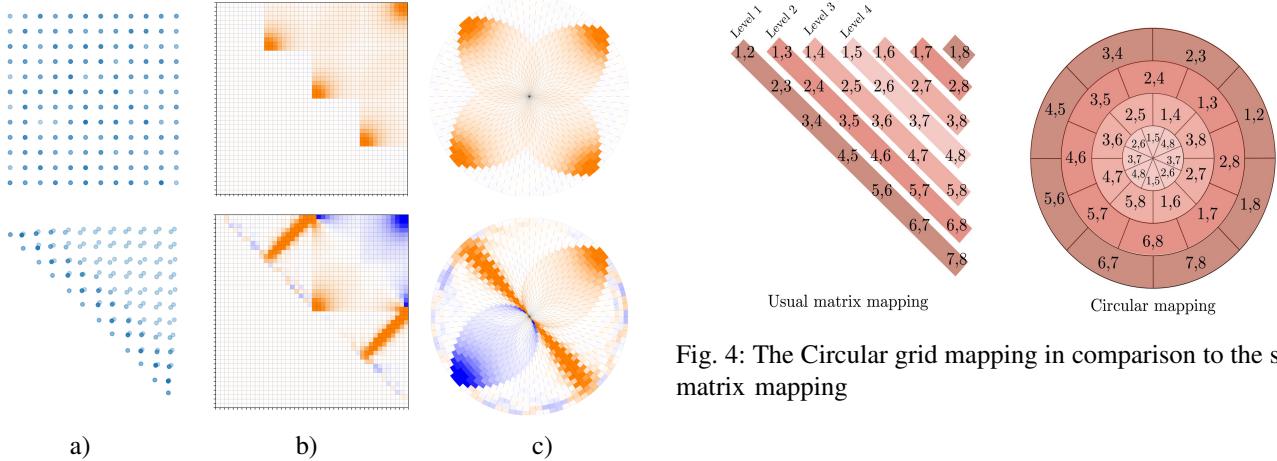


Fig. 3: a) A square napkin mesh. b) The corresponding original dGLI matrix as in [11] and c) Our proposed arrangement in the dGLI disk

The GLI can also be computed between two open curves/segments. Intuitively, this term measures how much the two curves/segments revolve around each other. The writhe matrix so computed between different open segments can be used for various applications [23, 24, 28]. [29] introduces a discrete version of the formula above based on the computation of the GLI between the segments that form the curves (Eq. 2).

$$\begin{aligned} \mathcal{G}(\gamma_{AB}, \gamma_{CD}) &= \arcsin(\vec{n}_a \cdot \vec{n}_b) + \arcsin(\vec{n}_b \cdot \vec{n}_c) \\ &\quad + \arcsin(\vec{n}_c \cdot \vec{n}_d) + \arcsin(\vec{n}_d \cdot \vec{n}_a) \end{aligned} \quad (2)$$

$$\begin{aligned} \vec{n}_a &= \frac{\vec{AC} \times \vec{AD}}{\|\vec{AC} \times \vec{AD}\|}, \quad \vec{n}_b = \frac{\vec{BC} \times \vec{BD}}{\|\vec{BC} \times \vec{BD}\|}, \\ \vec{n}_c &= \frac{\vec{CD} \times \vec{CA}}{\|\vec{CD} \times \vec{CA}\|}, \quad \vec{n}_d = \frac{\vec{DA} \times \vec{DB}}{\|\vec{DA} \times \vec{DB}\|} \end{aligned}$$

Since the measure GLI vanishes when the segments are coplanar, [11] introduced the directional derivative of the GLI along the zenithal axis, the **dGLI** for cloth representation (Eq. 3).

$$\text{dGLI}(\gamma_{AB}, \gamma_{CD}) = \frac{\mathcal{G}(\gamma_{AB+\epsilon}, \gamma_{CD+\epsilon}) - \mathcal{G}(\gamma_{AB}, \gamma_{CD})}{\epsilon} \quad (3)$$

Here, ϵ is a small perturbation in the zenithal direction of the cloth made only to points B and D .¹ In [11] the *dGLI coordinates* were defined as a matrix of all the crossed computations of the dGLI for a selected few edge segments on the cloth border. The heat-map of these dGLI coordinates matrix, showed interesting patterns and could classify different cloth states into classes.

¹For calculating dGLI, in this paper we use $\text{dGLI} = \frac{\mathcal{G}(x+2\epsilon) - \mathcal{G}(x+\epsilon)}{\epsilon}$. This does not change any of our analysis. We do this to handle consecutive edge segments.

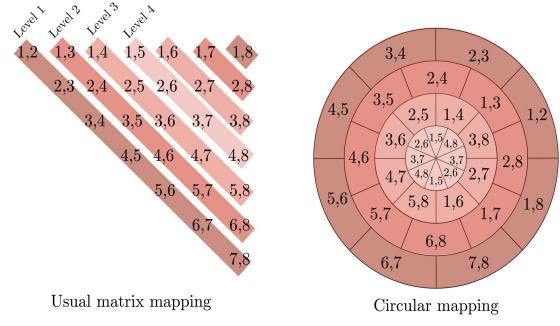


Fig. 4: The Circular grid mapping in comparison to the semi-matrix mapping

To understand the variations in dGLI for two segments, we numerically evaluate the dGLI for planar segments for varying angle and distance. The resulting graphs are shown in Fig. 2. Notice that dGLI is 0 when the lines are co-linear, positive on one side and negative on the other. The value of dGLI also decreases as the distance between the 2 segments increases.

IV. REPRESENTATION

We propose a representation in two steps. First, the **dGLI disk**, a representation inspired by [11], and second, the **CloSE** representation, a compact and continuous representation that is abstracted out of the dGLI disk.

A. The dGLI disk

The dGLI disk is a novel circular arrangement for the dGLI matrix [11]. To illustrate, we take the example of a square napkin folded diagonally as shown in Fig. 3. Unlike in [11], where the authors used only a few edge segments in the computation of the dGLI matrix, we use all the edge segments on the cloth border. This captures more information about the cloth. The matrix arrangement of the dGLI coordinates with all edge segments gives the heat-map in Fig. 3-b. Here, white represents zero values, and orange indicates positives and blues indicates negatives on a linear scale. In this form we can already see interesting patterns. However, they are difficult to interpret. Noticing that the corners appear on the diagonal and the top right corner, we propose a new arrangement for these dGLI coordinates on a circular grid such that the grid cells on the diagonal and the corner are now on the border of this disk and all other cells are arranged inside. This new mapping (Fig. 3-c) displays easy to identify features such as the corners and the fold line. Notice a bright flower pattern in the unfolded dGLI_{disk} indicating 4 corners. The folded dGLI_{disk} adds an orange line between the two folded corners passing through the center, and the petals corresponding to the folded corners change colors. These patterns are consistent across all shapes.

This re-mapping of the dGLI coordinates follows the method visually shown in Fig. 4. This way, the dGLI between the neighboring edge segments is mapped onto the outermost

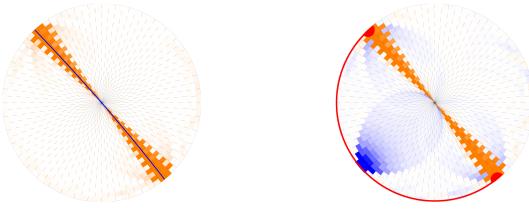


Fig. 5: For the example in Fig. 3: (Left) The dGLI Disk_{diff} highlights the fold line. (Right) The Δ dGLI Disk indicates the side folded.

(first) layer. The dGLI between the second neighbor edge segments is mapped onto the second layer and so on.

The pattern of the corners appears because higher dGLI values are observed when the edges are near and at an angle (refer Fig. 2). These dGLI values are positive or negative depending on their relative orientation. When folded, the angle reverses as seen from the zenithal direction, hence changing sign. In addition, a fold curve appears as the edge segments on the opposite sides of the fold come closer when the cloth is folded. We observe that all fold curves on our dGLI disk pass through the center and their end points denote the location of the fold. This fold curve can be easily isolated by taking the absolute difference between the dGLI disks, that is

$$\text{dGLI Disk}_{\text{diff}} = | |\text{dGLI Disk}_{\text{end}}| - |\text{dGLI Disk}_{\text{start}}| |. \quad (4)$$

Instead, the simple difference between the two dGLI disks helps us identifying the side folded, that is

$$\Delta\text{dGLI Disk} = \text{dGLI Disk}_{\text{end}} - \text{dGLI Disk}_{\text{start}} \quad (5)$$

Fig. 5 shows the dGLI Disk_{diff} and the Δ dGLI Disk for the example in Fig. 3.

B. The Cloth StatE (CloSE) representation

The CloSE representation is a compact and continuous representation that abstracts out the corners and the fold curves from the dGLI disk into a circle². The graphical description of how we computed is shown in Fig. 6.

CloSE is a list of vectors, the first vector encodes the corner locations in radians, i.e., their location on the dGLI disk. The second represents the fold locations and orientation. Each fold is represented by the two end points of the fold line (in radians) while their order dictates the orientation of the fold. i.e., a fold (f_1, f_2) will imply that the cloth border on the route from f_1 to f_2 in the anticlockwise sense is folded. This notation of fold definition ensures that all values of (f_1, f_2) are valid folds, this allows for continuous linear interpolation between folds making the CloSE representation *continuous*. Some examples of interpolations displaying the continuity in CloSE representation are shown in the accompanying video.

²Intuitively, think of the CloSE representation as the cloth border morphed onto a unit circle

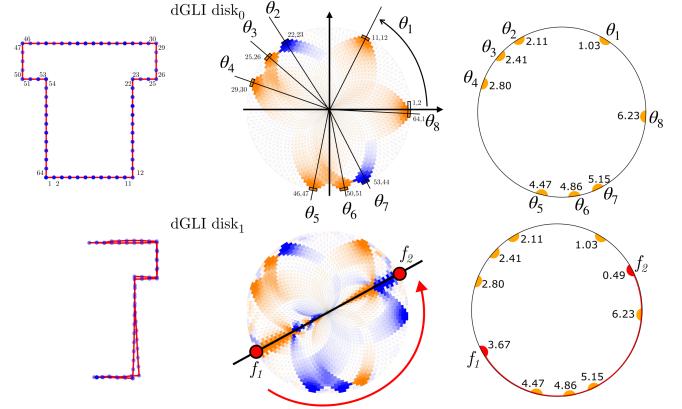


Fig. 6: Computation of the CloSE representation from the dGLI disks. In this case, the corners coordinates are $\{\theta_1, \dots, \theta_n\}$ and the folds (f_1, f_2) .

1) Computation of the corner coordinates $\theta_1, \dots, \theta_n$:

Following the notation in Fig. 6, to get the corners we simply iterate through the first layer in the dGLI disk and note the location of the points that are above a certain threshold. Since the first layer only contains dGLI between immediate neighbors, one corner will only give one bright cell.

This is a naive method that when the mesh is wrinkly, it gives several false corners that correspond to just wrinkles. However, we can still detect accurately the location of the folds and locate them with respect to the CloSE representation, as we will show in our results. The power of our method is that even if all the points of the border are considered corners, we can understand the fold state.

2) Computation of the fold coordinates (f_1, f_2) : The curve(s) that we can see on the dGLI Disk_{diff} represent the fold line(s). We employ a curve fitting method to automatically approximate this fold curve (s).

Given a dGLI Disk_{diff}, we arrange all its values in a vector $\{g_i\}$. We fit a curve of the form $\theta = a_0 + a_1 r$, where r and θ are radius and angle in polar coordinates. For simplicity we only consider $r > 0$. So, two opposite curves represent one fold curve. To fit the curve, we use a gradient-based optimization algorithm³ to find the minimum of

$$\text{loss} = \sum_i k_1 \cdot (\theta_i - \hat{\theta}_i)^2 \cdot (p_1 \cdot \sqrt{g_i}) - k_2 \cdot p_2 \cdot g_i \quad (6)$$

where g_i is the i th value of the dGLI Disk_{diff} in the vector form $\{g_i\}$, θ_i is the angle coordinate corresponding to the value g_i when arranged in the disk and $\hat{\theta}_i = a_0 + a_1 r_i$, being r_i the radius coordinate to the same g_i value in the disk. In addition,

$$p_1 = \exp\left(\frac{-(\theta_i - \hat{\theta}_i)^2}{2 \cdot \tau^2}\right) \quad (7)$$

$$p_2 = \frac{1}{k_3 + (\theta_i - \hat{\theta}_i)^2} \quad (8)$$

with $k_1, k_2, k, \tau \in \mathbb{R}$ being the parameters of the system. We define 20 equidistant curves around the disk as initial points

³L-BFGS-B method in `scipy.optimize.minimize` from the SciPy library

TABLE I: Evaluation results

Dataset	RMSE (Mean, Var)	Frechet (Mean, Var)
Clothilde	0.05, 0.03	0.09, 0.05
Napkin	0.07, 0.02	0.11, 0.04
VR_folding (First Folds)		
Tshirts	0.13, 0.07	0.21, 0.08
Top	0.17, 0.07	0.26, 0.09
Trousers	0.11, 0.04	0.17, 0.05
Skirts	0.12, 0.05	0.18, 0.06
VR_folding (Second Folds)		
Tshirts	0.20, 0.10	0.29, 0.12
Top	0.25, 0.10	0.33, 0.11
Trousers	0.22, 0.08	0.29, 0.09
Skirts	0.24, 0.08	0.33, 0.10

of our optimization, to ensure we converge to all the local minima. We then identify all the local minima with negative loss, each of them being a fold curve.

The loss function is made of two terms, the first term is positive: $k_1(\theta - \hat{\theta})^2 \cdot (p_1 \cdot \sqrt{dG_i})$, penalizes for having high error in curve fitting while also giving importance to the g_i values nearby by weighting it with the term p_1 . This term alone encourages local minimas that are far away from the points of interest, so, we add a second negative loss term, $k_2 \cdot p_2 \cdot g_i$, encouraging the curve to move near the points of interest. The only local minima that correspond to fold lines are the negatives ones, where the second term is dominant. For each of the curves obtained (a_0^i, a_1^i) , the corresponding fold coordinate f_i is obtained evaluating at $r = 1$, that is, $f_i = a_0^i + a_1^i$. For single folds, we will obtain two fold coordinates. For multiple folds, preliminary results are shown in Section VII.

3) *Ordering of (f_1, f_2) to encode the fold orientation:* Corners affected by the fold change sign. Taking the absolute value of $\Delta dGLI$ Disk, non-zero values are always on the same side of the fold curve, the side that is folded. We order the fold coordinates (f_1, f_2) such that the folded corners lie between f_1 to f_2 while traversing in the anti-clockwise sense.

V. EVALUATIONS

We evaluate how well the dGLI disk and hence the CloSE representation captures the fold location. We test this on 3 datasets:

- 1) **The Clothilde dataset:** Dataset we generated with MATLAB using the Clothilde simulator presented in [30], with 46 samples
- 2) **The Napkin dataset** from [25] up to first fold, with 204 samples
- 3) **The VR Folding dataset** from [31] with 5336 samples.

All the datasets contain the start and the end configuration of folding sequences. The Clothilde dataset has few but diverse examples, i.e., clothes of different shapes (Napkins, T-shirts, Circular towels and Trousers) but planar, that is, like towels in different shapes. The napkin dataset has more examples of different folds on a napkin. Though this dataset contains only napkins, the dataset is generated in a virtual environment by tracking human actions on the cloth, thereby

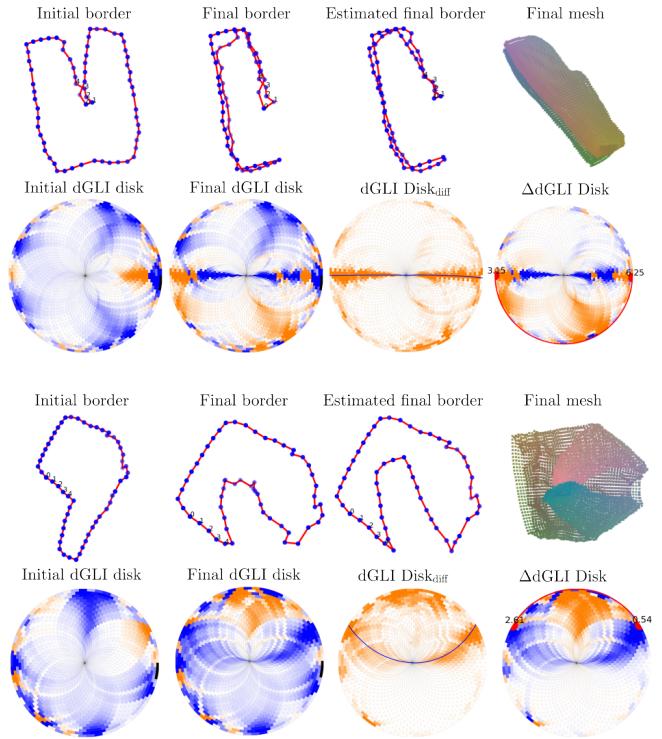


Fig. 7: Examples for the VR folding dataset: the first fold (top) and the second fold (bottom). The second rows show the dGLI disks, start and end, along with the $dGLI\ Disk_{diff}$ and the $\Delta dGLI\ Disk$ indicating the fold curve and side folded respectively

being more realistic and has a less accurate mesh and more noise. This dataset also contains $\approx 40\%$ examples where the cloth is placed at varied location and orientation. The VR Folding dataset, contains simulated but real-like clothes from the CLOTH3D dataset [32], unlike the previous two which assume the cloth is a 2D deformable object. This dataset is also generated in virtual environment by a human manipulator thus making this dataset more complex than the napkin dataset, real-like and very noisy. Also, this dataset is huge - containing 3083 first fold examples and 2253 second fold examples.

To retrieve the fold location from the dGLI Disk, we follow the optimization described in Section IV-B. We set the hyper parameters k_1, k_2, k_3 to 0.1, 0.05 and 0.1 respectively while τ was set to $\pi/4$. For every sample, we estimate the location of the folds. With that information, we can estimate the border after the fold by flipping the side that folds from the initial cloth border along the estimated fold location. We then compare the actual final cloth border with the estimated final cloth border by calculating the Root Mean Squared Error (RMSE) and the Frechet distance between the two curves. Results are shown in Table I. Fig. 7 shows two examples of the initial and final borders, the estimated final border and the corresponding dGLI Disk representations. For fair comparison between different shapes, all examples are scaled to exactly fit into a unit circle with the centroid of the

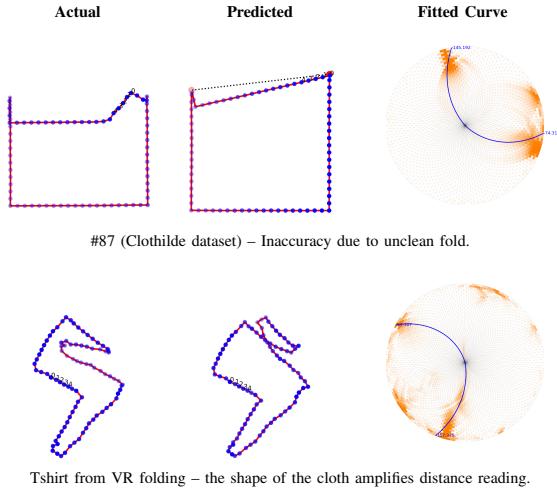


Fig. 8: Examples of ground truth borders, predicted borders, and fitted curves that give the highest errors.

border being the center of the circle.

Unlike the first two datasets, the *VR Folding dataset* provides a full cloth mesh that is closer to real garments. In this case, the mesh border does not directly correspond to the silhouette border we need, requiring a method to estimate it from the start configuration. To do so, we first take the depth image of the cloth in the zenithal direction, mask out the cloth and extract the silhouette border from this mask. We then retrace this border back onto the mesh. This process yields more points than desired leading to a border curve with branches. To obtain a cleaner closed single curve, we construct an ϵ -cover⁴ on the set of all border vertices and use only the centers of these ϵ -balls as the cloth border vertices. Once this border is obtained, the remainder of the procedure follows the same steps as for the first two datasets. For our experiments we used $\epsilon = 0.04$

A. Analysis

Table. I shows how we consistently get good results in all the three datasets. The Clothilde dataset is the cleanest hence we see the lowest error and the prediction error increases as the data becomes more noisy. Note that these predictions are still very accurate even for the noisiest, VR folding dataset (see examples in Fig. 7, and many more on the paper website).

Fig. 8 shows the ground truth border and the predicted border for the worst-case examples. Bigger errors are majorly due to two reasons, First, if the fold is improper (Fig.8[top]), i.e., there is an extra fold in addition to the primary fold confusing the dGLI disk. Second, if the shape is highly concave and the fold occurs in the concave section (Fig.8[bottom]). Here a slight error in fold location prediction leads to a big difference in the final meshes creating a high rmse loss. This kind of error does not indicate any fault in our algorithm but is rather a result of strictness of the measure we used.

⁴An ϵ -cover is a set of balls of radius ϵ that collectively cover all points in the set

All the examples in the Clothilde and the Napkin datasets predicted the existence of exactly one fold curve. However, the VR Folding dataset exhibited errors in 4% and 10% of the cases for the first and second folds, respectively. These errors were not due to limitations of our representation, but rather because the method to estimate the initial border curve failed to find the correct border or because the fold was very bad causing our method to detect more than one fold.

VI. APPLICATIONS

We present two applications that arise naturally from our CloSE representation: automatic semantic labeling and high- and low-level planning.

A. Semantic Labeling

Given the CloSE representation of a cloth state

$$((\theta_1, \dots, \theta_n), (f_1, f_2)),$$

where n is the number of corners, by construction each θ_i and f_i are numbers from $[0, 2\pi)$, because they are angles in radians. Note that θ_i are always in ascending order and correspond to the corners of the cloth in order following the border curve. That is, given the border as a sorted list of vertices, θ_1 is the coordinate corresponding to the first corner we find in that list. Some examples of the used order are shown at the bottom right of Fig. 9.

Therefore, we can identify what is the semantic state of cloth by using interval reasoning in the circle. Each fold is an interval and we can identify which of the θ_i are inside, and these are the folded corners. In addition, each f_i falls inside an edge of the cloth, that is

$$\theta_j < f_i < \theta_{j+1} \text{ for any } i \in 1, 2, \dots, n,$$

accounting for the $2\pi - 0$ equivalence. The distances of f_i to the neighbors θ_i, θ_{i+1} are proportional to the distance of where the fold is on the folded edge of the corresponding corner. Therefore, we can also identify where the fold is on the folded edge.

By simple reasoning we can identify if the cloth is folded in a symmetric half, if the two folded edges are both folded by the same proportion, etc. We cannot identify the shape but we can say how many corners it has. Several examples of the obtained labels are showed in Fig. 9. Note that we obtain these without any learning, by just reasoning on our feature vector, and depending on the desired type of labels we can give more abstract or more informed labels. You can find the code to generate the labels in the accompanying website.

B. Planning

Given the initial border of the cloth, and the CloSE representation of the desired fold state, we can plan a sequence of manipulations to provide the semantic (high-level) instructions and the low-level trajectory of the corners to manipulate.

Since the CloSE representation tracks the folds, one can reason over the cloth's current state and plan the high-level intermediate states. Fig. 10 shows one of the two cases that

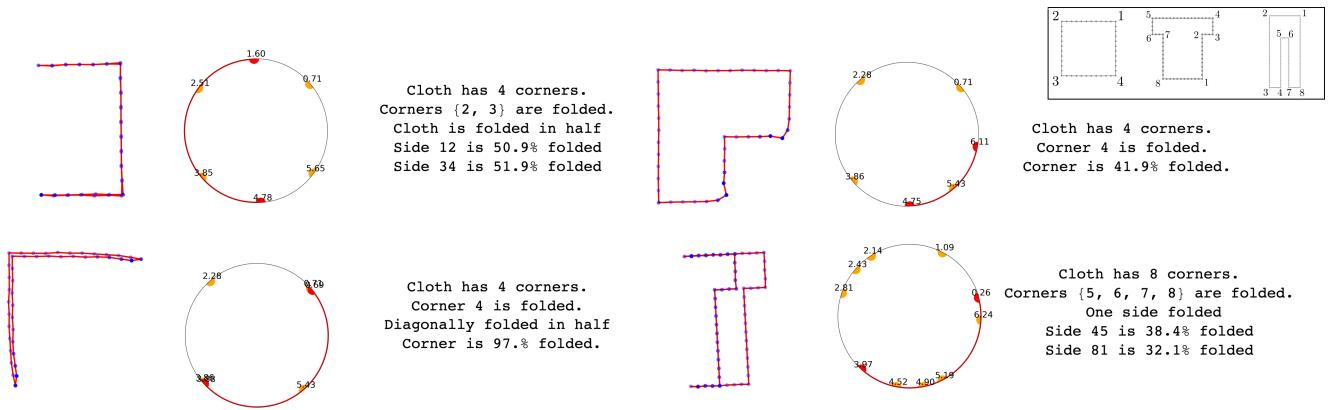


Fig. 9: Different examples of the CloSE representation and the corresponding automatic generated label. More examples can be found in the paper website.

we encounter when we are dealing with one fold. In case 1, both the initial and final configurations are in the same semantic region, i.e., the fold encloses the same corner(s). Then, we can directly manipulate the folded corner(s) to their goal position(s). You can see a case 1 example in the video.

In case 2, the initial and the goal configurations are in different semantic states, so we need to take a multi-step approach to reach the goal. Note that for single fold examples, the unfolded state is the intermediate state.

For low-level planning, that is, deciding where to pick the cloth and where to place it, the CloSE representation proves useful. In the cases where there are more than 2 folded corners that require to be moved, we can pick 2 corners such that the area of the trapezoid formed between such corners and the fold points on the border is maximum. The idea here is to control the maximum possible area of the cloth. Our manipulation actions are shown in Fig. 10b. We also run the trajectories in simulation and these can be seen in the video on the website.

VII. MULTI-FOLDS

We have only discussed the single fold until now. However, the dGLI disk does show interesting patterns even when there is more than one fold. Fig. 11 shows examples of multiple folds - their location on the unfolded cloth and the

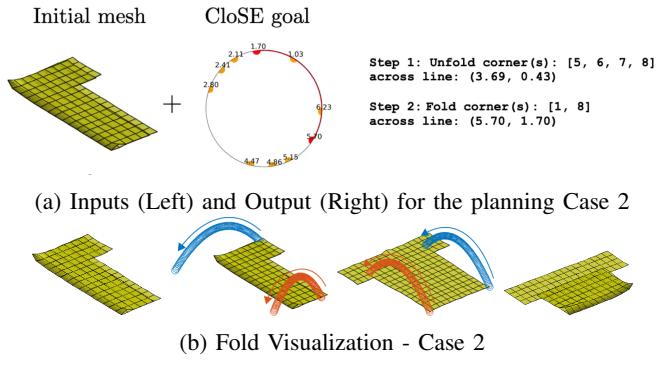


Fig. 10: Planning outputs

dGLI disk fitted with the curve we described earlier. There are evident patterns observed on the dGLI disk such as the fold locations and the side folded. However, these patterns are more intricate and not as simple as those for a single fold, hence we leave the extension of CloSE to more than one fold for future work.

Even without exploring the multi-fold scenarios, our proposed representation, restricted to single folds, is also useful as we can iteratively use it for more than one fold. Our evaluations demonstrate that the dGLI Disk can detect fold lines for subsequent folds even on the very noisy VR folding dataset.

VIII. CONCLUSION

In summary, we discuss a novel geometric representation of clothes states. First, we introduce the dGLI disk, an intermediate representation that is a rearrangement of the dGLI coordinates onto a disk. This representation reveals easy-to-detect geometric features that characterize the cloth folding state independently of the cloth shape. The dGLI disk representation could be directly fed as an extra feature

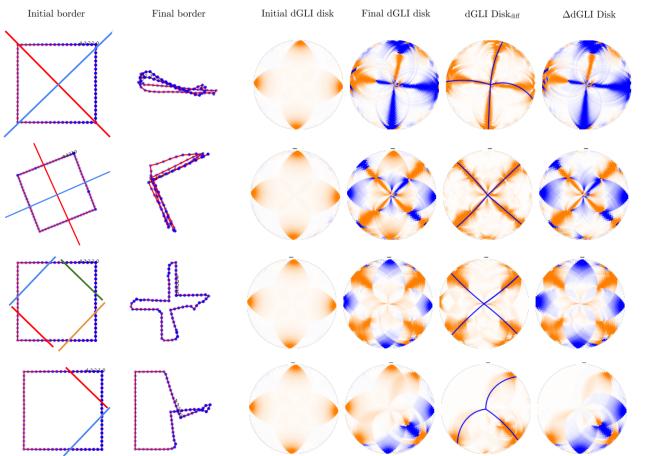


Fig. 11: Patterns in the dGLI disk for multi-fold

of a learning method, providing generality to shape just by construction.

We then abstract the features from our dGLI disk onto a circle. This circular representation, CloSE, is compact and continuous while maintaining the generality for different cloth shapes offered by the dGLI disk. We then empirically show that the fold locations are accurately captured by our geometric representation, which to our knowledge is an unexplored topic even in simulation. Finally, we show two important applications that come naturally from this representation: Semantic labeling and high- and low-level fold planning.

A good cloth representation is the one that allows a robot to reason about cloth states, and plan a sequence of motions to attain a target cloth configuration. Devising a powerful representation for clothes is a very important and difficult problem in robotics and computer graphics. This work is a step in that direction.

REFERENCES

- [1] H. Yin, A. Varava, and D. Kragic, “Modeling, learning, perception, and control methods for deformable object manipulation,” *Science Robotics*, vol. 6, no. 54, p. eabd8803, 2021.
- [2] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, “Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey,” *International Journal of Robotic Research*, vol. 37, no. 7, pp. 688–716, 2018.
- [3] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li, et al., “Challenges and outlook in robotic manipulation of deformable objects,” *IEEE Robotics & Automation Magazine*, vol. 29, no. 3, pp. 67–77, 2022.
- [4] A. Longhini, Y. Wang, I. Garcia-Camacho, D. Blanco-Mulero, M. Molletta, M. Welle, G. Alenyà, H. Yin, Z. Erickson, D. Held, et al., “Unfolding the literature: A review of robotic cloth manipulation,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 8, 2024.
- [5] J. Bednarik, P. Fua, and M. Salzmann, “Learning to reconstruct texture-less deformable surfaces from a single view,” in *Int. Conf. on 3d vision (3DV)*, pp. 606–615, 2018.
- [6] C. Chi and S. Song, “Garmentnets: Category-level pose estimation for garments via canonical space shape completion,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3324–3333, 2021.
- [7] A. Ramisa, G. Alenyà, F. Moreno-Noguer, and C. Torras, “A 3d descriptor to detect task-oriented grasping points in clothing,” *Pattern Recognition*, vol. 60, pp. 936–948, 2016.
- [8] J. Qian, T. Weng, L. Zhang, B. Okorn, and D. Held, “Cloth region segmentation for robust grasp selection,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 9553–9560, 2020.
- [9] M. Lippi, P. Poklukar, M. C. Welle, A. Varava, H. Yin, A. Marino, and D. Kragic, “Enabling visual action planning for object manipulation through latent space roadmap,” *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 57–75, 2022.
- [10] D. Tanaka, S. Arnold, and K. Yamazaki, “Emd net: An encode-manipulate-decode network for cloth manipulation,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1771–1778, 2018.
- [11] F. Coltraro, J. Fontana, J. Amorós, M. Alberich-Carramiñana, J. Borràs, and C. Torras, “A representation of cloth states based on a derivative of the gauss linking integral,” *Applied Mathematics and Computation*, vol. 457, p. 128165, 2023.
- [12] Z. Huang, X. Lin, and D. Held, “Mesh-based dynamics with occlusion reasoning for cloth manipulation,” in *Robotics: Science and Systems (RSS)*, 2022.
- [13] S. Miller, M. Fritz, T. Darrell, and P. Abbeel, “Parametrized shape models for clothing,” in *IEEE International Conference on Robotics and Automation*, pp. 4861–4868, 2011.
- [14] A. Doumanoglou, J. Stria, G. Peleka, I. Mariolis, V. Petrik, A. Karagakos, L. Wagner, V. Hlavac, T.-K. Kim, and S. Malassiotis, “Folding clothes autonomously: A complete pipeline,” *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1461–1478, 2016.
- [15] Y. Li, Y. Yue, D. Xu, E. Grinspun, and P. K. Allen, “Folding deformable objects using predictive simulation and trajectory optimization,” in *IEEE/RSJ Int. Conf. on Intel. Rob. and Sys.*, pp. 6000–6006, 2015.
- [16] R. Hoque, D. Seita, A. Balakrishna, A. Ganapathi, A. K. Tanwani, N. Jamali, K. Yamane, S. Iba, and K. Goldberg, “Visuo-spatial foresight for multi-step, multi-task fabric manipulation,” in *Rob.: Sci. and Sys. (RSS)*, 2020.
- [17] D. Seita, N. Jamali, M. Laskey, A. K. Tanwani, R. Berenstein, P. Baskaran, S. Iba, J. Canny, and K. Goldberg, “Deep transfer learning of pick points on fabric for robot bed-making,” in *International Symposium on Robotics Research (ISRR)*, 2019.
- [18] W. Yan, A. Vangipuram, P. Abbeel, and L. Pinto, “Learning predictive representations for deformable objects using contrastive estimation,” in *Conference on Robot Learning*, pp. 564–574, PMLR, 2021.
- [19] V. Ivan, D. Zarubin, M. Toussaint, T. Komura, and S. Vijayakumar, “Topology-based representations for motion planning and generalization in dynamic environments with interactions,” *The International Journal of Robotics Research*, vol. 32, no. 9–10, pp. 1151–1163, 2013.
- [20] D. Zarubin, V. Ivan, M. Toussaint, T. Komura, and S. Vijayakumar, “Hierarchical motion planning in topological representations,” *Proceedings of Robotics: Science and Systems VIII*, 2012.
- [21] F. T. Pokorny, J. A. Stork, and D. Kragic, “Grasping objects with holes: A topological approach,” in *2013 IEEE international conference on robotics and automation*, pp. 1100–1107, IEEE, 2013.
- [22] J. A. Stork, F. T. Pokorny, and D. Kragic, “Integrated motion and clasp planning with virtual linking,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3007–3014, IEEE, 2013.
- [23] J. A. Stork, F. T. Pokorny, and D. Kragic, “A topology-based object representation for clasping, latching and hooking,” in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 138–145, IEEE, 2013.
- [24] W. Yuan, K. Hang, H. Song, D. Kragic, M. Y. Wang, and J. A. Stork, “Reinforcement learning in topology-based representation for human body movement with whole arm manipulation,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 2153–2160, IEEE, 2019.
- [25] J. Borràs, A. Boix-Granell, S. Foix, and C. Torras, “A virtual reality framework for fast dataset creation applied to cloth manipulation with automatic semantic labelling,” in *IEEE International Conference on Robotics and Automation (ICRA 2023)*, pp. 0210031–0210311, IEEE, 2023.
- [26] W. Wang, G. Li, M. Zamora, and S. Coros, “Trtm: Template-based reconstruction and target-oriented manipulation of crumpled cloths,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12522–12528, IEEE, 2024.
- [27] R. Ren, M. Gurnani Rajesh, J. Sanchez-Riera, A. Lopez-Rodriguez, F. Zhang, Y. Tian, G. Alenyà, A. Agudo, Y. Demiris, K. Mikolajczyk, et al., “Grasp-oriented fine-grained cloth segmentation without real supervision,” in *Proceedings of the 2023 6th International Conference on Machine Vision and Applications*, pp. 147–153, 2023.
- [28] S. L. Ho, *Topology-based character motion synthesis*. PhD thesis, University of Edinburgh, 2011.
- [29] M. Levitt, “Protein folding by restrained energy minimization and molecular dynamics,” *Journal of molecular biology*, vol. 170, no. 3, pp. 723–764, 1983.
- [30] F. Coltraro, J. Amorós, M. Alberich-Carramiñana, and C. Torras, “A novel collision model for inextensible textiles and its experimental validation,” *Applied Mathematical Modelling*, vol. 128, pp. 287–308, 2024.
- [31] H. Xue, W. Xu, J. Zhang, T. Tang, Y. Li, W. Du, R. Ye, and C. Lu, “Garmenttracking: Category-level garment pose tracking,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21233–21242, 2023.
- [32] H. Bertiche, M. Madadi, and S. Escalera, “Cloth3d: clothed 3d humans,” in *European Conference on Computer Vision*, pp. 344–359, Springer, 2020.