Self-supervised Learning of Reconstructing Deformable Linear Objects under Single-Frame Occluded View

Song Wang, Guanghui Shen, Shirui Wu and Dan Wu

***Abstract*— Deformable linear objects (DLOs), such as ropes, cables, and rods, are common in various scenarios, and accurate occlusion reconstruction of them are crucial for effective robotic manipulation. Previous studies for DLO reconstruction either rely on supervised learning, which is limited by the availability of labeled real-world data, or geometric approaches, which fail to capture global features and often struggle with occlusions and complex shapes. This paper presents a novel DLO occlusion reconstruction framework that integrates self-supervised point cloud completion with traditional techniques like clustering, sorting, and fitting to generate ordered key points. A memory module is proposed to enhance the self-supervised training process by consolidating prototype information, while DLO shape constraints are utilized to improve reconstruction accuracy. Experimental results on both synthetic and real-world datasets demonstrate that our method outperforms state-of-the-art algorithms, particularly in scenarios involving complex occlusions and intricate self-intersections.**

I. INTRODUCTION

Deformable linear objects (DLOs) like ropes, cables, and rods are widely used in various industrial scenarios such as automotive, marine, and aerospace for power supply, signal transmission, and hydraulic transport [1]. Currently, DLOs in these industries are still manipulated manually for routing and plugging [2], which hampers productivity and quality assurance. Due to the complex manipulation environment, robot manipulation for these tasks has become a hot research topic, involving modeling [3], planning [4], control [5], and perception [6]. Given the infinite degrees of freedom (DoF) of a DLO, modeling and planning alone are insufficient. Therefore, it is essential to construct an accurate and robust visual perception module for a closed-loop control method.

To handle occlusion and noise during the perception process, it is imperative to execute both segmentation and reconstruction tasks. Segmentation can be achieved using a variety of image processing [7] and data-driven techniques [8]–[10]. While tracking-based methods are widely employed to mitigate the impact of occlusion in reconstruction [11]–[14], an occluded initial state can significantly compromise their performance [15]. So this paper mainly focuses on the problem of DLO occlusion reconstruction, especially reconstructing the DLO's state represented by ordered key points [16] within a single frame.

Currently, there are two main approaches to this problem: learning based [17], [18] and geometric method [19]–[21].

This work was financially supported by the National Natural Science Foundation of China [Grant No.52375019]. S. Wang, G. Shen, S. Wu, D. Wu are with Beijing Key Laboratory of Precision / Ultra-Precision Manufacturing Equipment Control, Department of Mechanical Engineering, Tsinghua University, Beijing, China. Corresponding author: Dan Wu (wud@tsinghua.edu.cn)

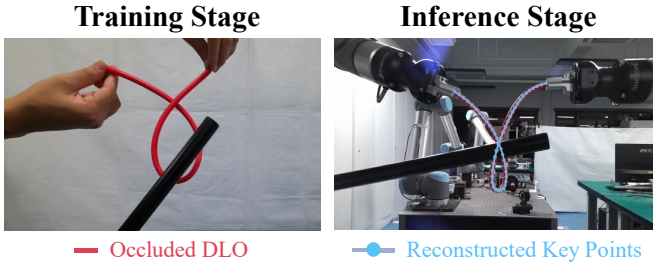

Fig. 1: Illustration of our 3D efficient self-supervised DLO reconstruction algorithm: It simplifies data collection by allowing manual manipulation of both ends of the DLO under an RGB-D camera, enabling the robot to robustly infer the DLO's 3D state, even under severe occlusions.

The learning based method involves directly regressing key points by first creating key point labels and then using supervised training to regress the ordered key points. However, this approach presents two significant limitations:

- Training is limited to virtual environments, as real-world labels are unattainable [22].
- Regressing ordered key points is both challenging and unnecessary, as key points only need to be evenly distributed along the DLO in practical applications, without strict adherence to predefined labels.

On the other hand, geometric methods typically involve extracting key points through 2D skeleton extraction or 3D clustering techniques such as GMM or K-means [23]. These key points are then connected by establishing an ordered sequence, followed by fitting the data using B-splines [20] or Bezier curves [21] to complete the reconstruction. However, these methods only consider local geometric properties and have their own limitations:

- It focuses only on local information around defects, lacking the ability to capture and leverage global features.
- Reconstruction relies purely on geometric data, neglecting material properties and mechanical characteristics.

To address these problems above, we propose a novel DLO occlusion reconstruction framework. It first employs a self-supervised method to complete the DLO's point clouds. To overcome the weak supervisory signals inherent in self-supervised learning, we introduce a memory module that extracts memory across samples to reinforce the supervisory signals. For the complexities introduced by the highly intricate shapes of DLOs, we integrate DLO prior knowledge

Manuscript 964 submitted to 2025 IEEE International Conference on Robotics and Automation (ICRA). Received September 13, 2024.

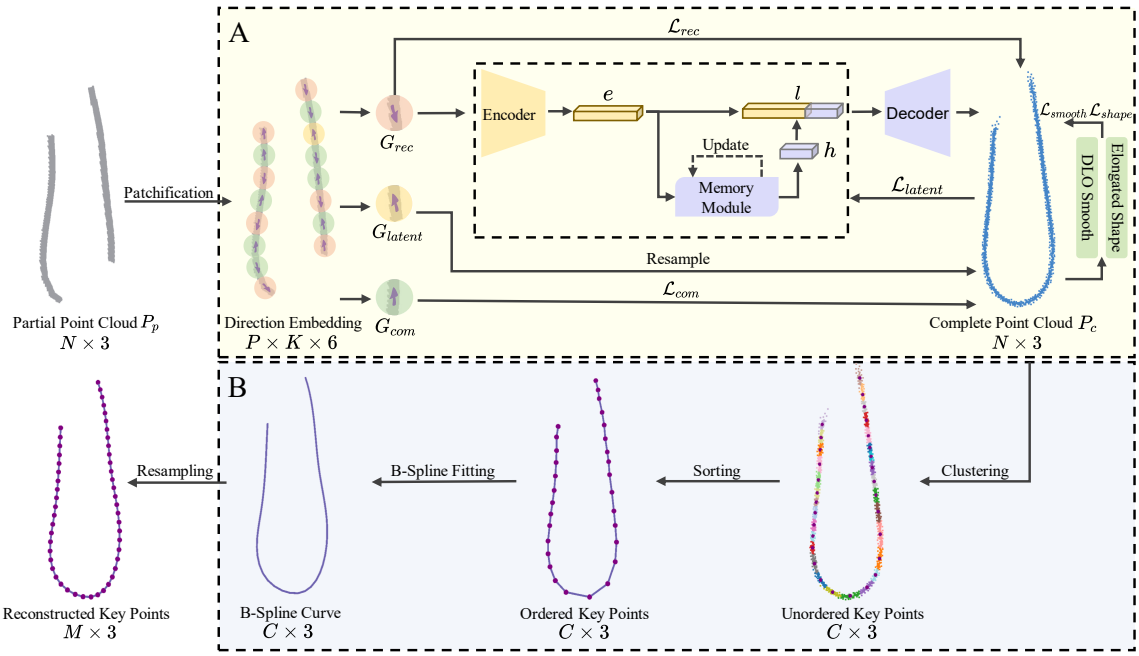

Fig. 2: The framework of our algorithm: A. Self-supervised DLO Point cloud completion, B. Ordered key points generation. After embedding direction information, the partial point cloud P_p is fed into our self-supervised DLO point cloud completion neural network to generate the complete point cloud P_c. To estimate the state of the DLO, P_c undergoes clustering, sorting, B-spline fitting and resampling, ultimately resulting in the reconstructed key points.

constraints, which guide the training process for improved performance. Subsequently, a series of traditional methods (clustering, sorting, fitting and resampling) are applied to generate ordered key points. This framework leverages the feature extraction capabilities of deep learning to capture global characteristics and represent information beyond pure geometry. By adopting a self-supervised learning approach, the framework eliminates the need for labeled data in virtual environments, effectively bridging the Sim2Real gap (see Fig.1). The main contributions of this work are:

- a self-supervised DLO occlusion single-frame reconstruction framework that integrates learning and traditional methods, enabling efficient reconstruction with partial point clouds from an RGB-D camera (see Fig.2).
- a memory module to enhance self-supervised training by consolidating prototype information across samples, improving occlusion estimation and point cloud completion accuracy.
- a DLO shape constraints loss leveraging structural priors, enhancing input information and supervisory signals during training to achieve superior DLO's configuration representations.

Experiments on both synthetic and real-world DLO datasets demonstrate that the proposed method outperforms state-of-the-art algorithms in DLO reconstruction.

II. Method

A. Problem Statement

The problem focused on this paper is the challenge of reconstructing a DLO from a single-frame occluded view. We assume that the point cloud of the DLO has been segmented from the raw full point cloud using RGB image segmentation. For our algorithm, the input consists of the segmented point cloud, which is partial and noisy, and the output is a sequence of ordered key points that accurately represent the DLO's configuration.

B. Self-supervised DLO Point Cloud Completion

We employ Partial2Complete (P2C) [24] as the foundational framework for self-supervised point cloud completion. This framework divides the input point cloud P_p into three parts: G_{rec}, G_{com} and G_{latent}. Initially, G_{rec} is fed into the network to predict the complete point cloud P_c. By leveraging the proposed Region-Aware Chamfer Distance (RCD), the losses are respectively computed with G_{rec} and G_{com} to obtain $\mathcal{L}_{rec}$ and $\mathcal{L}_{com}$. Subsequently, G_{latent} is utilized to resample the predicted point cloud and perform inference, thereby providing supervision for the latent representations using $\mathcal{L}_{latent}$.

Although P2C enhances the supervision signals in the self-supervised training process by utilizing RCD to restrict loss calculation to local regions, the missing point clouds in the input still lack supervisory signals, leaving the issue of weak supervision still unresolved. Additionally, the highly intricate shapes of DLOs further result in the suboptimal performance of P2C in this problem.

1) Memory module: To enhance the supervision signals, we design a cross-sample memory module to extract intra-class prototypes. For an individual sample, the supervision signals compromised by missing point clouds are challenging

Manuscript 964 submitted to 2025 IEEE International Conference on Robotics and Automation (ICRA). Received September 13, 2024.

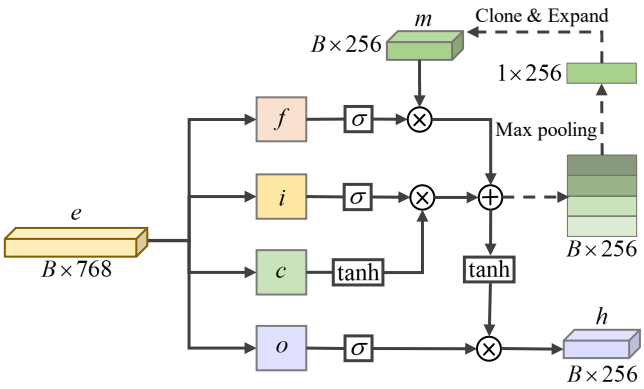

Fig. 3: Memory module: Using the gating mechanism and max pooling to manage memory. Update memory with max pooing when training and use the stored memory to infer.

to restore. However, for a batch of samples of the same class, the class characteristics can be distilled into intra-class prototypes to supplement the missing information. Inspired by LSTM [25], we introduce a gating mechanism to manage memory and employ max pooling to refine memory across samples.

As shown in Fig. 3, we introduce input gate i, forget gate f , output gate o and candidate gate c to control the encoded feature e and memory m, i determines what new information we're going to store in the memory, f is responsible for deciding what information to discard from the memory. c is used for refining the candidate memory information and o provides the activation to the final output of this module. Different from the classic LSTM's memory updating mechanism, our memory is updated by max pooling. With the help of max pooling which is irrelevant to the input order within batch when training, our model can refine the most significant feature to shape the intra-class prototypes, this process can be formulated as:

$$\begin{aligned} m_{new} &= \max\text{pooling}(i \cdot c + f \cdot m_{prev}) \\ h &= o \cdot \tanh(i \cdot c + f \cdot m_{prev}) \end{aligned} \tag{1}$$

where i, f, o, and c represent the outputs of the respective gates, respectively. m_{prev} is the memory state from the previous batch, m_{new} is the updated memory and h is the output state.

2) DLO shape constraints:

a) Direction embedding: Despite DLO having diverse configurations, it consistently maintains smooth and continuous characteristics. Even when occluded in an RGB-D camera views, the visible point clouds, retains these smooth and continuous properties locally. To better leverage this characteristic of DLO, we design a direction embedding module. This module extends the input dimensions with tangential information during model input, enabling the model to better understand the correlations in completing and reconstructing the point cloud.

Similar to P2C's patchification, We use farthest point sampling (FPS) [26] to sample M points as initial patch centers $\tilde{c} = \{\tilde{c}_i\}_{i=1}^{M}$, Then, we gather the K-nearest neighbors of each initial center point to obtain a patch $g_i = \{p | p \in \mathcal{N}_k^{P_p}(\tilde{c}_i)\}$ where $\mathcal{N}_k^{P_p}(\tilde{c}_i)$ denotes the set of K-nearest neighbors for $\tilde{c}_i$ in P_p. To obtain the representative direction, we recalculate the centers $C = \{c_i\}_{i=1}^{M}$ using all the points within each patch. After that, we decompose the eigenvalue of the covariance matrix cov_i between c_i and p_{ij}, defined as:

$$cov_i = \sum_{j=1}^{k} \frac{(p_{ij} - c_i) \cdot (p_{ij} - c_i)^T}{k}, c_i = \frac{1}{k} \sum_{j=1}^{k} p_{ij} \tag{2}$$

where the eigenvector corresponding to the largest eigenvalue of cov_i is the normalized tangential vector t_i. For each point p_{ij} in g_i, we embed t_i into its three-dimensional coordinates to achieve direction embedding.

To further regularize the completion of DLO, we introduce DLO smooth loss and elongated shape loss. We use a similar method of patchification as in the direction embedding module for complete point cloud P_c, so we can obtain $\hat{g}_i$, $\hat{c}_i$, $\hat{t}_i$ and eigenvalue λ_1, λ_2 and λ_3 ($\lambda_1 \geq \lambda_2 \geq \lambda_3$) where $i \in [1, M]$.

b) DLO smooth loss: To improve the smoothness of the DLO shape, we gather the 2-nearest neighbors of each center point $\hat{c}_i$ and sort them locally $\hat{c}_{s0} \rightarrow \hat{c}_{s1} \rightarrow \hat{c}_{s2}$. Then we design DLO smooth loss by the similarity of tangential directions between adjacent patches, which can be formulated as:

$$\mathcal{L}_{smooth} = \sum_{s=1}^{M} \frac{1 - (|t_{s0} \cdot t_{s1}| + |t_{s1} \cdot t_{s2}|)/2}{M} \tag{3}$$

where $s \in [1, M]$, $|t_{s0} \cdot t_{s1}|$, $|t_{s1} \cdot t_{s2}|$ represent the cosine values of the angles θ_{01}, θ_{12} between two adjacent tangential vectors.

c) Elongated shape loss: In contrast to point cloud completion tasks for other objects, DLO has a greater aspect ratio. Employing the elongated shape as a constraint aids the model in more effectively completing point cloud completion tasks. We use the eigenvalues of each patch to design our elongated shape loss, formulated as:

$$L_{shape} = -\sum_{i=1}^{M} \frac{1}{M} \cdot \frac{\lambda_1 - (\lambda_2 + \lambda_3)/2}{\lambda_1 + \lambda_2 + \lambda_3 + \varepsilon} \tag{4}$$

where $\lambda_1 - (\lambda_2 + \lambda_3)$ represents the difference between the primary direction and the sum of the other directions. $\varepsilon = 10^{-9}$, which is used to prevent numerical problem. Now we have the overall loss defined as:

$$\begin{aligned} \mathcal{L} =& \alpha_r \mathcal{L}_{rec} + \alpha_c \mathcal{L}_{com} + \alpha_l \mathcal{L}_{latent} \\ &+ \alpha_{sm} \mathcal{L}_{smooth} + \alpha_{sh} \mathcal{L}_{shape} \end{aligned} \tag{5}$$

where α_r, α_c, α_l, α_{sm} and α_{sh} are weighting parameters.

C. Ordered Key Points Generation

After DLO point cloud completion, we need to generate ordered key points using unordered point cloud to get the state of the DLO. The main process is divided into four steps: clustering, sorting, fitting and resampling.

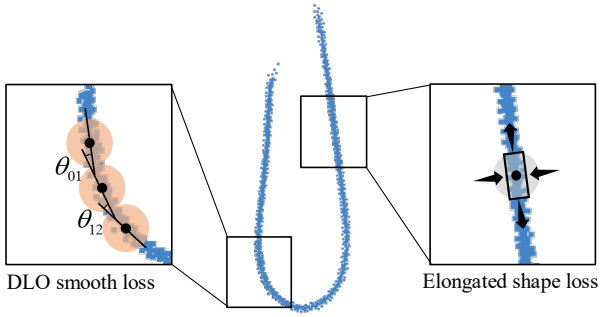

Fig. 4: **DLO smooth loss:** the similarity of tangential directions between adjacent patches, a smaller DLO smooth loss ensures a smoother and more continuous DLO. **Elongated shape loss:** the difference between the primary direction and the sum of the other directions, inspired by the high aspect ratio character of DLO. The larger this difference, the greater the aspect ratio, indicating a more elongated DLO shape.

1) Clustering: To fully leverage the known conditions and improve prediction accuracy, we first align and merge the P_c and P_p by replacing the unoccluded portions of the point cloud with P_c before clustering. Subsequently, we perform clustering on the merged point cloud using a Gaussian Mixture Model (GMM) to generate evenly distributed centroids, which represent the shape of the DLO as unordered key points.

2) Sorting: To determine the connections between the unordered key points and arrange them in order, inspired by [20], the initial state has an empty sorted set and an unsorted set containing all unordered key points. We randomly select a starting point as the endpoint of the sorted segment, then greedily find the point that minimizes the connection cost, adding it to the sorted set and using it as the new endpoint for further calculations, until the unsorted set is empty. The cost of adding the i-th point to the connection set is defined as follows:

$$\begin{aligned} J_i &= min\{J_{si}, J_{ei}\} \\ J_{si} &= m \cdot \mathrm{dis}(s,i) + (1-m) \cdot \mathrm{ori}(s,i) \\ J_{ei} &= m \cdot \mathrm{dis}(e,i) + (1-m) \cdot \mathrm{ori}(e,i) \end{aligned} \tag{6}$$

where J_{si}, J_{ei} represent the costs associated with connecting the i-th point to the start and end points of the sorted set, respectively. $\mathrm{dis}(i,j)$ denotes the Euclidean distance between points i and j, while $\mathrm{ori}(i,j)$ refers the cosine of the angle between the tangential vectors of points i and j. m signifies the weight factor, typically set to 0.8. Finally, the ordered key points are determined based on the connection relationships.

3) Fitting and resampling: The ordered key points are used as control points for fitting smooth B-splines. Afterward, we uniformly resample a specified number of ordered key points on the B-spline for representation and evaluation. The B-spline fitting and ordered key points resampling are implemented using the Python packages *scipy.interpolate.splprep* and *scipy.interpolate.splev*.

III. Experiments

A. Simulation Experiments

1) Data collection and model training: Given the challenge of obtaining ground-truth data for occluded DLO in real-world scenarios, to thoroughly evaluate algorithm performance, we use Issac Sim [27] as the simulator to generate a diverse set of DLOs, each with 50 key points. To enhance data collection efficiency and better simulate occlusion scenarios, we use the Replicator plugin [28] for auxiliary collection. Under this setup, the DLO with additional semantic label has their endpoints moving randomly within two overlapping hemispheres with partial intersection [16], occlusion cube moves concurrently within the activity range of the DLO to simulate plausible occlusions and the camera captures point clouds of the visible portions of the DLO. Additionally, to simulate defects arising from suboptimal camera imaging quality, we introduce noise into the output point clouds. We then generate a total of 7,000 DLO data samples, partitioning them into training and validation sets with a ratio of 0.8 to 0.2. Meanwhile, 1,000 DLO samples are produced for the test set. We sample $N = 2048$ points from the input point clouds and train the model for 300 epochs on a GeForce RTX 4080 Super GPU. For the loss functions, we set the weights for $\mathcal{L}_{rec}$, $\mathcal{L}_{con}$, $\mathcal{L}_{latent}$,$\mathcal{L}_{smooth}$, $\mathcal{L}_{shape}$ to 1, 1000, 0.1, 0.005 and 0.005, respectively. For further details on the training implementation, please refer to our website[1].

2) Evaluation Metrics: To facilitate a more accurate comparison with similar reconstruction algorithms and to evaluate the overall performance of the algorithm in manipulation tasks, we adopt five metrics similar to [21]:

- D_{1bi} measures the overall similarity.
- D_1 measures the degree of fitting, although the reconstructed DLO may be incomplete.
- D_2 estimates the worst-case distance between the reconstructed points and the target position during DLO grasping or placement.
- D_3 estimates the percentage of poorly reconstructed points by using a distance threshold.
- D_4 measures the physical smoothness of the DLO.

which can be formulated as:

$$\begin{aligned} D_{1bi}(X,Y) &= \mathrm{CD}(X,Y) \\ D_1(X,Y) &= \frac{1}{|X|}\sum_{x\in X}\min_{y\in Y} L_2(x,y)^2 \\ D_2(X,Y) &= \max_{x\in X}\min_{y\in Y} L_2(x,y) \\ D_3(X,Y \mid T) &= \frac{\sum_{x\in X}\mathbf{1}\left(\min_{y\in Y} L_2(x,y) > T\right)}{|X|} \\ D_4(X) &= \sum_{i=1}^{n-2} \mathrm{angle}(\Delta X_i, \Delta X_{i+1})^2 \end{aligned} \tag{7}$$

where X are the 3D reconstructed key point set, Y are the ground-truth key point set, $\mathrm{CD}(\cdot)$ denotes the Chamfer distance, $|\cdot|$ denotes the number of points in the set, $L_2(\cdot)$

[1]https://mp2cdlo.github.io/MP2CDLO/

Manuscript 964 submitted to 2025 IEEE International Conference on Robotics and Automation (ICRA). Received September 13, 2024.

TABLE I: Five Metrics and Success Rate on the Synthetic DLO Datasets

ID	Algorithms	Success Rate(%) ↑	D_{1bi}(mm) ↓	D_1(mm) ↓	D_2(mm) ↓	D_3 [2]↓	D_4 ↓
1	Lv	85	23.92	10.38	23.81	.276	5.20
2	DLOFTBs	95	17.35	7.76	19.54	.085	5.47
3	Sun[1]	87	20.77	7.34	20.09	.065	4.96
4	P2C	100	13.02	6.44	16.17	.028	4.90
	MP2C	100	**12.79**	**6.22**	12.58	.026	4.81
	MP2CDLO	**100**	12.89	6.24	**11.68**	**.025**	**4.66**

[1] The Rec method is evaluated here.
[2] D_3 has a threshold of 10 mm.

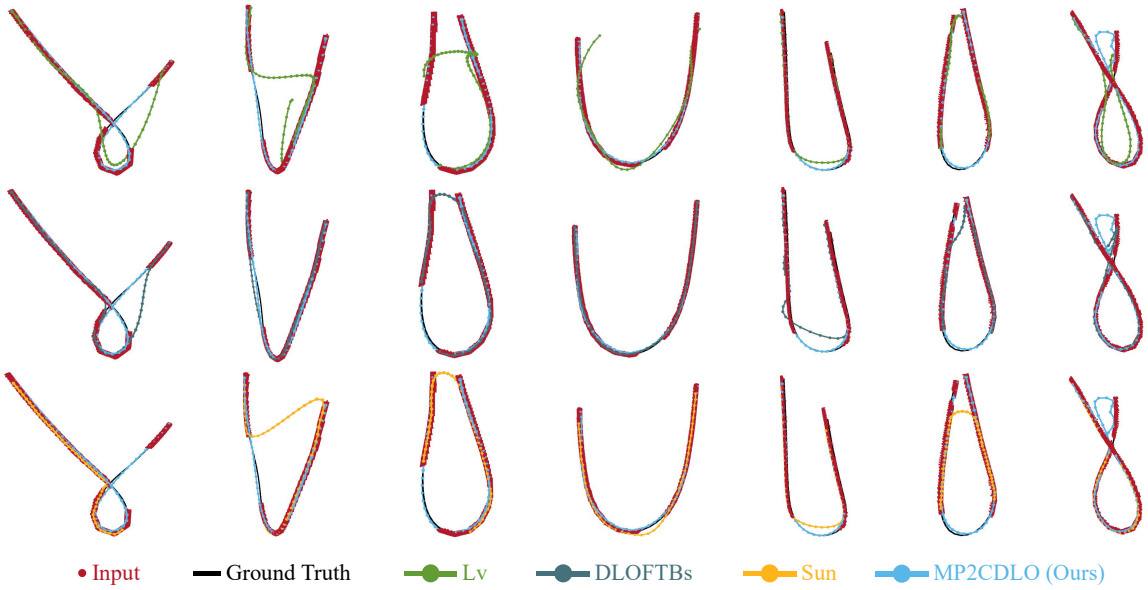

Fig. 5: MP2CDLO reconstruction visualization compared with other advanced algorithms on synthetic DLO datasets. We illustrate scenarios of self-intersections(1st and last columns), extensive occlusions(2nd and 5th columns), and close proximity of the ends(3rd and 6th columns). In these extreme cases, MP2CDLO generally performs well, while other algorithms exhibit significant deviations from the ground-truth and even incorrect connections.

denotes the $L2$ norm, T is the distance threshold, $\Delta X_{i+1} = X_{i+2} - X_{i+1}$ and $\text{angle}(A, B) = \arccos(A \cdot B)/(\|A\| \cdot \|B\|)$. To enhance intuitiveness, we report D_{1bi} and D_1 using $L2$ norm instead of squared $L2$ norm.

Additionally, to mitigate the impact of significant increases in the distance metric caused by reconstruction failures, we also use SuccessRate to measure the number of cases the DLO is correctly reconstructed when the metric $D_{1bi} < 50mm$.

3) Algorithm comparison and ablation study: We quantitatively evaluate the performance of our proposed algorithm using five metrics mentioned above on our synthetic datasets, comparing it with other advanced algorithms: Lv [18], DLOFTBs [20] and Sun [21], where Lv's method is supervised learning based method and the others are geometric model method. For the ablation study, we evaluate three versions of the proposed method: 1) **P2C**: the original P2C used for point cloud completion, combined with our ordered key point generation method for output. 2) **MP2C**: P2C enhanced with our memory module, which strengthens the supervision signals. 3) **MP2CDLO**: MP2C further augmented with DLO shape constraints, representing our final algorithm.

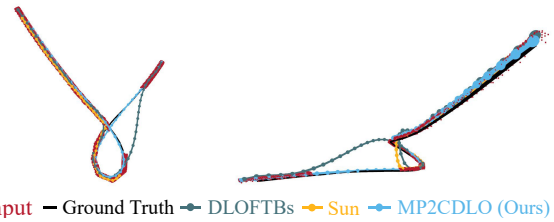

Fig. 6: Illustration of a viewpoint occlusion self-intersection scenario: the RGB-D camera viewpoint (left) shows a severe intersection, while the top-down view (right) reveals clearer connection relationships.

As shown in Table. I, our proposed algorithms outperform the others, while Lv's method does not demonstrate strong performance. This is because Lv's method relies on training with labeled datasets and has limited generalization capability, making high-performance transfer difficult

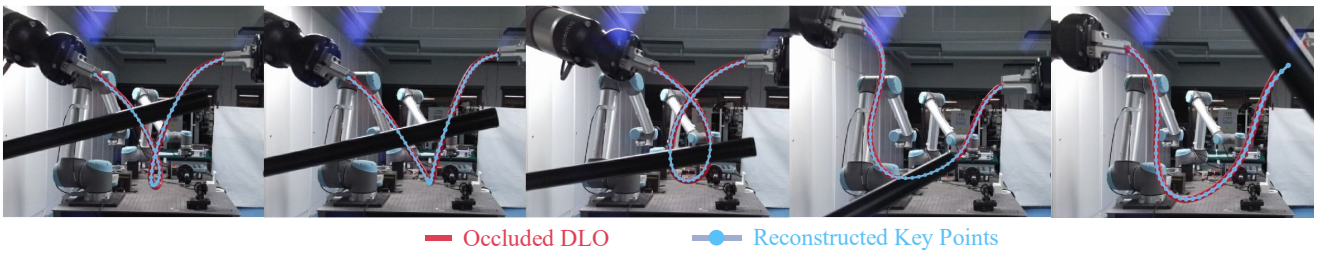

Fig. 7: MP2CDLO reconstruction visualization in a real-world experiment: The DLO is manipulated by two robots, with occasional occlusions. We illustrate scenarios of viewpoint occlusion self-intersection (1st column), spatial proximity self-intersection (3rd column), extensive occlusions (2nd and 4th columns) and end-point occlusion (last column).

to achieve. The other two algorithms, while not requiring dataset training, are limited by their reliance solely on geometric properties, which restricts their ability to achieve high-quality reconstructions. In contrast, all three versions of our proposed method demonstrate high $\mathrm{Success\,Rate}$ and good performance on other metrics, which means the framework that integrates learning and traditional methods works well and effectively mitigates the occurrence of severe reconstruction failures.

Fig. 5 illustrates that MP2CDLO performs well in most cases, but other algorithms tend to produce incorrect connections under these conditions. MP2CDLO can successfully complete the reconstruction by utilizing length and other physical properties, even in the presence of severe occlusions and close proximity of the ends. However, when the ends are in close proximity and self-intersections occur simultaneously, the model struggles to generate high-resolution point clouds, resulting in incorrect connections. (see last column). Furthermore, since DLOFTBs and Sun's method establish connection relationships based on 2D images, the lack of depth information results in poor performance when dealing with self-intersections caused by viewpoint occlusions (see Fig. 6).

In the ablation study, MP2C performs better than P2C across all metrics, especially on D_2, because the memory stores the intra-class prototypes which helps prevent severe deviations from the ground-truth. Although MP2CDLO shows slight variations in D_{1bi} and D_1 compared to MP2C, the DLO shape constraints lead to improvements in both D_2 and D_4, which means smoother and more realistic reconstructed point clouds.

B. *Real-world Experiments*

We select a new DLO with a length of 0.5m to evaluate the pipeline of our reconstruction algorithm. Since our algorithm does not rely on data labels, we can easily fine-tune our completion model on new prediction targets to achieve better inference performance.

The real-world experiment consists of two stages: the training stage and the inference stage. During the training stage, we randomly move the two ends of the DLO, grasped by hand, within the view of an RGB-D camera to collect real-world data. We then fine-tune our pre-trained model, which is trained on our synthetic dataset without any labels. In the inference stage, the fine-tuned model can be transferred to robot manipulation tasks directly.

The input DLO point clouds are captured by Azure Kinect DK and segmented from background via color thresholding in HSV space. During the training phase, we collect 3 minutes of data at 30 FPS with the Azure Kinect DK. After downsampling and shuffling the data, we use 1,800 frames to fine-tune our model, which takes 15 minutes for 300 epochs. We then test this model in our robot manipulation environments, where two UR5e robots manipulate a DLO. On average, MP2CDLO takes 90ms in total, with clustering taking 40ms and our completion model taking less than 30ms on the CPU (Intel i7-13700KF) and GPU (GeForce RTX 4080 Super). Since there is no ground-truth for real-world data, we reproject the reconstructed key points onto the raw RGB images to evaluate the algorithm's performance.

Fig. 7 shows the inference results of MP2CDLO, even when self-intersections (caused by spatial proximity rather than mere viewpoint occlusion) and occlusions by obstacles occur simultaneously, MP2CDLO effectively completes the reconstruction and maintains accurate connections. Furthermore, in cases of end-point occlusions, MP2CDLO is also capable of achieving partial completion.

IV. CONCLUSIONS

In this paper, we propose MP2CDLO, a novel framework for DLO occlusion reconstruction that includes a self-supervised point cloud completion model and an ordered key point generation method. Enhanced with memory module and augmented with DLO shape constraints, MP2CDLO addresses the issue of weak supervisory signals in self-supervised training and produces smoother, more realistic DLO reconstructions. By leveraging deep learning's feature extraction capabilities, it captures more than just geometric information and bridges the Sim2Real gap through self-supervised learning. Both simulation and real-world experiments demonstrate that MP2CDLO can handle challenging scenarios such as viewpoint occlusion self-intersections and spatial proximity self-intersections. It outperforms other advanced algorithms, achieving state-of-the-art performance in the occluded DLO reconstruction task. Future work will focuses on developing a higher-quality completion model, improving the algorithm's runtime efficiency, and enabling practical industrial deployment.

Manuscript 964 submitted to 2025 IEEE International Conference on Robotics and Automation (ICRA). Received September 13, 2024.

REFERENCES

[1] 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.

[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," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.

[3] S. Wu, J. Zhang, and D. Wu, "Equilibrium manipulation planning for a soft elastic rod considering an external distributed force and intrinsic curvature," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 442–11 449, 2022.

[4] A. Sintov, S. Macenski, A. Borum, and T. Bretl, "Motion planning for dual-arm manipulation of elastic rods," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6065–6072, 2020.

[5] M. Yu, K. Lv, C. Wang, M. Tomizuka, and X. Li, "A coarse-to-fine framework for dual-arm manipulation of deformable linear objects with whole-body obstacle avoidance," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 153–10 159.

[6] 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.

[7] D. De Gregorio, G. Palli, and L. Di Stefano, "Let's take a walk on superpixels graphs: Deformable linear objects segmentation and model estimation," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 662–677.

[8] A. Caporali, R. Zanella, D. De Greogrio, and G. Palli, "Ariadne+: Deep learning–based augmented framework for the instance segmentation of wires," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 8607–8617, 2022.

[9] A. Caporali, K. Galassi, R. Zanella, and G. Palli, "Fastdlo: Fast deformable linear objects instance segmentation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9075–9082, 2022.

[10] A. Caporali, K. Galassi, B. L. Žagar, R. Zanella, G. Palli, and A. C. Knoll, "Rt-dlo: Real-time deformable linear objects instance segmentation," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 11, pp. 11 333–11 342, 2023.

[11] T. Tang, C. Wang, and M. Tomizuka, "A framework for manipulating deformable linear objects by coherent point drift," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3426–3433, 2018.

[12] C. Chi and D. Berenson, "Occlusion-robust deformable object tracking without physics simulation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6443–6450.

[13] Y. Wang, D. McConachie, and D. Berenson, "Tracking partially-occluded deformable objects while enforcing geometric constraints," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 199–14 205.

[14] J. Xiang, H. Dinkel, H. Zhao, N. Gao, B. Coltin, T. Smith, and T. Bretl, "Trackdlo: Tracking deformable linear objects under occlusion with motion coherence," *IEEE Robotics and Automation Letters*, 2023.

[15] T. Tang and M. Tomizuka, "Track deformable objects from point clouds with structure preserved registration," *The International Journal of Robotics Research*, vol. 41, no. 6, pp. 599–614, 2022.

[16] M. Yu, K. Lv, H. Zhong, S. Song, and X. Li, "Global model learning for large deformation control of elastic deformable linear objects: An efficient and adaptive approach," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 417–436, 2022.

[17] S. Huo, A. Duan, C. Li, P. Zhou, W. Ma, H. Wang, and D. Navarro-Alarcon, "Keypoint-based planar bimanual shaping of deformable linear objects under environmental constraints with hierarchical action framework," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5222–5229, 2022.

[18] K. Lv, M. Yu, Y. Pu, X. Jiang, G. Huang, and X. Li, "Learning to estimate 3-d states of deformable linear objects from single-frame occluded point clouds," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7119–7125.

[19] A. Keipour, M. Bandari, and S. Schaal, "Deformable one-dimensional object detection for routing and manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4329–4336, 2022.

[20] P. Kicki, A. Szymko, and K. Walas, "Dloftbs–fast tracking of deformable linear objects with b-splines," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7104–7110.

[21] S. Zhaole, H. Zhou, L. Nanbo, L. Chen, J. Zhu, and R. B. Fisher, "A robust deformable linear object perception pipeline in 3d: From segmentation to reconstruction," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 843–850, 2023.

[22] M. Yan, Y. Zhu, N. Jin, and J. Bohg, "Self-supervised learning of state estimation for manipulating deformable linear objects," *IEEE robotics and automation letters*, vol. 5, no. 2, pp. 2372–2379, 2020.

[23] P. Zhou, P. Zheng, J. Qi, C. Li, H.-Y. Lee, A. Duan, L. Lu, Z. Li, L. Hu, and D. Navarro-Alarcon, "Reactive human–robot collaborative manipulation of deformable linear objects using a new topological latent control model," *Robotics and Computer-Integrated Manufacturing*, vol. 88, p. 102727, 2024.

[24] R. Cui, S. Qiu, S. Anwar, J. Liu, C. Xing, J. Zhang, and N. Barnes, "P2c: Self-supervised point cloud completion from single partial clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 14 351–14 360.

[25] S. Hochreiter, "Long short-term memory," *Neural Computation MIT-Press*, 1997.

[26] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[27] NVIDIA, "NVIDIA Isaac Sim," 2024. [Online]. Available: https://developer.nvidia.com/isaac-sim

[28] NVIDIA, "Omniverse Replicator," 2024. [Online]. Available: https://docs.omniverse.nvidia.com/extensions/latest/ext$_{r}eplicator.html$