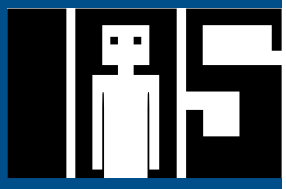
# Learning-based 3D State Estimation of Deformable Linear Objects without Accurate Initialisation

**Lernbasierte 3D-Zustandsschätzung deformierbarer linearer Objekte ohne genaue Initialisierung**

Master thesis by Jiahong Xue
Date of submission: July 3, 2026

1. Review: M.Sc. Berk Gueler
2. Review: Prof. Dr. Jan Peters
Darmstadt

TECHNISCHE
UNIVERSITÄT
DARMSTADT



## Erklärung zur Abschlussarbeit gemäß §22 Abs. 7 und §23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Jiahong Xue, die vorliegende Masterarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß §23 Abs. 7 APB überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 3. Juli 2026

Jiahong Xue

# Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, M.Sc. Berk Gueler, for his continuous guidance and support throughout this thesis. He encouraged me both when the experiments succeeded and, more importantly, when they did not. During difficult stages, he patiently discussed possible causes, alternative approaches, and new research directions with me. I am particularly grateful for the understanding he showed towards my personal circumstances, while still encouraging me to maintain steady progress. His patience, kindness, and constructive guidance contributed greatly to the completion of this work.

I would also like to thank Prof. Dr. Jan Peters and the entire Intelligent Autonomous Systems group for providing the opportunity, research environment, and resources required for this thesis.

Finally, I am deeply grateful to my parents, my family, my girlfriend, and my friends for their continued encouragement and support. Their trust and companionship gave me the opportunity and motivation to pursue and complete my master's degree.

# Abstract

Reliable manipulation of deformable linear objects (DLOs) in tasks such as knot untangling requires an estimate of the complete ordered three-dimensional centerline, including segments that are occluded or involved in self-intersections. Temporal DLO tracking methods based on state propagation, including TrackDLO, require an initial state from which subsequent estimates are updated. After initialisation errors or tracking loss, recovery therefore requires a separate reinitialisation mechanism.

This thesis presents a single-frame estimator that reconstructs an ordered 64-particle DLO centerline from one RGB-D observation without using a previous state. The current observation is represented by an unordered set of visible 3D points and a coarse ordered prior obtained by lifting a two-dimensional path into 3D. These geometric conditions guide a velocity model trained with flow matching to generate the complete centerline. An observation-based plausibility gate rejects inconsistent predictions and returns the coarse prior as fallback.

The method is evaluated against the initialisation procedure used in TrackDLO (hereafter TrackDLO-init) on 59 synthetic validation sequences. TrackDLO-init produces 31 valid initialisations, whereas the proposed method returns an estimate for all 59 sequences. On the 31 paired valid cases, the proposed method reduces the mean index error from $5.911$ cm to $2.061$ cm and the discrete Fréchet distance from $11.365$ cm to $4.599$ cm. Ablation studies show that learned refinement reduces the mean index error relative to the coarse prior, while velocity prediction benefits more clearly from iterative sampling than direct-state prediction. Evaluation on real TrackDLO failure-case recordings further reveals a substantial simulation-to-reality gap in the learned component. Real-domain adaptation and more reliable prior construction therefore remain important directions for future work.

# Zusammenfassung

Die zuverlässige Manipulation deformierbarer linearer Objekte (DLOs), beispielsweise beim Entwirren von Knoten, erfordert eine Schätzung der vollständigen geordneten dreidimensionalen Mittellinie. Dies schließt auch Abschnitte ein, die verdeckt oder an Selbstüberschneidungen beteiligt sind. Zeitliche DLO-Trackingverfahren, die auf der Fortschreibung des Zustands basieren, darunter TrackDLO, benötigen einen Anfangszustand, von dem aus nachfolgende Schätzungen aktualisiert werden. Nach Fehlern bei der Initialisierung oder einem Verlust der Verfolgung ist daher ein separater Mechanismus zur Reinitialisierung erforderlich.

Diese Arbeit stellt einen Einzelbildschätzer vor, der aus einer RGB-D-Beobachtung ohne vorherigen Zustand eine geordnete, aus 64 Partikeln bestehende DLO-Mittellinie rekonstruiert. Die aktuelle Beobachtung wird durch eine ungeordnete Menge sichtbarer 3D-Punkte und einen groben geordneten Prior dargestellt, der durch die Überführung eines zweidimensionalen Pfades in den 3D-Raum entsteht. Diese geometrischen Bedingungen steuern ein mit Flow Matching trainiertes Geschwindigkeitsmodell zur Erzeugung der vollständigen Mittellinie. Ein beobachtungsbasiertes Plausibilitätskriterium verwirft inkonsistente Vorhersagen und gibt stattdessen den groben Prior zurück.

Die Methode wird anhand von 59 synthetischen Validierungssequenzen mit dem in TrackDLO verwendeten Initialisierungsverfahren (im Folgenden TrackDLO-init) verglichen. TrackDLO-init erzeugt 31 gültige Initialisierungen, während die vorgeschlagene Methode für alle 59 Sequenzen eine Schätzung liefert. Für die 31 gültigen Vergleichsfälle sinkt der mittlere Indexfehler von $5{,}911$ cm auf $2{,}061$ cm und die diskrete Fréchet-Distanz von $11{,}365$ cm auf $4{,}599$ cm. Die Ablationsstudien zeigen, dass die gelernte Verfeinerung den mittleren Indexfehler gegenüber dem groben Prior reduziert und dass die Geschwindigkeitsvorhersage stärker vom iterativen Sampling profitiert als die direkte Zustandsvorhersage. Die reale Auswertung zeigt zudem eine deutliche Lücke zwischen Simulation und Realität im gelernten Modellanteil. Domänenanpassung und eine robustere Prior-Konstruktion bleiben daher wichtige Aufgaben.

# Contents

# Figures and Tables

## List of Figures



## List of Tables

# Abbreviations

| Notation | Description |
|---|---|
| CDCPD | constrained deformable coherent point drift |
| CPD | coherent point drift |
| DLO | deformable linear object |
| DPS | diffusion posterior sampling |
| GPU | graphics processing unit |
| HSV | hue–saturation–value |
| ODE | ordinary differential equation |
| PCN | point completion network |
| RGB | red–green–blue |
| RGB-D | red–green–blue and depth |
| RQ | research question |
| SAM | Segment Anything Model |

# 1 Introduction

Ropes, cables, and wires are common examples of deformable linear objects (DLOs). Robots manipulate DLOs in tasks such as knot untangling, where self-crossings and self-occlusion make perception difficult [1]. Detecting the object region in an image is not sufficient for this task. A robot must also know the object's 3D shape [2], [3]. It must identify the endpoints and preserve the connectivity along the object [2], [4]. This thesis focuses on estimating this state for a single DLO from an RGB-D observation.

Unlike the pose of a rigid object, the state of a DLO is high-dimensional [3], [5]. The target state in this thesis is a complete ordered 3D centerline represented by a sequence of points from one endpoint to the other. The order is important because it preserves adjacency and local correspondence along the object. It also identifies the endpoints and describes how the centerline is connected. An unordered set of points does not provide this information. However, estimating such an ordered state from RGB-D observations is not straightforward because the available observation does not directly provide the complete centerline [3].

## 1.1 The Observation-to-State Gap under Self-Occlusion

An RGB-D observation and a segmentation mask provide geometry for the visible parts of a deformable linear object. The mask identifies the object pixels, while the depth map provides 3D measurements of the visible surface. Under occlusion, the point cloud can be incomplete and does not provide the ordered node sequence required by the target state representation [2], [3].

Crossings introduce both visibility and ordering ambiguities. At a crossing, two parts that are far apart along the object can project to the same image region. The front part hides

the back part, leaving the hidden centerline without a direct depth measurement [2]. A 2D skeleton can also contain several possible paths through the intersection. For example, mBEST uses an explicit path-selection criterion to obtain an ordered pixel sequence at crossings [4]. Fitting a centerline only to visible 3D points therefore cannot recover the complete state in regions without direct observations [3].

Task-oriented methods such as AssistDLO reconstruct observable 3D traces for teleoperation assistance, but do not target the complete fixed-cardinality ordered centerline considered in this thesis [5].

A coarse ordered prior provides complementary structure, but it is not a direct measurement of the complete centerline. The coarse ordered prior can have ambiguous ordering near crossings, while regions without visible support remain geometrically uncertain. It therefore provides approximate order but does not determine the target geometry on its own.

Temporal trackers solve a related problem by propagating an ordered state across frames. The initialisation procedure used in TrackDLO (hereafter TrackDLO-init) provides the initial 3D configuration and node order before the temporal updates [2]. Since these updates start from the initialised state, a valid initialisation is required before tracking can begin. The estimator in this thesis instead predicts an ordered state from the current RGB-D observation and a coarse geometric prior, so it can be evaluated as an alternative initialisation procedure.

The research problem is therefore to combine visible 3D evidence, which is partial and unordered, with a coarse prior, which is ordered but geometrically uncertain. Neither source determines the complete ordered state on its own.

## 1.2 Proposed Approach: Geometry-Conditioned Single-Frame Refinement

This thesis addresses the observation-to-state gap with a geometry-conditioned single-frame estimator. For each RGB-D observation, the first stage constructs two complementary geometric conditions. The visible evidence is an unordered set of 3D surface points measured from depth pixels selected by a SAM3-derived binary DLO mask [6]. The coarse ordered prior is obtained by extracting an ordered 2D path from the SAM3 mask and lifting its nodes into 3D with nearby visible measurements. A support indicator records which

prior nodes have direct observation support. Together, the visible points provide measured geometry, while the path-lifted prior provides approximate order across the complete centerline. An optional node-wise local depth-residual condition is also evaluated.

The second stage uses a geometry-conditioned velocity-flow model trained with Flow Matching to predict a velocity field for the ordered DLO state. Flow Matching learns a vector field along a probability path from a simple source distribution to the target data distribution [7]. During inference, the model starts from a Gaussian source state for the ordered point sequence and updates it over several refinement steps. At each step, the predicted velocity depends on both the geometric conditions and the current intermediate state $\mathbf{x}_t$. Here, the current frame refers to the RGB-D observation, whereas $\mathbf{x}_t$ is an intermediate state within the flow refinement, not a video frame.

An observation-based plausibility check identifies unreliable model outputs and returns the geometric prior as a fallback. The final output is a complete ordered 3D centerline estimated from the current observation and the coarse prior. Because it does not require a previously tracked ordered state, the estimator can also provide an initial state for temporal tracking.

## 1.3 Research Questions

The central research question of this thesis is:

> To what extent can a complete ordered 3D DLO centerline be recovered from a single RGB-D observation under self-occlusion?

This central question is divided into three specific research questions:

**RQ1** **Complementary geometric evidence.** What complementary roles do an ordered but geometrically coarse prior and unordered visible 3D evidence play in reconstructing a complete ordered 3D DLO centerline?

**RQ2** **Direct-state and velocity parameterization.** How does the choice between direct-state and velocity prediction affect reconstruction behaviour across multiple refinement steps?

**RQ3** **Node-wise local depth evidence.** How does node-wise local depth information affect particle-level correspondence and curve-level reconstruction accuracy under incomplete visibility?

## 1.4 Contributions

The main contributions of this thesis are:

- **Geometry-conditioned velocity-flow framework.** A geometry-conditioned velocity-flow framework is developed for complete ordered 3D DLO state estimation from a single RGB-D observation. The framework combines visible 3D evidence with a coarse ordered prior and does not require an accurate ordered state from the previous frame. It also includes an observation-based fallback that returns the geometric prior when the learned prediction is judged unreliable.
- **Path-lifted coarse ordered prior with visible support.** An ordered 2D path is lifted into 3D using visible 3D measurements to obtain a coarse ordered prior. A support representation indicates which parts of this prior are directly supported by the current observation.
- **Systematic analysis of prediction parameterization and local depth evidence.** Controlled ablations compare direct clean-state prediction with velocity prediction and test node-wise local depth-residual conditioning. In the evaluated setting, velocity prediction produces stronger dependence on the intermediate state, while local depth residuals do not provide consistent gains under self-occlusion.
- **Synthetic RGB-D data generation and initialisation evaluation.** A simulation-based pipeline is developed to generate controlled RGB-D observations together with complete ordered 3D ground-truth centerlines. The pipeline supports controlled evaluation across simple and self-intersecting configurations. Compared with TrackDLO-init, the proposed estimator produces valid ordered estimates across a broader range of configurations and achieves lower geometric errors in most cases where both procedures succeed.

# 2 Foundations and Related Work

Ordered 3D state estimation combines two forms of information: an endpoint-to-endpoint centerline order and partial RGB-D measurements of the visible object surface. Existing methods address different parts of this problem through image-based path extraction, temporal tracking, learning-based point-set processing, and generative modelling.

## 2.1 DLO State Representation and RGB-D Observation

Unlike a rigid object, a deformable linear object cannot be described by a single position and orientation. The shape of a deformable linear object may change along its entire length and must therefore be represented by a spatial curve. Robotic DLO research uses representations ranging from discrete centerlines to physics-based rod models, depending on the estimation or manipulation task [8], [9]. A common discrete representation is an ordered chain of 3D nodes sampled along the DLO [3], [9]. In this thesis, the DLO state is represented by an ordered sequence of $N$ particles sampled along the rope centerline:

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^\mathsf{T} \in \mathbb{R}^{N\times 3}. \tag{2.1}$$

In Equation (2.1), each particle $\mathbf{x}_i = [x_i, y_i, z_i]^\mathsf{T}$ represents one position on the rope centerline. The particle order follows the rope from one endpoint to the other and therefore describes both its geometry and its internal sequence. Unless stated otherwise, the DLO states used for learning and evaluation are expressed in the camera coordinate system. This choice provides a common frame for the depth observation, 3D unprojection, image projection, and predicted rope state.

The input observation in this thesis consists of an RGB image, a depth map, a SAM3-derived binary DLO mask [6], and the camera intrinsics. The RGB image contains appearance information, while the SAM3 mask identifies image pixels that belong to the rope. For a

pixel selected by the SAM3 mask with image coordinates $(u, v)$ and valid camera $z$-depth $d$, the corresponding visible 3D point can be obtained by unprojection:

$$x = \frac{(u - c_x)d}{f_x}, \qquad y = \frac{(v - c_y)d}{f_y}, \qquad z = d. \tag{2.2}$$

Here, $f_x$ and $f_y$ are the focal lengths, while $c_x$ and $c_y$ are the principal-point coordinates. The variable $d$ denotes the camera $z$-depth at pixel $(u, v)$, rather than the distance along the viewing ray. Applying this operation to the valid rope pixels produces an observed point cloud:

$$P_{\text{obs}} = \left\{ \mathbf{p}_j \in \mathbb{R}^3 \right\}_{j=1}^{M}. \tag{2.3}$$

The set in Equation (2.3) describes visible 3D measurements on the rope surface. They should not be confused with the ordered centerline particles in $\mathbf{X}$.

Two main difficulties arise from this observation model. First, $P_{\text{obs}}$ is an unordered point set, while the target state $\mathbf{X}$ must follow a consistent order along the rope centerline. Second, the observed point cloud is limited to visible surfaces. When one rope segment occludes another, the depth map records the front visible surface, while the hidden segment has no direct measurement at the same image location. As a result, $P_{\text{obs}}$ is generally partial, unordered, and visibility-limited [3]. Visible 3D evidence alone is therefore not sufficient to reliably determine the complete ordered centerline in occluded configurations. These limitations motivate state estimation methods that combine visible observations with ordered geometric structure and learned shape information.

## 2.2 Image-Based DLO Perception and Ordered Paths

A common starting point for image-based DLO perception is to segment the object from the background. FASTDLO [10] addresses instance segmentation of DLOs in images, while general promptable segmentation models provide another source of object masks. SAM introduced prompt-based segmentation for images [11]. SAM2 extended this formulation to images and videos [12], and SAM3 added concept-based detection and segmentation [6]. A segmentation mask identifies the object region, but it does not by itself define an order along the DLO centerline.

Several DLO-specific methods combine image segmentation with geometric processing. De Gregorio et al. estimate a DLO model using random walks on a superpixel adjacency graph [13]. RT-DLO combines learned semantic segmentation with graph-based processing

for real-time DLO instance segmentation [14]. Sun et al. present an RGB-D pipeline that covers DLO segmentation, occlusion-aware 3D reconstruction, and curve smoothing [15]. These methods provide object masks, image-plane curves, or reconstructed 3D geometry, depending on their intended task.

Other methods construct an ordered curve from the segmented image region. mBEST thins the mask to a skeleton and traverses the skeleton pixels to produce an ordered centerline [4]. At an image-plane intersection, the skeleton contains several branches, so the method must decide which branches belong to the same continuous traversal. mBEST selects the branch pairing that minimizes cumulative bending energy. This criterion resolves 2D connectivity at a crossing, but it does not determine which overlapping segment lies in front in 3D. DLOFTBs also starts from a DLO mask, orders the extracted skeleton segments, and fits a B-spline to the resulting path [16]. These methods recover ordered image-plane structure, but they do not directly provide the complete ordered 3D centerline required in this thesis.

AssistDLO uses segmentation, Voronoi-based skeletonization, and graph pruning to extract a continuous 2D DLO trace from each camera view [5]. The trace points are back-projected with the measured depth and fused across two wrist-mounted RGB-D cameras. The resulting non-parametric 3D point set represents observable geometry for teleoperation assistance, including local grasp guidance and visual augmentation [5]. This representation is suited to its task-level objective. The prior-construction stage in this thesis adapts the trace-extraction stage of AssistDLO [5] to produce a fixed-cardinality ordered 3D centerline, including locations without direct depth support.

The SAM3 mask identifies the DLO region, while the ordered 2D path represents its traversal in the image plane. Neither representation defines an ordered 3D state. In particular, the 2D path does not contain the depth of each centerline point or directly determine the relative depth ordering of overlapping rope segments. The observed point cloud $P_{\text{obs}}$, obtained from the SAM3 mask, depth map, and camera intrinsics, provides metric 3D information for visible parts of the rope. However, as discussed in Section 2.1, $P_{\text{obs}}$ is partial, unordered, and limited by visibility. The two representations therefore provide complementary information: the ordered 2D path provides image-plane order and coarse structure, while $P_{\text{obs}}$ provides visible 3D geometry. The construction of an ordered geometric prior from these complementary sources is described in Chapter 3.

## 2.3 Temporal DLO Tracking

Many temporal DLO trackers update a model from the previous frame instead of estimating the complete state independently in every frame. A common basis is Coherent Point Drift (CPD), which formulates non-rigid point set registration as a probabilistic alignment problem [17]. CDCPD registers a previous rope or cloth model to the current segmented point cloud using visibility reasoning and geometric regularization [18]. CDCPD2 extends this formulation with motion-model regularization and convex constraints for self-intersection and obstacle penetration [19]. Structure Preserved Registration instead aligns a deformable object model with an observed point cloud while regularizing local and global structure [20]. Yang et al. learn a low-dimensional latent representation of DLO shape and perform particle-filter tracking in that latent space [21]. TrackDLO represents the DLO as an ordered node sequence and uses motion coherence to estimate the motion of occluded nodes from visible ones [2].

The initialisation procedure used in TrackDLO (hereafter TrackDLO-init) is treated as a separate step before temporal updates. TrackDLO-init de-projects an ordered pixel chain into 3D and samples equally spaced nodes along the resulting chain [2]. The tracker then updates this ordered state in later frames using the previous node positions and the current observation. TrackDLO-init therefore supplies both the initial 3D configuration and the order of the tracked nodes.

These methods use temporal continuity and can provide stable estimates when the initialised state and inter-frame correspondences remain reliable. However, registration-based trackers require an initial state and update later states from earlier estimates. Lv et al. note that pure tracking-based methods rely on accurate initial states and provide few effective ways to correct accumulated drift or reinitialise after failure [3]. CDCPD2 assumes relatively small changes between consecutive frames [19], while TrackDLO reports good depth resolution and small inter-frame motion among its requirements [2]. In contrast, this thesis estimates the ordered state from the current RGB-D observation and a geometric prior, without requiring a propagated previous-frame state.

## 2.4 Learning-Based DLO State Estimation

Learning-based methods provide an alternative to manually designed geometric pipelines and registration-based trackers. Collecting paired real observations and complete ordered 3D states is difficult and time-consuming. The methods discussed below use simulation to generate this supervision [3], [22].

Point clouds are unordered sets, so their processing should not depend on input order. Deep Sets formalizes permutation-invariant functions, PointNet uses shared point-wise features with symmetric pooling, and PointNet++ adds hierarchical local neighborhoods [23]–[25]. Attention provides another set-processing mechanism: the Transformer introduced self-attention, Set Transformer adapted it to permutation-invariant sets, and Perceiver maps large inputs to latent arrays through cross-attention [26]–[28].

Point-cloud completion recovers shapes from partial observations. PCN and TopNet use coarse-to-fine or hierarchical decoders, whereas PoinTr and SnowflakeNet use transformers or structured point growth [29]–[32]. Their outputs are unordered surface point sets, not fixed-cardinality DLO centerlines ordered between two endpoints.

Lv et al. estimate an ordered sequence of 3D nodes from a single occluded point cloud [3]. Their model combines a global regression branch with a point-to-point voting branch, followed by non-rigid registration-based fusion. The global branch provides a smooth complete shape, while the voting branch uses local visible geometry for more precise node estimates.

UniStateDLO formulates both single-frame estimation and cross-frame tracking from partial point clouds as conditional generative problems [22]. UniStateDLO's single-frame module also produces global and local coarse predictions, but uses a conditional diffusion model to fuse them into the final ordered nodes. For cross-frame tracking, the previous nodes provide coarse locations around which local point-cloud features are collected, and a second diffusion model predicts node-wise motion. The present thesis studies the same single-frame output but conditions the estimation on an image-derived ordered geometric prior and a separate set of visible 3D observations.

## 2.5 Flow Matching and Observation-Guided Generation

DDPMs model data generation as the reverse of a gradual noise process [33]. A neural network predicts the added noise at each noise level, and samples are generated through iterative reverse steps. Flow Matching trains a continuous-time vector field along prescribed probability paths between source and target distributions [7]. Straight source-to-target paths are also used by Rectified Flow, which learns an ODE to transport paired samples along these paths [34]. For the formulation considered in this thesis, the path between a source state $\mathbf{x}_0$ and a clean ordered rope state $\mathbf{x}_1$ is linear:

$$\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1, \qquad t \in [0,1]. \tag{2.4}$$

Here, $\mathbf{x}_0$ denotes a source state sampled from a standard normal distribution in the normalized model space, $\mathbf{x}_1$ denotes the clean ordered rope state, and $\mathbf{x}_t$ is the intermediate state at continuous time $t$. The corresponding target velocity is constant along the linear path:

$$\mathbf{v}_{\text{target}} = \mathbf{x}_1 - \mathbf{x}_0. \tag{2.5}$$

Continuous generative models have also been developed for 3D point sets. PointFlow models point-cloud generation with continuous normalizing flows [35], while Luo and Hu apply diffusion models directly to point clouds [36]. Point-Voxel Diffusion combines point and voxel representations for 3D shape generation and completion [37]. These methods generate unordered shape point sets and provide the generative background for observation-conditioned reconstruction.

Generative refinement can use observations of the target object as additional conditioning. PC$^2$ [38] applies conditional diffusion to single-image 3D reconstruction and generates a sparse point cloud from an RGB image and a known camera pose. At each denoising step, it projects local image features onto the partially denoised points, so the conditioning is evaluated at the current intermediate 3D state.

Observation information can also be introduced through guidance during generative sampling. Diffusion Posterior Sampling (DPS) [39] combines a diffusion prior with gradients of the measurement likelihood to approximate posterior sampling for noisy linear and nonlinear inverse problems. FlowDPS [40] extends this posterior-guidance idea to flow-based generative models by incorporating measurement likelihood information into the flow dynamics. These methods provide general examples of observation-guided generation. The present work instead conditions the learned velocity field directly on geometric observations; the detailed design is given in Chapter 3.

The reviewed work leaves a gap between image-plane structure, visible 3D evidence, and a complete ordered 3D DLO state. Image-based methods recover masks, ordered image-plane traces, or task-relevant 3D point sets, but these representations do not by themselves determine the complete ordered 3D centerline [4], [5]. Learning-based methods can infer ordered states from partial point clouds [3], [22], but this thesis studies a different combination of information: an image-derived ordered prior, separate visible 3D evidence, and the current intermediate state. Chapter 3 introduces the resulting single-frame, geometry-conditioned velocity-flow formulation.

# 3 Methodology

An ordered 3D centerline cannot be recovered from either the visible RGB-D points or the image-plane path alone. The proposed method therefore keeps these two sources separate and uses them jointly to condition a velocity-flow estimator. It estimates the DLO state from a single RGB-D observation and does not require a state propagated from a previous frame.

## 3.1 Problem Formulation and Method Overview

The available input consists of an RGB image $\mathbf{I}$, a depth map $\mathbf{D}$, a SAM3-derived binary DLO mask $\mathbf{M}$ [6], and the camera intrinsics $\mathbf{K}$. All inputs belong to the same frame. The target is a complete ordered centerline $\mathbf{X}_1 \in \mathbb{R}^{N\times 3}$, where $N = 64$. Each row of $\mathbf{X}_1$ contains one centerline point in the camera coordinate system. The rows follow the DLO from one endpoint to the other.

The observation is converted into three geometric conditions. The point set $P_{\text{obs}}$ contains visible 3D measurements and has no centerline order. The path-lifted prior provides $N$ ordered 3D points, but its geometry is only approximate. A support indicator records which prior nodes have nearby visible measurements. The SAM3 mask and the camera intrinsics are also retained for the local projection features used inside the flow model. These quantities are derived from the current observation; they are not additional sensor inputs.

At flow time $t$, the model receives the current intermediate state $\mathbf{x}_t$ together with the geometric conditions. It predicts one velocity vector for each ordered centerline point. A multi-step Euler sampler integrates these velocities into the normalized state $\mathbf{x}_K$, which is transformed to the camera-space estimate $\mathbf{X}_{\text{pred}}$. An observation-based plausibility check then decides whether this estimate is used. If the estimate is rejected, the path-lifted prior is returned as a fallback. The full estimation process is shown in Figure 3.1.

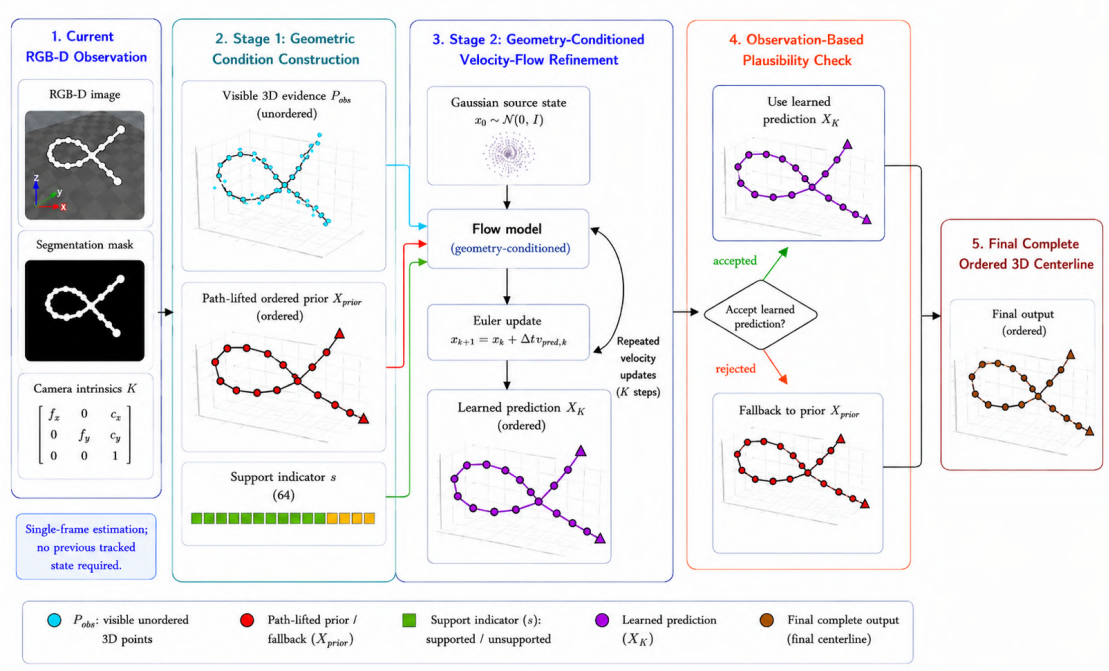



Figure 3.1: Overview of the proposed single-frame estimator. Stage 1 converts the current RGB-D observation and SAM3 mask into the unordered visible point set $P_{\text{obs}}$, a path-lifted ordered prior, and its support indicator. Stage 2 conditions a velocity-flow model on these quantities and applies repeated Euler updates to obtain the learned ordered estimate. The plausibility check accepts this estimate when it is consistent with the current observation. Otherwise, the ordered prior is returned as a fallback. The selected state is the final complete ordered 3D centerline.

The two geometric conditions have different roles. $P_{\text{obs}}$ preserves measured 3D geometry, while the path-lifted prior provides order across the centerline. Neither condition is treated as the final state on its own.

## 3.2 Visible 3D Evidence from RGB-D

The visible 3D evidence is obtained from the SAM3 mask, depth map, and camera intrinsics. A pixel $(u, v)$ is retained when its SAM3 mask value is greater than $0.5$. The retained pixel's depth value must also be finite and greater than $10^{-6}$ meters. Each valid pixel is transformed into a camera-space point using Equation (2.2). The depth value is the camera $z$-depth, not the distance along the viewing ray. The resulting raw point set, denoted by $\widetilde{P}_{\text{obs}}$, is expressed in meters in the camera coordinate system.

The number of valid pixels changes between observations. However, batched model inputs require a common tensor shape. The raw point set is therefore converted to $K_{\text{obs}} = 256$ entries before it is normalized. When more than 256 points are available, deterministic 3D farthest point sampling is applied [25]. The first point is the point farthest from the point-cloud centroid. Each later point maximizes its minimum Euclidean distance to the points that have already been selected. This procedure gives a spatially distributed subset while preserving the unordered point-set representation.

When fewer than 256 valid points are available, all measured points are kept. Existing points are then repeated to fill the remaining entries. If no valid point is available, the point tensor is filled with zeros. The final model input has the shape $P_{\text{obs}} \in \mathbb{R}^{256\times 3}$. A Boolean observation-validity mask $\mathbf{m}_{\text{obs}}$ of length 256 distinguishes measured points from repeated or zero padding. The observation-validity mask entries are true for measured points and false for padding. For a batch, this mask has the shape $B \times 256$. Here, $B$ denotes the batch size. It is later inverted and used as the key-padding mask in cross-attention. The $256$ evidence points remain unordered and are separate from the $N = 64$ ordered centerline particles. Their role is to represent visible surface geometry, not the complete ordered DLO state.

## 3.3 Ordered 2D Path and Path-Lifted 3D Prior

The SAM3 mask provides the image region associated with the DLO, but it does not directly define a global order along the object. AssistDLO provides a mask-to-trace construction based on sparse contour sampling, a Voronoi graph, and graph pruning [5]. It selects the endpoints of the main visible trace by their geodesic distance in the pruned graph and back-projects the remaining trace nodes using the measured depth. In the present pipeline, a separate upstream module provides an ordered 2D path from the SAM3 mask. The construction described below starts from this path. It adds fixed-cardinality resampling, association with the visible 3D observations, interpolation of unsupported nodes, and deterministic endpoint ordering. These operations are specific to the path-lifted prior in this thesis and are not part of the AssistDLO representation.

The ordered 2D path and the observed point set $P_{\text{obs}}$ provide complementary information. The path defines image-plane order but does not by itself assign a 3D position to every centerline location. In contrast, $P_{\text{obs}}$ provides visible camera-space geometry but remains partial and unordered. The path-lifting procedure combines these two representations to obtain a coarse ordered 3D structure. Figure 3.2 summarizes the complete Stage 1 construction.

After consecutive duplicate points have been removed, the ordered image-plane path provided by the upstream module is represented as $\mathcal{Q} = (\mathbf{q}_1, \ldots, \mathbf{q}_M)$. Here, $\mathbf{q}_m = (u_m, v_m)^\top \in \mathbb{R}^2$ is the pixel coordinate of the $m$-th path sample in traversal order, and $M$ is the number of retained path samples before fixed-cardinality resampling and may vary between observations. For these points, the cumulative arclength is

$$\ell_1 = 0, \qquad \ell_m = \sum_{r=2}^{m} \|\mathbf{q}_r - \mathbf{q}_{r-1}\|_2 \,. \tag{3.1}$$

The target arclengths are uniformly placed between the two endpoints,

$$s_i = \frac{i-1}{N-1}\ell_M, \qquad i \in \{1, \ldots, N\}, \qquad N = 64. \tag{3.2}$$

Each resampled node $\bar{\mathbf{q}}_i$ is obtained by linear interpolation on the path segment containing $s_i$. The first and last nodes are then set exactly to $\mathbf{q}_1$ and $\mathbf{q}_M$, respectively. This endpoint assignment preserves the original path endpoints while producing a fixed number of ordered nodes.

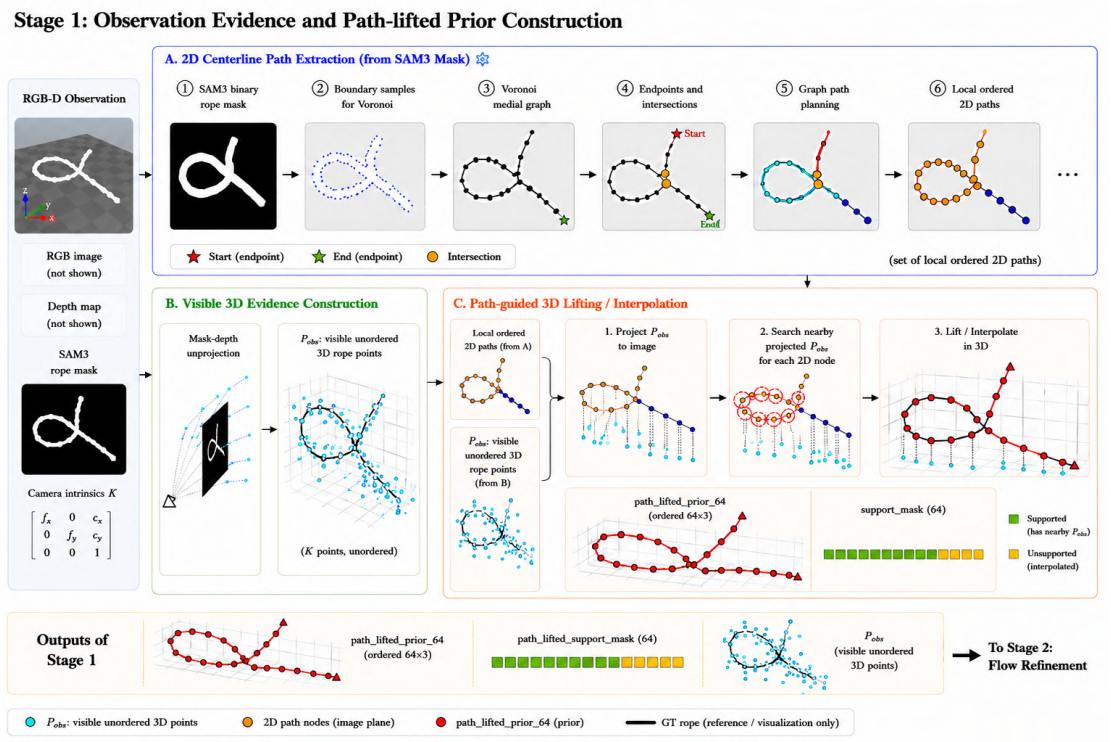



Figure 3.2: Detailed construction of the Stage 1 conditions. The upstream mask-to-path stage extracts ordered image-plane structure from the SAM3 mask. Valid SAM3-mask pixels and their depth measurements are separately unprojected to form the unordered visible point set $P_{\text{obs}}$. The proposed path-lifting stage projects these observations into the image, associates them with the resampled path nodes, and fills nodes without direct support by interpolation or endpoint extension. The proposed path-lifting stage outputs the ordered prior $\mathbf{X}_{\text{prior}}$, the corresponding support indicator, and the separate unordered point set $P_{\text{obs}}$. The ground-truth curve shown in the illustration is used only as a visual reference and is not an input to Stage 1.

Each observed point $\mathbf{p}_j \in P_{\text{obs}}$ is projected into the image and associated with an arclength position $\hat{s}_j$ on the path. For node $i$, the candidate set is

$$\mathcal{C}_i = \left\{ j \;\middle|\; \|\pi_{\mathbf{K}}(\mathbf{p}_j) - \bar{\mathbf{q}}_i\|_2 \le r_{\text{px}}, \quad |\hat{s}_j - s_i| \le r_{\text{arc}} \right\}, \tag{3.3}$$

where $\pi_{\mathbf{K}}$ denotes projection with the camera intrinsics. The thresholds are fixed at $r_{\text{px}} = 12$ pixels and $r_{\text{arc}} = 36$ pixels following a small-scale sensitivity analysis on 11 representative samples spanning simple and self-intersecting shapes. Across five tested radius–window combinations, the mean lifting error varied by less than 1 mm; the selected values are therefore retained as a stable fixed configuration rather than claimed as the numerical optimum. The two conditions prevent a point that is close in the image but belongs to a distant part of the path from being used for the lifting step. This restriction is particularly relevant near image-plane crossings.

For a non-empty candidate set, the weight assigned to observation $j$ is

$$w_{ij} = \exp\left( -\frac{1}{2} \frac{\|\pi_{\mathbf{K}}(\mathbf{p}_j) - \bar{\mathbf{q}}_i\|_2^2}{\sigma_{\text{px}}^2} \right) \exp\left( -\frac{1}{2} \frac{|\hat{s}_j - s_i|^2}{\sigma_{\text{arc}}^2} \right). \tag{3.4}$$

Here, $\sigma_{\text{px}} = 6$ pixels and $\sigma_{\text{arc}} = 12$ pixels. The lifted camera-space position is the normalized weighted average

$$\widetilde{\mathbf{x}}_i = \frac{\sum_{j \in \mathcal{C}_i} w_{ij} \mathbf{p}_j}{\sum_{j \in \mathcal{C}_i} w_{ij}}. \tag{3.5}$$

The support indicator is defined directly from the candidate set:

$$h_i = \mathbb{I}\left[ |\mathcal{C}_i| > 0 \right]. \tag{3.6}$$

Thus, $h_i = 1$ means that at least one projected observation passed both association thresholds for node $i$. It is not a ground-truth visibility label and does not state that the lifted position is geometrically exact.

Nodes with $h_i = 0$ are completed along the ordered sequence. Let $l(i)$ and $r(i)$ denote the nearest supported nodes to the left and right of node $i$. If both exist, the missing position is linearly interpolated as

$$\widetilde{\mathbf{x}}_i = (1 - \alpha_i)\widetilde{\mathbf{x}}_{l(i)} + \alpha_i \widetilde{\mathbf{x}}_{r(i)}, \qquad \alpha_i = \frac{i - l(i)}{r(i) - l(i)}. \tag{3.7}$$

If only one supported neighbour exists, its position is copied to the unsupported endpoint region. The support indicator remains zero for every filled node because interpolation or

endpoint extension does not add observation support. Equations (3.1)–(3.7) define the complete resampling, association, lifting, and completion procedure.

Finally, the direction of the ordered prior is canonicalized in camera coordinates. The endpoint with the smaller $x$-coordinate is selected as the first node; if the $x$-coordinates are equal, $y$ and then $z$ are used as tie-breakers. If the current first endpoint is larger under this coordinate-wise rule, the complete node sequence and its support indicator are reversed. The resulting outputs are $\mathbf{X}_{\text{prior}} \in \mathbb{R}^{64\times 3}$ and $\mathbf{h} \in \{0,1\}^{64}$. The ordering rule is deterministic and does not use the ground-truth state. The ordered prior supplies coarse structure to the flow model, whereas $P_{\text{obs}}$ remains a separate unordered source of measured geometry.

## 3.4 Geometry-Conditioned Velocity Model

The model operates in one globally normalized 3D coordinate system. Let $\mathcal{T}$ be the training split and let $\mathbf{X}_{s,i}$ denote ground-truth particle $i$ of training sample $s$ in camera coordinates. The global mean and isotropic scale are

$$\boldsymbol{\mu} = \frac{1}{|\mathcal{T}|N} \sum_{s\in\mathcal{T}} \sum_{i=1}^{N} \mathbf{X}_{s,i}, \tag{3.8}$$

$$\sigma_{\text{iso}} = \left( \frac{1}{|\mathcal{T}|N} \sum_{s\in\mathcal{T}} \sum_{i=1}^{N} \|\mathbf{X}_{s,i} - \boldsymbol{\mu}\|_2^2 \right)^{1/2}. \tag{3.9}$$

Both statistics are computed only from the ground-truth particles of the training split. The normalization from camera coordinates to model coordinates and its inverse are

$$\text{Norm}(\mathbf{X}) = \frac{\mathbf{X} - \boldsymbol{\mu}}{\sigma_{\text{iso}}}, \qquad \text{Norm}^{-1}(\mathbf{x}) = \sigma_{\text{iso}}\mathbf{x} + \boldsymbol{\mu}. \tag{3.10}$$

Equations (3.8)–(3.10) define the coordinate transformation used by the model.

At flow time $t$, the current intermediate state is $\mathbf{x}_t \in \mathbb{R}^{N\times 3}$, with $N = 64$. The geometric conditions are collected as

$$\mathbf{c} = \left(\mathbf{X}_{\text{prior}}, \mathbf{h}, P_{\text{obs}}, \mathbf{m}_{\text{obs}}, \mathbf{M}, \mathbf{K}\right). \tag{3.11}$$

They contain the ordered prior and its support indicator, the unordered visible point set and its validity mask, the SAM3 mask, and the camera intrinsics. The camera-space 3D

prior and observation points in this set are transformed with Equation (3.10) before being embedded by the model. The model defines the mapping

$$\mathbf{v}_\theta : (\mathbf{x}_t, t, \mathbf{c}) \longmapsto \mathbf{v}_{\text{pred}} \in \mathbb{R}^{N\times 3}. \tag{3.12}$$

The velocity mapping assigns one 3D velocity to each ordered centerline node. The variable $t$ denotes continuous flow time and is unrelated to a video frame index. Figure 3.3 shows the role of this mapping in the refinement process and its internal architecture.

The ordered state is represented by one token per centerline node. For node $j$, the initial token is

$$\mathbf{z}_j^{(0)} = E_x\left(\mathbf{x}_{t,j}\right) + E_t(t) + \mathbf{e}_j + E_{\text{prior}}\left(\text{Norm}(\mathbf{X}_{\text{prior},j})\right) + E_h(h_j), \tag{3.13}$$

where $E_x$, $E_t$, $E_{\text{prior}}$, and $E_h$ are learned projections to a common hidden dimension. The learned embedding $\mathbf{e}_j$ identifies the position of node $j$ in the ordered centerline. The prior term embeds the complete normalized 3D coordinate of the corresponding prior node and therefore supplies coarse ordered geometry rather than only a depth value. The indicator $h_j$ distinguishes nodes with direct observation support from nodes completed by interpolation or endpoint extension.

The visible point set is encoded independently. Each point $\mathbf{p}_m \in P_{\text{obs}}$ is mapped by a shared point-wise multilayer perceptron to a context token $\mathbf{g}_m = E_{\text{obs}}(\text{Norm}(\mathbf{p}_m))$. The ordered particle tokens then query these context tokens through cross-attention:

$$\mathbf{Z}^{(\text{obs})} = \text{CrossAttn}\left(\text{query} = \mathbf{Z}^{(0)}, \text{key} = \mathbf{G}, \text{value} = \mathbf{G}; \mathbf{m}_{\text{obs}}\right). \tag{3.14}$$

The validity mask prevents repeated or zero-padded points from contributing. This asymmetric attention follows the general query-to-input construction used in attention models [26], [28]. It lets every ordered node read the valid measurements without assigning an artificial order to $P_{\text{obs}}$. All valid context tokens remain available to each query.

The second observation interaction is local and state-dependent. For node $j$, the normalized current position is first transformed back to camera coordinates and then projected: $\mathbf{X}_{t,j}^{\text{cam}} = \text{Norm}^{-1}(\mathbf{x}_{t,j})$ and $\mathbf{u}_j = \pi_{\mathbf{K}}(\mathbf{X}_{t,j}^{\text{cam}})$. At this location, the model samples the SAM3 mask value, a locally smoothed SAM3 mask response, the distance to the segmented region, and the two image-plane components of the displacement to the nearest SAM3-mask pixel. A sixth value indicates whether the projection lies in front of the camera and inside the image. These quantities form the local feature vector $\boldsymbol{\phi}_{\mathbf{M}}(\mathbf{u}_j) \in \mathbb{R}^6$. It is embedded and added to the corresponding particle token,

$$\mathbf{z}_j^{(\text{loc})} = \mathbf{z}_j^{(\text{obs})} + \gamma E_{\text{loc}}\left(\boldsymbol{\phi}_{\mathbf{M}}(\mathbf{u}_j)\right), \tag{3.15}$$

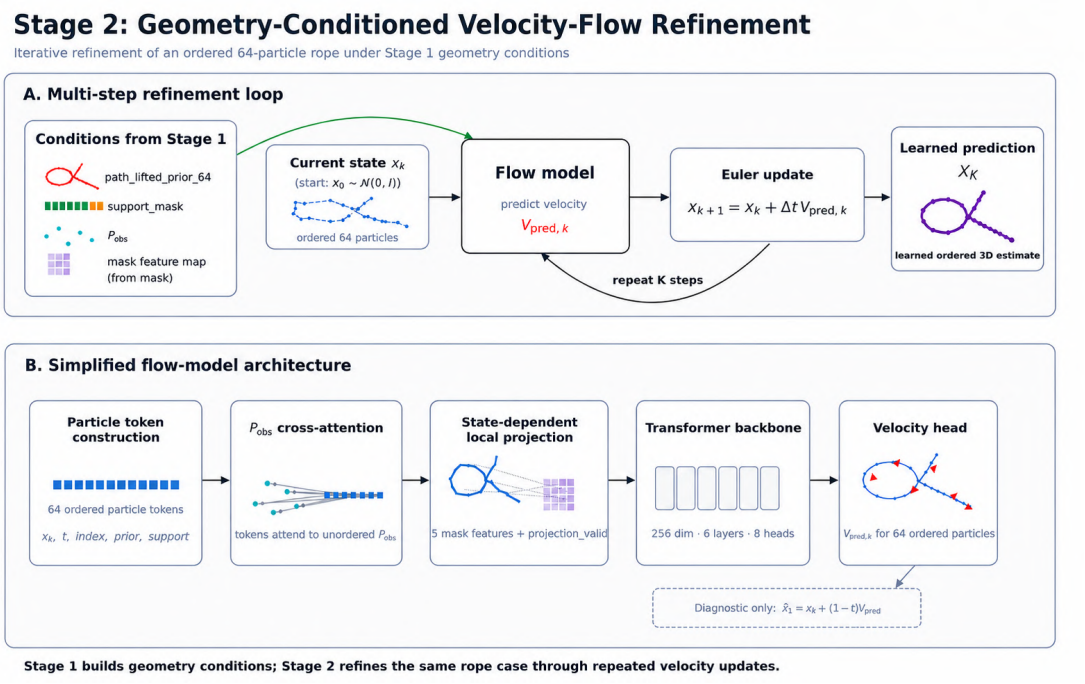



Figure 3.3: Detailed view of the Stage 2 velocity refinement. The upper panel places the geometry-conditioned velocity model inside the multi-step Euler refinement loop. The lower panel shows the model architecture. Ordered particle tokens are formed from the current state, flow time, particle order, path-lifted prior, and support indicator. They read the unordered visible evidence through cross-attention and receive local mask information at their current image projections before entering the Transformer backbone. A particle-wise head produces the velocity field used by the sampler. The one-step clean-state reconstruction shown below the model is used for training diagnostics and auxiliary terms, not as the Euler update. The local-projection block depicts the six-channel mask-only configuration.

where $E_{\text{loc}}$ is a multilayer perceptron and $\gamma$ is learned. This design follows the projection-conditioning principle of PC$^2$ [38], but uses geometric mask features instead of RGB image features. Because $\mathbf{u}_j$ depends on $\mathbf{x}_{t,j}$, the local condition is recomputed when the intermediate state changes.

A depth-residual ablation replaces the six-channel mask-only feature vector with an eight-channel mask-and-depth vector. Let $z_j^{\text{cam}}$ be the camera-space depth of the current node after the normalized state has been transformed back to camera coordinates, and let $\mathbf{D}(\mathbf{u}_j)$ be the observed depth at its projection. The signed depth residual is

$$r_j^{\text{depth}} = \frac{\text{clip}\left(\mathbf{D}(\mathbf{u}_j) - z_j^{\text{cam}}, -0.2\,\text{m}, 0.2\,\text{m}\right)}{0.2\,\text{m}}. \tag{3.16}$$

Thus, the residual is normalized to the interval $[-1, 1]$. It is marked valid only if the 3D projection is valid, the projected pixel lies inside the image, both $z_j^{\text{cam}}$ and $\mathbf{D}(\mathbf{u}_j)$ are finite, and the observed depth lies between $0.05\,\text{m}$ and $0.5\,\text{m}$. The normalized residual and this binary validity value form the two additional channels. The six-channel mask-only input and the eight-channel mask-and-depth input are two mutually exclusive configurations and are evaluated separately. Equations (3.11)–(3.16) specify the geometric conditions and their model embeddings.

The resulting ordered tokens are processed by a Transformer encoder with hidden dimension 256, six layers, and eight attention heads [26]. Here, self-attention models relations among the 64 ordered DLO nodes, whereas the preceding cross-attention transfers information from the unordered visible point set. A particle-wise output head maps each final token to $\mathbf{v}_{\theta,j}(\mathbf{x}_t, t, \mathbf{c}) \in \mathbb{R}^3$. Stacking these outputs gives the velocity field in Equation (3.12). The next two sections define the training target and explain how this field is integrated during inference.

## 3.5 Training Objective

The normalized clean state is $\mathbf{x}_1 = \text{Norm}(\mathbf{X}_1)$. The intermediate state, path-lifted prior, and $P_{\text{obs}}$ use the same mean vector and single scalar scale. The Gaussian source $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is sampled directly in this normalized model space.

For $t \in [0, 1]$, training uses the linear flow path in Equation (2.4) and the corresponding velocity target in Equation (2.5). The model predicts $\mathbf{v}_{\text{pred}} = \mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{c})$. The primary loss

is the mean squared velocity error over the batch, the ordered particles, and the three coordinate dimensions:

$$\mathcal{L}_{\text{vel}} = \frac{1}{3BN} \sum_{b=1}^{B} \sum_{i=1}^{N} \left\| \mathbf{v}_{\text{pred},b,i} - (\mathbf{x}_{1,b,i} - \mathbf{x}_{0,b,i}) \right\|_2^2 . \tag{3.17}$$

The velocity prediction also gives a one-step estimate of the clean endpoint,

$$\hat{\mathbf{x}}_1 = \mathbf{x}_t + (1-t)\mathbf{v}_{\text{pred}}. \tag{3.18}$$

This estimate is transformed back to camera coordinates before the geometry regularizers are evaluated, giving $\hat{\mathbf{X}}_1 = \text{Norm}^{-1}(\hat{\mathbf{x}}_1)$. For sample $b$, let the predicted segment lengths and the mean ground-truth segment length be

$$\hat{d}_{b,i} = \left\| \hat{\mathbf{X}}_{1,b,i+1} - \hat{\mathbf{X}}_{1,b,i} \right\|_2 , \tag{3.19}$$

$$\bar{d}_b^{\text{gt}} = \frac{1}{N-1} \sum_{i=1}^{N-1} \left\| \mathbf{X}_{1,b,i+1} - \mathbf{X}_{1,b,i} \right\|_2 . \tag{3.20}$$

The spacing loss is the mean absolute relative deviation from this sample-specific reference length:

$$\mathcal{L}_{\text{spacing}} = \frac{1}{B(N-1)} \sum_{b=1}^{B} \sum_{i=1}^{N-1} \frac{\left| \hat{d}_{b,i} - \bar{d}_b^{\text{gt}} \right|}{\bar{d}_b^{\text{gt}}}. \tag{3.21}$$

The bending term penalizes sharp local reversals in the predicted curve. For each interior particle, define the two adjacent directed segments

$$\mathbf{e}_{b,i}^{-} = \hat{\mathbf{X}}_{1,b,i} - \hat{\mathbf{X}}_{1,b,i-1}, \qquad \mathbf{e}_{b,i}^{+} = \hat{\mathbf{X}}_{1,b,i+1} - \hat{\mathbf{X}}_{1,b,i}, \tag{3.22}$$

and their angle cosine

$$c_{b,i} = \frac{\left(\mathbf{e}_{b,i}^{-}\right)^{\mathsf{T}} \mathbf{e}_{b,i}^{+}}{\left\| \mathbf{e}_{b,i}^{-} \right\|_2 \left\| \mathbf{e}_{b,i}^{+} \right\|_2}. \tag{3.23}$$

Using a threshold of $90^\circ$, whose cosine is zero, gives

$$\mathcal{L}_{\text{bending}} = \frac{1}{B(N-2)} \sum_{b=1}^{B} \sum_{i=2}^{N-1} \left[ \max\left(0, -c_{b,i}\right) \right]^2 . \tag{3.24}$$

This term depends only on the predicted curve and does not use a ground-truth bending target. The complete training objective is

$$\mathcal{L} = \mathcal{L}_{\text{vel}} + \lambda_{\text{spacing}}\mathcal{L}_{\text{spacing}} + \lambda_{\text{bending}}\mathcal{L}_{\text{bending}}. \tag{3.25}$$

Equations (3.17)–(3.25) define the training objective and its geometric regularizers. The numerical loss weights are specified in the experimental setup.

In the direct-state ablation, the output head is interpreted directly as $\hat{\mathbf{x}}_1$ instead of a velocity. The primary objective of the direct-state ablation compares $\hat{\mathbf{X}}_1$ with $\mathbf{X}_1$ using a camera-space Smooth L1 loss with $\beta = 0.02\,\text{m}$, together with the same spacing and bending regularizers.

## 3.6 Inference and Plausibility-Based Fallback

Inference begins with $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ in normalized space. For $K$ Euler steps, the time and step size are

$$t_k = \frac{k}{K}, \qquad \Delta t = \frac{1}{K}, \qquad k \in \{0, \ldots, K-1\}. \tag{3.26}$$

The final update is therefore evaluated at $t_{K-1} = (K-1)/K$, not at $t = 1$. For the velocity model,

$$\mathbf{v}_{\text{pred},k} = \mathbf{v}_\theta(\mathbf{x}_k, t_k, \mathbf{c}), \tag{3.27}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t\, \mathbf{v}_{\text{pred},k}. \tag{3.28}$$

The direct-state ablation uses the same time grid and number of steps. The clean-state prediction $\hat{\mathbf{x}}_{1,k}$ of the direct-state ablation is converted to an instantaneous direction and integrated as

$$\mathbf{d}_k = \frac{\hat{\mathbf{x}}_{1,k} - \mathbf{x}_k}{\max(1 - t_k, \varepsilon)}, \qquad \mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t\, \mathbf{d}_k, \tag{3.29}$$

where $\varepsilon$ prevents division by zero. Equations (3.26)–(3.29) define the velocity and direct-state sampling updates. The choice of $K$ is evaluated in the experiments chapter. After the final step, $\mathbf{x}_K$ is transformed back to camera coordinates to obtain the learned prediction $\mathbf{X}_{\text{pred}}$.

Before this prediction is returned, an observation-based plausibility check compares it with the path-lifted prior $\mathbf{X}_{\text{prior}}$. Both curves are considered in camera coordinates. For an ordered curve $\mathbf{X}$, define its segment lengths and total arclength as

$$d_i(\mathbf{X}) = \|\mathbf{X}_{i+1} - \mathbf{X}_i\|_2 , \qquad L(\mathbf{X}) = \sum_{i=1}^{N-1} d_i(\mathbf{X}). \tag{3.30}$$

The hard geometric validity condition requires finite coordinates for both curves and

$$0.5 \leq \frac{L(\mathbf{X}_{\text{pred}})}{L(\mathbf{X}_{\text{prior}})} \leq 2.0, \qquad \frac{\max_i d_i(\mathbf{X}_{\text{pred}})}{L(\mathbf{X}_{\text{pred}})} \leq 0.10. \tag{3.31}$$

The first ratio rejects large changes in total curve length. The second detects a prediction in which one segment accounts for an implausibly large part of the curve.

The second condition compares image-mask agreement. The 64 curve nodes are projected into the image, and every projected segment is linearly densified at one-pixel intervals. The SAM3 mask is dilated by four pixels. The mask-agreement score $S_{\mathbf{M}}(\mathbf{X})$ is the fraction of the densified projected samples that fall on this dilated mask. The prediction must satisfy

$$S_{\mathbf{M}}(\mathbf{X}_{\text{pred}}) \geq S_{\mathbf{M}}(\mathbf{X}_{\text{prior}}) - 0.05, \tag{3.32}$$

and both input curves must again contain only finite coordinates. The learned prediction is accepted only if the hard geometric condition and the mask condition are both true. The final output is therefore

$$\mathbf{X}_{\text{out}} = \begin{cases} \mathbf{X}_{\text{pred}}, & \text{if the prediction is accepted,} \\ \mathbf{X}_{\text{prior}}, & \text{otherwise.} \end{cases} \tag{3.33}$$

Equations (3.30)–(3.33) define the complete plausibility and fallback rule.

# 4 Experiments

This chapter evaluates whether the proposed estimator can recover complete ordered three-dimensional DLO centerlines from a single observation, and which parts of the method are responsible for its behaviour. The evaluation combines three complementary forms of evidence. A held-out synthetic benchmark provides complete ordered ground truth for measuring reconstruction accuracy and for comparison with the initialisation procedure used in TrackDLO (hereafter TrackDLO-init). Official real-world failure-case recordings then test the complete pipeline outside the synthetic training distribution, where only output availability and consistency with visible image evidence can be assessed. Finally, controlled ablations examine the conditioning representation, prediction parameterization, local depth residual, and plausibility gate in relation to the research questions from Section 1.3. Together, these experiments distinguish in-distribution accuracy from out-of-distribution robustness and from the contribution of individual design choices.

## 4.1 Experimental Design

The comparisons require a common data split, inference protocol, and set of metrics so that observed differences can be attributed to the evaluated method rather than to changes in experimental conditions. This section therefore establishes the data, comparison methods, training protocol, and evaluation criteria shared by the subsequent experiments. The quantitative evaluation uses held-out synthetic configurations with complete ordered ground truth. Additional recordings without ground truth are considered separately in the failure-case evaluation.

### 4.1.1 Datasets and Splits

The synthetic dataset was generated with the Unity-based acquisition pipeline described in this thesis. Each sequence represents one static DLO configuration and contains an RGB-D observation, a binary segmentation mask, camera intrinsics, and a complete ordered ground-truth centerline with $N = 64$ particles. The simulated materials include blue, red, green, cyan-green, and dark-gray DLO appearances.

The assembled raw corpus contains 297 simulated sequences. Retaining only sequences for which the required observation evidence and geometric conditions can be constructed leaves 293 reliable sequences. The split is performed at the sequence level: 234 sequences are assigned to training and 59 sequences to validation. No sequence occurs in both subsets. This sequence-level split prevents the same static DLO configuration from appearing in training and validation. There is no separate test split.

The simulator records 100 frames for each static configuration. These frames share the same ground-truth geometry and are therefore not treated as independent geometric configurations. The training split contains 23,399 valid frame records. One empty rendering frame is removed because no DLO is visible and neither observation evidence nor a path-lifted prior can be constructed.[1] The validation split contains 5,900 frames. This frame count describes the size of the validation split rather than a second, frame-level result set. The sequence-level benchmark in Section 4.2 uses one representative frame from each of the 59 sequences. The middle frame, frame 50, is selected by a fixed policy for this comparison and for the qualitative visualizations.

The 59 validation sequences cover nine scenario groups: 6 C-shapes, 6 S-shapes, 2 straight configurations, 8 intersections, 7 complicated C-shapes, 8 complicated S-shapes, 2 complicated straight configurations, 8 complicated intersections, and 12 additional intersection configurations. Figure 4.1 shows three representative examples. The figure is illustrative; all nine groups are included in the quantitative evaluation.

### 4.1.2 Compared Methods

The external comparison uses the initialisation procedure used in TrackDLO, hereafter denoted TrackDLO-init [2]. This procedure segments the current observation using material-specific HSV thresholds and constructs an ordered centerline through its classical

[1] The removed record is frame 0 of `Straight/seq_0012`; the remaining training sequences retain all valid frames.

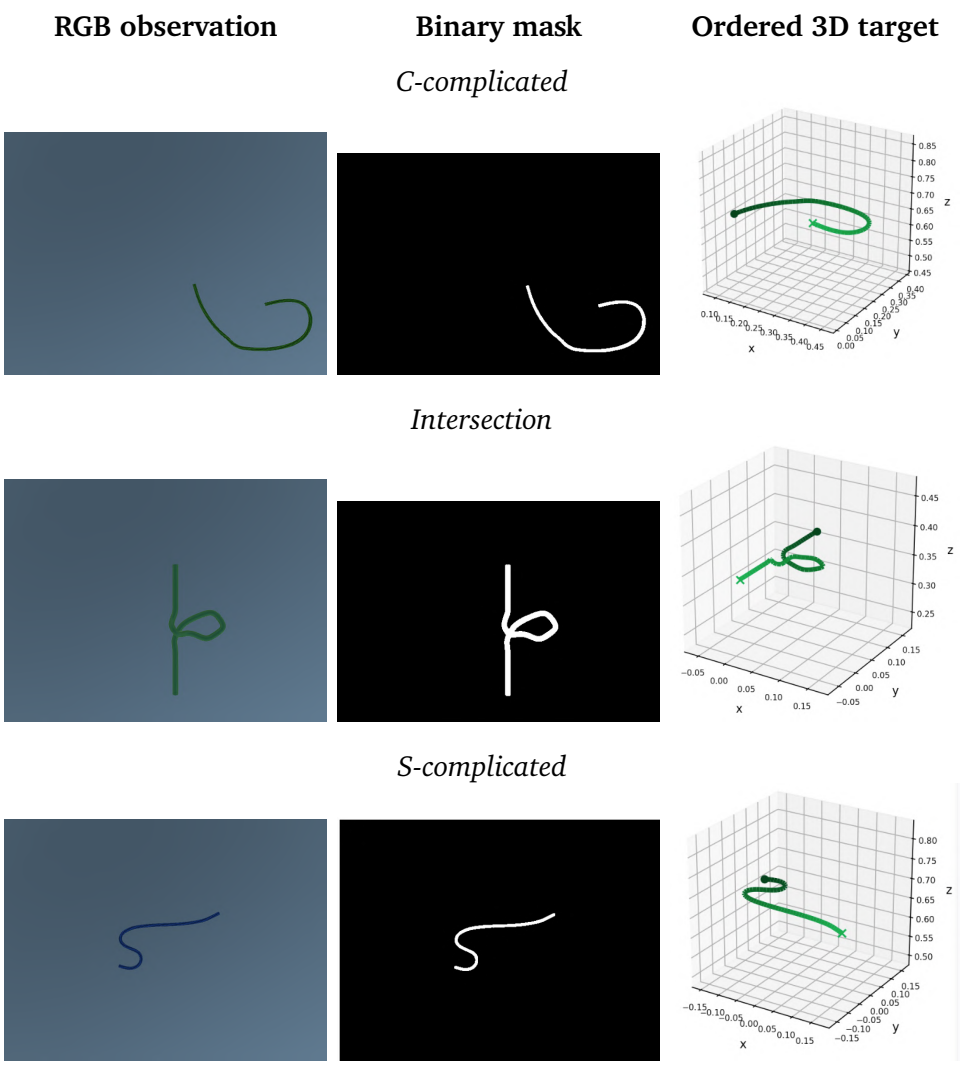



Figure 4.1: Representative samples from the synthetic validation split. The rows show a complicated C-shape, a self-intersecting configuration, and a complicated S-shape. The columns contain the RGB observation, the corresponding binary mask, and the complete ordered 3D ground-truth centerline. These examples visualize the observation and target modalities; the evaluation includes all nine scenario groups.

Table 4.1: Configuration overview for TrackDLO-init and the proposed estimator. Observation-derived intermediate quantities are listed as part of the corresponding pipeline rather than as additional sensor inputs.

| **Aspect** | **TrackDLO-init** | **Proposed estimator** |
|---|---|---|
| Current-frame data | RGB and depth | RGB, depth, and camera intrinsics |
| DLO mask construction | Material-specific HSV thresholding | SAM3-derived binary DLO mask |
| State construction | Skeleton and spline-based classical initialisation | Path-lifted prior followed by geometry-conditioned velocity-flow refinement |
| Ordered output | $64 \times 3$ camera-space centerline | $64 \times 3$ camera-space centerline |
| Temporal history | Not used by the initialisation procedure | Not used |
| Material-specific estimator settings | HSV thresholds configured for each material | No material-specific estimator parameters after mask extraction |
| Evaluation stochasticity | Deterministic, one run per input | Four fixed Gaussian source draws per input |

image-processing pipeline. The parameters of the publicly released open-source TrackDLO implementation are retained, except for the material-specific HSV thresholds, which are configured for the five evaluated appearances.

The proposed method uses a SAM3-derived binary DLO mask [6], the current depth observation, and the camera intrinsics. Both $P_{\text{obs}}$ and the path-lifted prior are constructed from this same current observation. The prior is therefore an internal geometric condition rather than additional information from another frame or from the training set. Both methods estimate a 64-particle ordered centerline from one RGB-D frame and receive no state propagated from a previous frame. Table 4.1 summarizes the comparison.

### 4.1.3 Training and Inference Setup

The model has 5.82 million trainable parameters. Training uses AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and zero weight decay. The initial learning rate is $3 \times 10^{-4}$. A cosine schedule decreases it to $3 \times 10^{-6}$ over 20,000 optimizer steps. The batch size is 64, and the final checkpoint at step 20,000 is used for the reported evaluation. Training is performed on one NVIDIA A100-SXM4 GPU with 40 GB of memory.

The active objective is the velocity loss from Equation (3.17), together with the spacing and bending terms from Equations (3.21) and (3.24). The reported training objective contains only these three terms. The velocity term has unit weight, while $\lambda_{\text{spacing}} = 0.02$ and $\lambda_{\text{bending}} = 0.01$.

To expose the model to orientation changes without altering the relation between the target and its observation conditions, the geometric evidence is augmented jointly. For each sampled training record, the rotation augmentation is applied with probability $0.75$. When it is applied, an angle is drawn uniformly over a complete rotation around the camera optical axis. The ground-truth particles, path-lifted prior, and $P_{\text{obs}}$ are rotated by the same 3D rotation $\mathbf{R}$ around the camera $z$-axis. The SAM3 mask is transformed consistently with the homography $\mathbf{H} = \mathbf{K}\mathbf{R}\mathbf{K}^{-1}$. This augmentation changes the in-plane orientation while preserving the geometric relation between the target and all observation conditions. Intersection records are sampled with twice the weight of the other scenario groups.

Unless stated otherwise, inference uses $K = 10$ Euler steps and the final 20,000-step checkpoint. The learned estimator is evaluated with four fixed and reproducible Gaussian source draws per input. Reported model errors are averaged over these draws. TrackDLO-init is deterministic and is therefore executed once per evaluated input. Sampler length and model variants are varied only in the corresponding ablation studies.

### 4.1.4 Evaluation Metrics and Validity Criteria

Let $\hat{\mathbf{X}} = (\hat{\mathbf{X}}_1, \ldots, \hat{\mathbf{X}}_N)$ be an estimated ordered centerline and $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$ its ground-truth counterpart. The mean index error is

$$E_{\text{idx}}(\hat{\mathbf{X}}, \mathbf{X}) = \frac{1}{N} \sum_{i=1}^{N} \left\| \hat{\mathbf{X}}_i - \mathbf{X}_i \right\|_2 . \tag{4.1}$$

It measures pointwise 3D accuracy under the ordered particle correspondence.

The discrete Fréchet distance measures the discrepancy between two ordered curves [41]. Let $\Gamma$ be the set of monotone couplings between their ordered nodes. The metric is

$$d_{\mathrm{F}}(\hat{\mathbf{X}}, \mathbf{X}) = \min_{\gamma \in \Gamma} \max_{(i,j) \in \gamma} \left\| \hat{\mathbf{X}}_i - \mathbf{X}_j \right\|_2 . \tag{4.2}$$

Unlike the index error, this metric permits a monotone correspondence along the two curves while preserving their traversal order. Both metrics are evaluated in camera coordinates and reported in centimeters.

The proposed estimator uses the deterministic endpoint convention defined in Section 3.3. TrackDLO-init does not assign a semantic start endpoint. The TrackDLO-init output is therefore evaluated in both the published and reversed node orders. For each metric $E$, the baseline score is

$$E_{\text{TrackDLO-init}} = \min \left( E(\hat{\mathbf{X}}, \mathbf{X}), E(\mathrm{rev}(\hat{\mathbf{X}}), \mathbf{X}) \right) . \tag{4.3}$$

This direction selection is applied only to TrackDLO-init. Equations (4.1)–(4.3) define the two reconstruction metrics and the baseline direction treatment.

A TrackDLO-init trial is marked as *failed* if the evaluation process receives no message on the initialisation output topic. If a $64 \times 3$ result is published, it is accepted only when all coordinates are finite, all depths are positive, every projected node lies inside the image, and the mean projected distance to the nearest HSV-mask pixel does not exceed three pixels. A published result that violates any of these checks is marked as an *output failing validity checks*. Geometric errors are compared only on samples for which both methods provide a valid output. The valid-output coverage is reported separately so that failed baseline cases are not hidden by the paired error averages.

## 4.2 Synthetic Benchmark and TrackDLO-init Comparison

The principal external comparison asks whether the proposed learned estimator improves single-frame initialisation over a classical geometric pipeline as the DLO topology becomes more difficult. Two properties must be considered jointly: the accuracy of an estimate when a method returns a valid centerline, and the fraction of configurations for which such an estimate is available. Separating these properties is important because a low conditional error can otherwise conceal frequent initialisation failures.

### 4.2.1 Comparison Protocol

The comparison is conducted at the sequence level on all 59 validation sequences. One representative frame is selected from each sequence according to the middle-frame policy defined in Section 4.1.1. Because the 100 frames of a sequence share the same stored ground-truth geometry, this protocol assigns equal weight to each distinct validation configuration rather than weighting a configuration by its repeated frames.

TrackDLO-init is deterministic and is run once for every selected input. The proposed estimator uses $K = 10$ Euler steps and four fixed Gaussian source draws for each input. The proposed estimator's per-sequence error is first averaged over these four draws. The validity criteria and endpoint-direction treatment are those defined in Section 4.1.4.

The availability of a valid output and the geometric accuracy of that output are reported separately. TrackDLO-init produces a valid result for 31 of the 59 sequences. Accordingly, the paired error comparison contains exactly these 31 inputs and uses the same ground truth for both methods. The 24 trials without a published result and the four published outputs that fail the validity checks do not enter either paired error mean. They remain visible through the separately reported coverage of $31/59 = 52.5\%$.

### 4.2.2 Quantitative and Qualitative Results

Table 4.2 reports output availability and paired reconstruction errors by scenario. The proposed estimator attains a lower mean index error on 29 of the 31 paired inputs ($93.5\%$) and a lower discrete Fréchet distance on 28 ($90.3\%$). Over the paired subset, its mean index error is $2.061$ cm, compared with $5.911$ cm for TrackDLO-init, while the corresponding discrete Fréchet distances are $4.599$ cm and $11.365$ cm. These paired means characterize accuracy conditional on both methods returning a valid result; they do not incorporate the 28 invalid or missing TrackDLO-init outputs.

Figure 4.2 visualizes the same paired means and annotates the TrackDLO-init coverage of every scenario. No error bar is shown for the complicated straight scenario because TrackDLO-init does not return a valid output for either sequence. The scenario means with one or two valid pairs should be interpreted together with their displayed coverage rather than as stable estimates over the full scenario group.

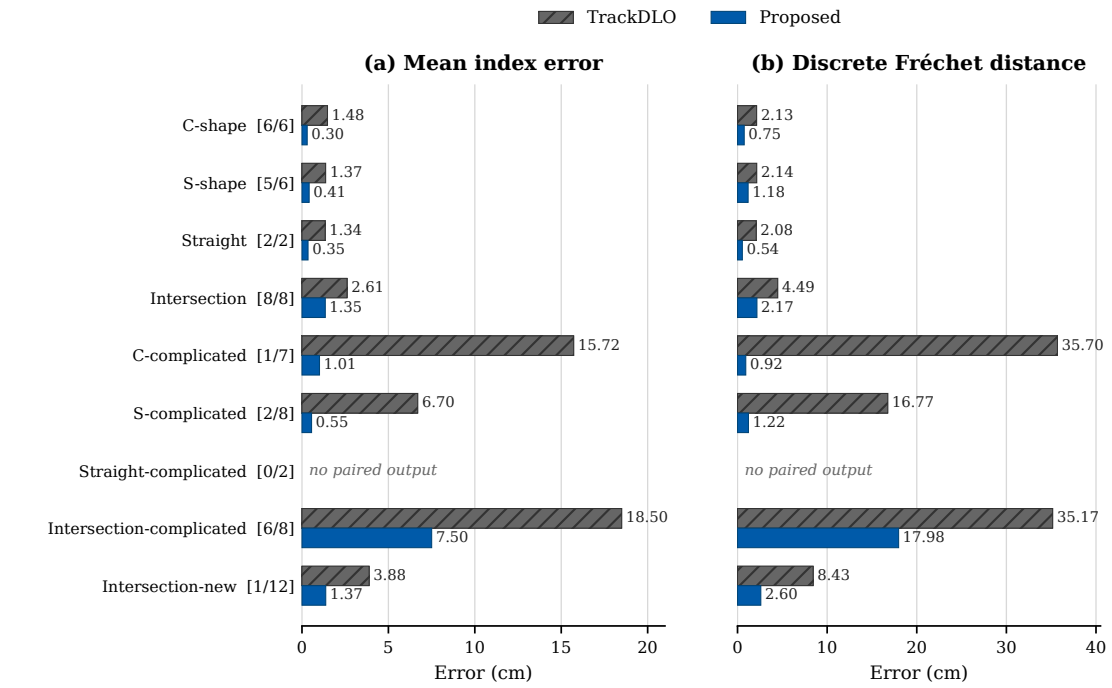
Figure 4.3 complements the aggregate metrics with six fixed examples. The first two rows show regular intersections for which the proposed estimates remain close to the ordered ground truth. The remaining rows include complicated C- and S-shapes and two

Table 4.2: Sequence-level comparison with TrackDLO-init by validation scenario. "No output" denotes trials for which no initialisation message is published; "invalid" denotes published results that fail the validity checks from Section 4.1.4. Error columns are means over the paired valid subset in each row. The overall errors therefore use 31 paired sequences, while coverage is $31/59 = 52.5\%$.

| | | TrackDLO-init output | | | Index error (cm) | | Fréchet (cm) | |
|---|---|---|---|---|---|---|---|---|
| Scenario | $n$ | Valid | No output | Invalid | TrackDLO-init | Ours | TrackDLO-init | Ours |
| C-shape | 6 | 6 | 0 | 0 | 1.475 | 0.301 | 2.133 | 0.746 |
| S-shape | 6 | 5 | 1 | 0 | 1.369 | 0.415 | 2.144 | 1.181 |
| Straight | 2 | 2 | 0 | 0 | 1.342 | 0.350 | 2.081 | 0.537 |
| Intersection | 8 | 8 | 0 | 0 | 2.606 | 1.348 | 4.490 | 2.165 |
| C-complicated | 7 | 1 | 6 | 0 | 15.724 | **1.013** | 35.699 | **0.919** |
| S-complicated | 8 | 2 | 6 | 0 | 6.700 | **0.554** | 16.774 | **1.220** |
| Straight-complicated | 2 | 0 | 2 | 0 | – | – | – | – |
| Intersection-complicated | 8 | 6 | 0 | 2 | 18.501 | 7.503 | 35.170 | 17.975 |
| Intersection-new | 12 | 1 | 9 | 2 | 3.883 | 1.374 | 8.434 | 2.597 |
| Overall | 59 | 31 | 24 | 4 | 5.911 | **2.061** | 11.365 | **4.599** |

multi-loop intersections, thereby retaining both favorable and difficult outcomes. For stochastic predictions, all four fixed source draws are shown whenever they are available in the recorded evaluation output; the C- and S-shaped examples contain the single recorded draw.

Taken together, the quantitative and qualitative evidence shows that the proposed estimator improves the conditional reconstruction accuracy of single-frame initialisation while retaining estimates for configurations in which TrackDLO-init is frequently unavailable. The advantage is clearest in the complicated C- and S-shaped groups, where the baseline coverage decreases sharply. However, the result does not imply that complete ordered reconstruction is solved uniformly. The proposed errors remain large on several complicated intersections, and the qualitative examples reveal variability across stochastic draws. The comparison therefore supports improved accuracy and coverage relative to the selected classical baseline, while identifying multi-loop and strongly self-intersecting geometry as a remaining failure regime.




Brackets report valid TrackDLO outputs / evaluated sequences.

Figure 4.2: Per-scenario paired comparison with TrackDLO-init. The left panel shows the mean index error and the right panel the discrete Fréchet distance. Bracketed counts give the number of valid TrackDLO-init outputs relative to the number of evaluated sequences. Each error pair is computed on the same valid inputs; missing and invalid TrackDLO-init outputs are excluded from the bars and represented by coverage.

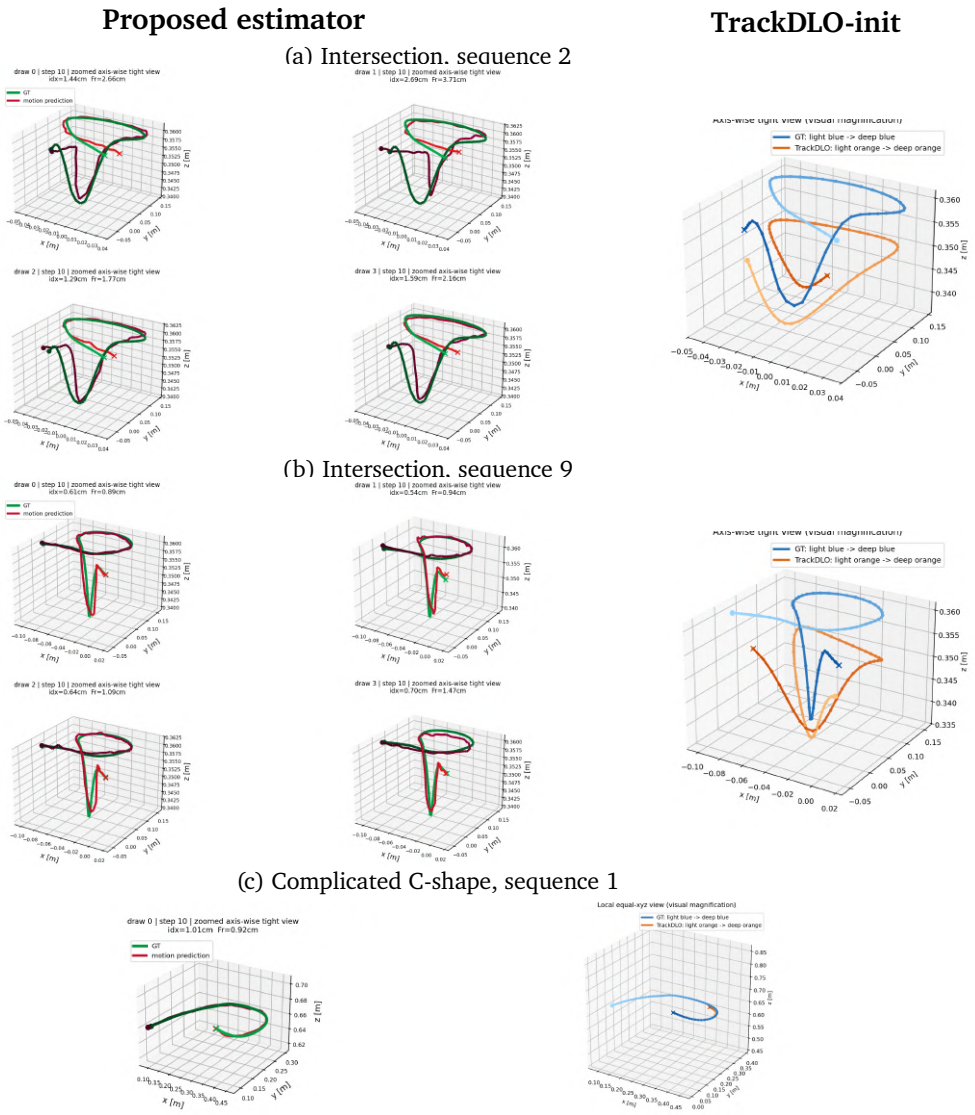



Figure 4.3: Paired qualitative comparison on six validation sequences (first part). The proposed-method column displays all fixed draws contained in the evaluation sheet, while TrackDLO-init has one deterministic output. Both columns use their magnified three-dimensional view and are displayed at equal height within each row. Green and blue curves denote ground truth in the proposed and TrackDLO-init panels, respectively; red and orange curves denote the corresponding estimates.

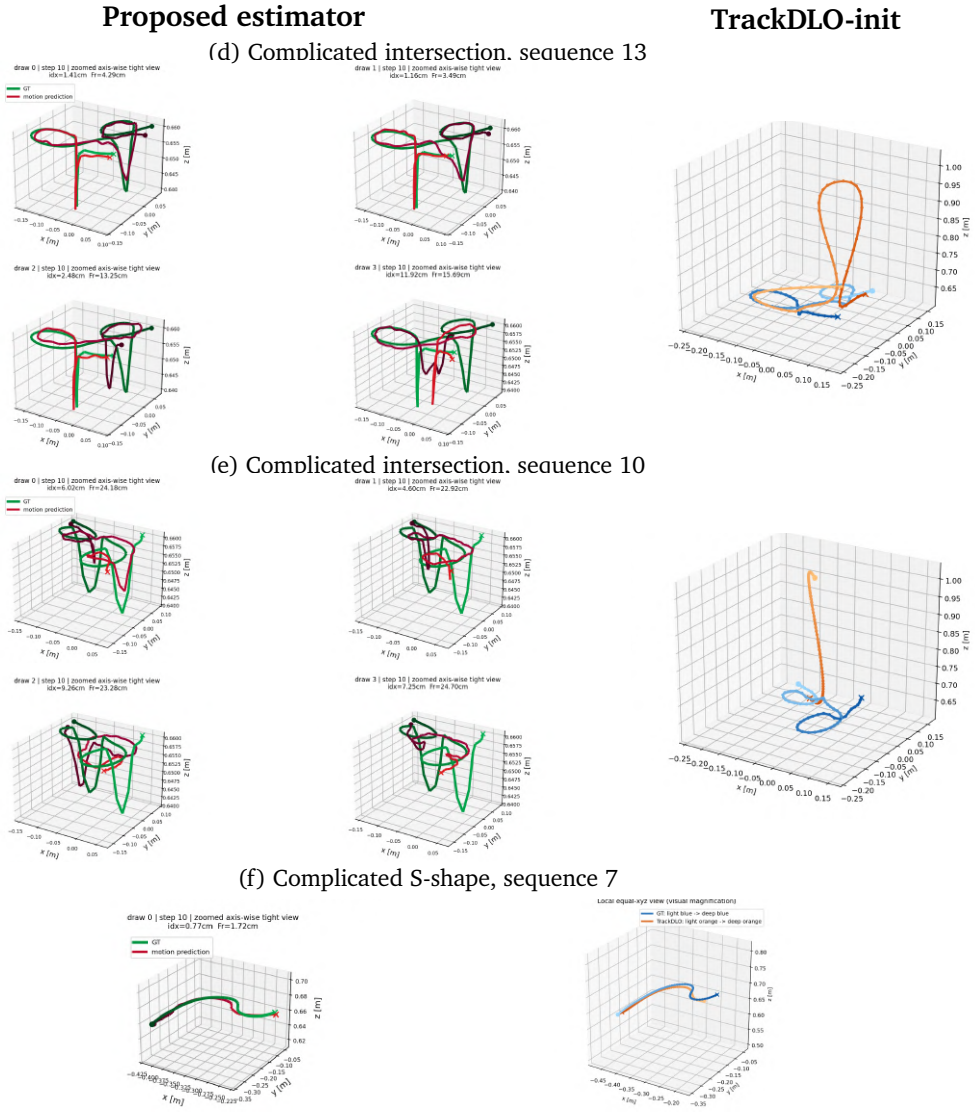



Figure 4.3: Paired qualitative comparison on six validation sequences (continued). The complicated examples show that multi-loop geometry and large changes in local curve orientation remain challenging and that performance can vary across the four stochastic draws.

## 4.3 Official Failure-Case Recordings

The synthetic benchmark measures reconstruction against complete ground truth, but it cannot reveal how a synthetic-trained model behaves under real sensor noise and domain shift. The official TrackDLO failure-case recordings provide a complementary test: they contain precisely the limited depth, rapid motion, and endpoint visibility conditions that challenge single-frame initialisation. Since these recordings do not include ordered three-dimensional ground truth, the purpose of this experiment is not to rank reconstruction accuracy. Instead, it examines output availability, agreement with current image evidence, and the role of the plausibility-based fallback when the learned refinement is applied outside its training distribution.

### 4.3.1 Evaluation Protocol

The evaluation uses the three failure-case recordings released with TrackDLO [42], corresponding to limited depth resolution, fast tip motion, and motion without visible tips. Output availability and consistency with the SAM3 mask are reported for each recording and complemented by qualitative overlays.

TrackDLO-init is evaluated deterministically on 20 frames from each recording. For the proposed system, four fixed Gaussian sources are evaluated for every frame on which the observation pipeline can construct a path-lifted prior. All 20 frames are eligible in the first two scenarios, yielding 80 trials per scenario. In the third scenario, frames 11–15 contain no usable path segment from the upstream geometric trace extraction ($K = 0$). Since a path-lifted prior cannot be constructed, these five frames are excluded rather than being assigned an artificial zero-valued prior. The third scenario therefore contains 15 eligible frames and 60 trials. In total, the fallback evaluation comprises 55 of the 60 source frames and 220 frame–draw trials.

The TrackDLO-init and proposed-system counts use different denominators and are not treated as a forced frame-by-frame pairing. For TrackDLO-init, a valid output must satisfy the criteria from Section 4.1.4. For the proposed system, *eligible-frame coverage* records whether the path-lifted prior can be constructed, while *returned-trial coverage* records whether the flow-and-fallback stage returns an output after that prerequisite is met. This distinction prevents the conditional coverage of the fallback mechanism from obscuring failures in its upstream prior construction.

The SAM3-mask-consistency score is computed as in the plausibility check from Section 3.6. The ordered centerline is projected to the image, adjacent projections are densified at one-pixel intervals, and the fraction of samples lying on the four-pixel-dilated SAM3 mask is measured. This score tests agreement with visible image evidence only. It is not a substitute for three-dimensional reconstruction error.

### 4.3.2 Qualitative Results and Fallback Behaviour

Table 4.3 summarizes output availability and mask consistency. TrackDLO-init returns a valid output on 19 of 20 frames in each of the first two recordings. In the third recording, 13 outputs are valid, six published outputs fail the validity checks, and one frame produces no output. These counts give an aggregate valid-output coverage of $51/60 = 85.0\%$.

The proposed pipeline constructs its path-lifted prior on 55 of the 60 source frames. None of the 220 raw flow predictions passes the plausibility gate. The system therefore returns the path-lifted prior in every eligible trial, giving a conditional returned-trial coverage of $220/220 = 100\%$. This result demonstrates the operational role of the fallback, but it must not be read as 100% end-to-end coverage of the original recordings: five frames in the third scenario remain ineligible because no prior can be constructed.

For valid TrackDLO-init outputs, the mean mask-consistency scores range from $90.9\%$ to $96.8\%$. The returned path-lifted priors range from $96.2\%$ to $100.0\%$, whereas the rejected raw flow predictions obtain substantially lower agreement. The similar TrackDLO-init and fallback scores do not establish comparable three-dimensional accuracy.

The rejection of all raw flow predictions is consistent with a pronounced simulation-to-reality gap in the learned component. The velocity model is trained exclusively on synthetic observations, whereas these recordings contain real RGB-D measurements outside its training distribution. In contrast, the path-lifted prior is reconstructed from the current observation through explicit geometric operations and does not rely on the synthetic-trained flow mapping. The substantially higher mask consistency of the path-lifted prior is therefore compatible with the gate rejecting the learned predictions and the system returning the prior in every eligible trial. This observation indicates domain sensitivity of the learned refinement rather than proving that the prior is universally more accurate. Indeed, five frames in the third recording already fail during upstream prior construction.

Figure 4.4 shows one fixed frame from each recording. In the left column, the gray curve is the path-lifted prior and the four colored curves are the raw $K = 10$ flow predictions. Because none of these predictions passes the gate, the gray prior is the output returned

Table 4.3: Behaviour on the three official TrackDLO failure-case recordings. Panel (a) reports output availability. The proposed-system trial count includes four fixed Gaussian draws for every eligible frame; its returned count is therefore conditional on successful path-lifted-prior construction. Panel (b) reports agreement with the SAM3 mask. These percentages are two-dimensional observation-consistency scores, not three-dimensional reconstruction errors.

**(a) Output availability**

| | TrackDLO-init | | | | | Proposed system | |
|---|---|---|---|---|---|---|---|
| Scenario | Frames | Valid | Invalid | No output | Valid coverage | Eligible frames | Returned trials |
| Limited depth resolution | 20 | 19 | 0 | 1 | 95.0% | 20/20 | 80/80 |
| Fast tip motion | 20 | 19 | 0 | 1 | 95.0% | 20/20 | 80/80 |
| Motion without visible tips | 20 | 13 | 6 | 1 | 65.0% | 15/20 | 60/60 |
| Overall | 60 | 51 | 6 | 3 | 85.0% | 55/60 | 220/220 |

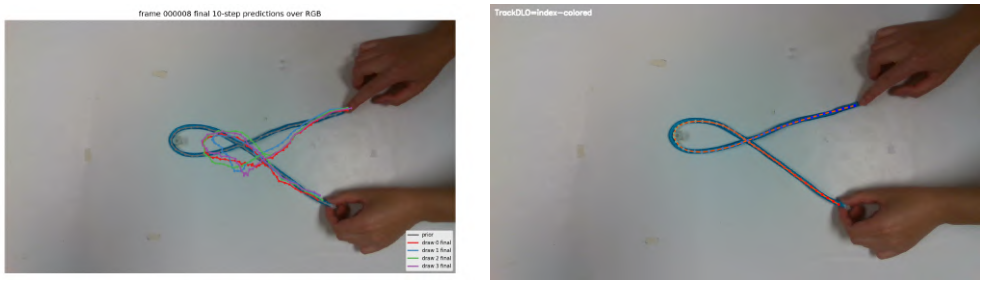
**(b) Mask consistency**

| Scenario | TrackDLO-init valid output | Raw flow prediction | Returned prior |
|---|---|---|---|
| Limited depth resolution | 96.1% | 28.7% | 99.4% |
| Fast tip motion | 96.8% | 28.3% | 96.2% |
| Motion without visible tips | 90.9% | 38.8% | 100.0% |
| Flow predictions accepted | – | 0/220 | – |

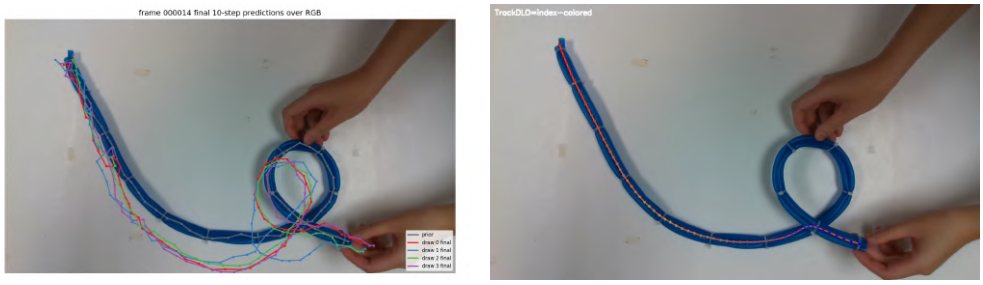
by the complete system. The right column shows the deterministic TrackDLO-init output on the same frame. The overlays expose the failure of the learned predictions and the corrective role of fallback without implying three-dimensional accuracy in the absence of ground truth.

**Prior and four flow predictions** **TrackDLO-init**

(a) Limited depth resolution, frame 8




(b) Fast tip motion, frame 14




(c) Motion without visible tips, frame 7

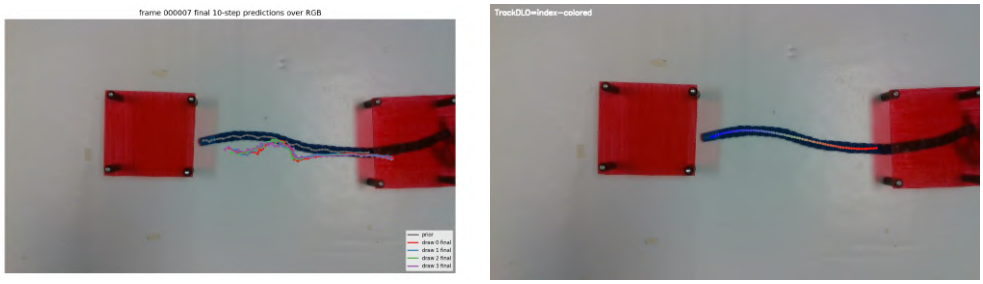



Figure 4.4: Qualitative overlays from the three official failure-case recordings. The left panels contain the path-lifted prior in gray and four raw flow predictions. All displayed flow predictions are rejected, so the system returns the gray prior. The right panels show TrackDLO-init on the identical frame. Since the recordings contain no ordered three-dimensional ground truth, the figure illustrates observation consistency and fallback behaviour rather than reconstruction accuracy.

## 4.4 Ablation Studies

The external evaluations establish how the complete system behaves, but do not identify which design choices produce that behaviour. The following controlled comparisons therefore isolate the conditioning representation, prediction parameterization, and local depth residual in direct correspondence with RQ1–RQ3. A final analysis evaluates whether the plausibility gate improves the returned state when ground truth is available. Unless stated otherwise, the ablations use the same 59 sequence-level validation inputs, four reproducible Gaussian sources per stochastic model, and the $K = 10$ headline sampler defined in Section 4.1.3. The gate analysis uses an independently generated set of 236 stochastic frame–draw predictions. The raw-prediction statistics for the gate analysis are therefore reported from the exact rows to which the gate is applied rather than copied from another ablation table.

### 4.4.1 Geometric Conditioning

RQ1 concerns how the coarse ordered prior and visible three-dimensional evidence contribute to the final reconstruction. It is examined by comparing three levels of geometric conditioning. *Prior alone* directly evaluates the path-lifted prior and does not invoke a learned model. *Flow without* $P_{\text{obs}}$ retains the ordered prior and local image-derived conditioning but removes the cross-attended unordered three-dimensional observation set. *Full model* uses both the ordered prior and $P_{\text{obs}}$. All three configurations are evaluated on exactly the same representative frame from each of the 59 validation sequences.

Table 4.4 shows that learned refinement reduces the overall index error from $5.039$ cm for the prior to $3.363$ cm without $P_{\text{obs}}$ and $3.102$ cm for the full model. The corresponding Fréchet distances do not improve monotonically: $6.615$ cm, $6.506$ cm, and $6.749$ cm, respectively. Thus, adding visible three-dimensional evidence improves the aggregate ordered particle correspondence but does not uniformly improve curve-level accuracy.

The scenario-level results are similarly mixed. The full model yields the lowest index error on the standard C-shape, S-shape, intersection, and complicated-intersection groups, whereas removing $P_{\text{obs}}$ is preferable for several other groups. Most notably, the complicated-straight prior has an index error of only $0.640$ cm, but both learned variants deteriorate sharply to $16.333$ cm and $14.180$ cm. This contrast demonstrates that learned refinement can substantially degrade an already accurate prior in a difficult configuration, motivating the plausibility gate evaluated later in this section.

Table 4.4: RQ1 conditioning ablation on the common 59-sequence validation set. Every result is reported as mean index error / discrete Fréchet distance in centimeters. Prior alone directly evaluates the path-lifted prior; the two learned variants use $K = 10$ Euler steps. The overall row is computed across all 59 sequences rather than by averaging the scenario means.

| Scenario | $n$ | Prior alone | Flow without $P_{\text{obs}}$ | Full model |
|---|---|---|---|---|
| C-shape | 6 | 0.926 / 1.281 | 0.252 / 0.501 | 0.222 / 0.463 |
| S-shape | 6 | 0.906 / 1.179 | 0.287 / 0.508 | 0.262 / 0.526 |
| Straight | 2 | 0.937 / 1.237 | 0.259 / 0.382 | 0.286 / 0.381 |
| Intersection | 8 | 6.615 / 8.011 | 2.013 / 3.413 | 1.474 / 2.092 |
| C-complicated | 7 | 0.607 / 1.012 | 0.426 / 0.760 | 0.472 / 1.043 |
| S-complicated | 8 | 0.457 / 0.948 | 0.469 / 0.849 | 0.604 / 1.385 |
| Straight-complicated | 2 | 0.640 / 0.967 | 16.333 / 32.124 | 14.180 / 27.751 |
| Intersection-complicated | 8 | 12.982 / 17.027 | 9.245 / 17.773 | 8.015 / 17.793 |
| Intersection-new | 12 | 9.873 / 13.014 | 5.433 / 10.934 | 5.596 / 13.209 |
| Overall | 59 | 5.039 / 6.615 | 3.363 / **6.506** | **3.102** / 6.749 |

### 4.4.2 Prediction Parameterization

RQ2 concerns how the choice between velocity and direct clean-state prediction affects reconstruction across multiple refinement steps. The velocity parameterization used by the proposed model is therefore compared with direct clean-state prediction. The velocity model estimates the vector field integrated by the Euler sampler. The direct model instead predicts a clean state $\hat{\mathbf{X}}_1$ at each call, from which the sampler constructs an update direction as described in Section 3.6. Both models are evaluated with the same fixed sources at 1, 4, 10, and 20 sampler steps.

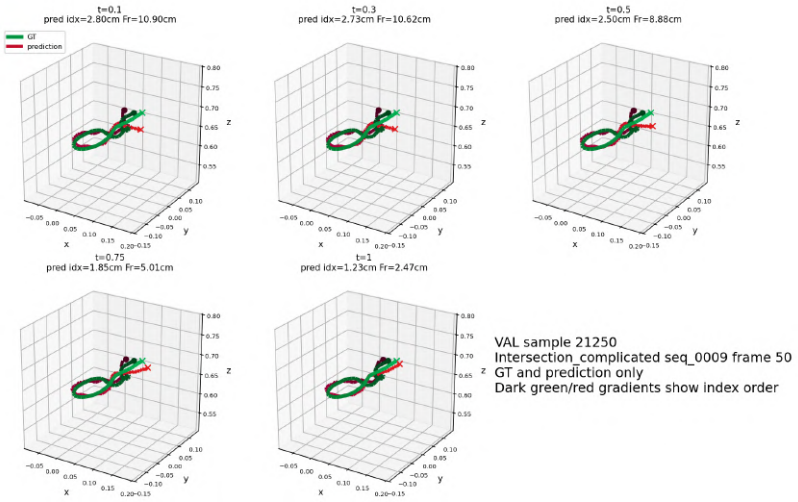
As reported in Table 4.5, the velocity model improves from $3.325$ cm at one step to $3.102$ cm at ten steps and then remains stable. The direct model changes only from $3.567$ cm to $3.530$ cm over the same range and reaches $3.525$ cm at 20 steps. The direct model's nearly step-invariant error indicates only weak dependence of the predicted clean state on the intermediate noisy state and sampling time. Consequently, repeated updates largely

Table 4.5: RQ2 comparison of velocity and direct-state prediction. The 1-, 4-, 10-, and 20-step columns report mean index error in centimeters; the final column reports the discrete Fréchet distance at the ten-step headline setting.

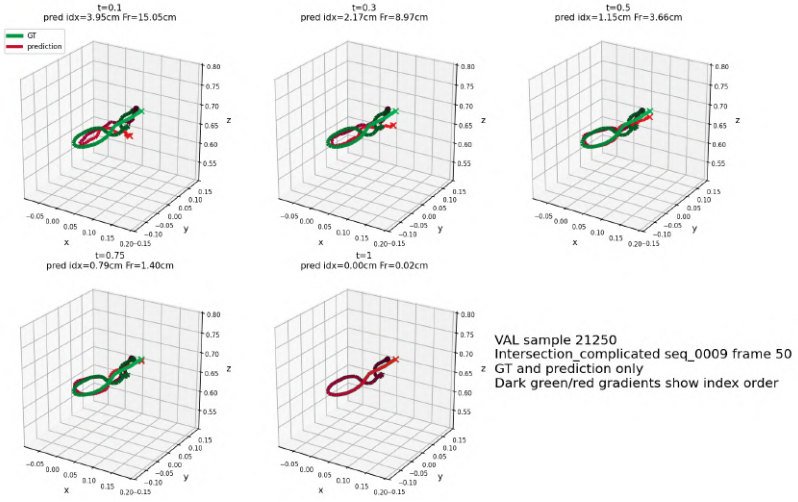
| Parameterization | 1 step | 4 steps | 10 steps | 20 steps | 10-step Fréchet |
|---|---|---|---|---|---|
| Velocity | 3.325 | 3.114 | 3.102 | 3.104 | 6.749 |
| Direct state | 3.567 | 3.532 | 3.530 | 3.525 | 6.924 |

point toward a similar endpoint and provide little additional refinement. At the ten-step headline setting, velocity prediction also obtains the lower Fréchet distance, $6.749$ cm compared with $6.924$ cm.

Figure 4.5 examines this behaviour on a complicated-intersection input and a complicated S-shape input while varying the flow time $t$. The interpolated state $\mathbf{x}_t$ becomes progressively closer to the clean state as $t$ increases. Nevertheless, the direct model produces nearly the same global curve structure throughout the sweep; its changes are concentrated mainly around local alignment and the endpoints. The velocity model exhibits a stronger time-dependent change in the reconstructed state. Together with the sampler-step comparison, these examples support the interpretation that the direct model relies only weakly on the intermediate flow state and therefore behaves approximately as a single-output regressor. The figure provides illustrative examples rather than an additional aggregate metric.

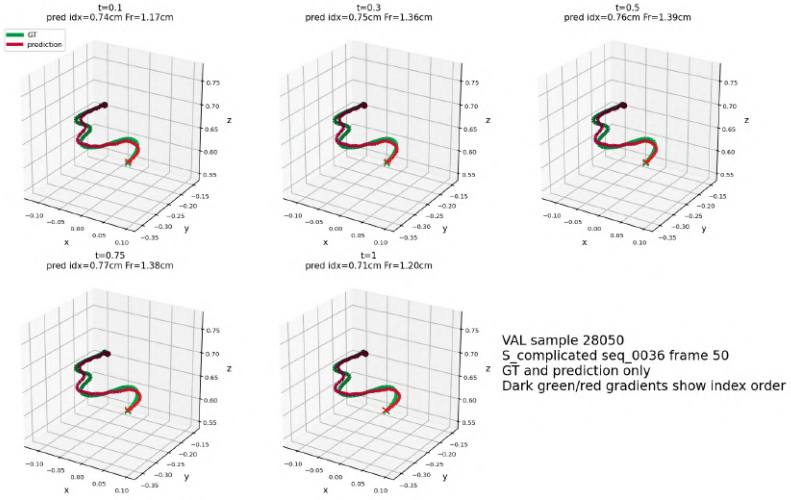



(a) Direct-state prediction.

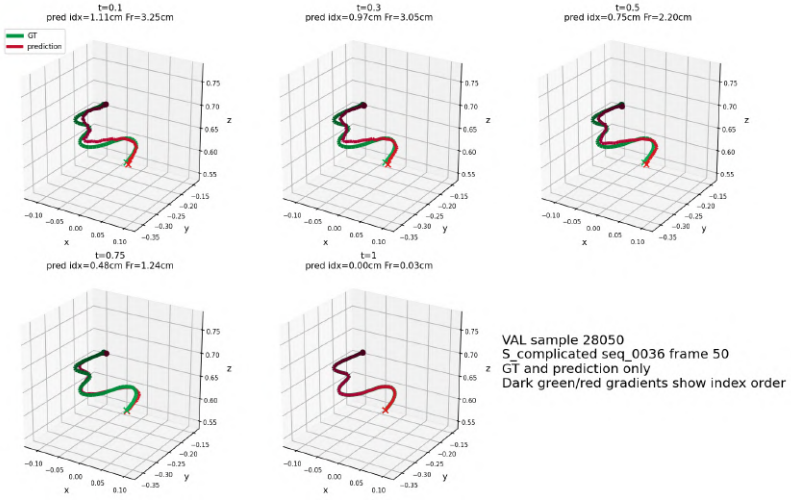



(b) Velocity prediction with clean-state reconstruction.

Figure 4.5: Time sweep for a complicated-intersection validation sample (first part). Predictions are compared with the ground truth at $t \in \{0.1, 0.3, 0.5, 0.75, 1.0\}$. Direct-state outputs preserve a similar global shape, whereas the velocity reconstruction changes more strongly. The velocity reconstruction at $t = 1$ coincides with $\mathbf{x}_1$ by construction and is not an independent endpoint prediction.




(c) Direct-state prediction.




(d) Velocity prediction with clean-state reconstruction.

Figure 4.5: Time sweep for a complicated S-shape validation sample (continued). Predictions are compared with the ground truth at $t \in \{0.1, 0.3, 0.5, 0.75, 1.0\}$. Direct-state outputs preserve a similar global shape, whereas the velocity reconstruction changes more strongly. The velocity reconstruction at $t = 1$ coincides with $\mathbf{x}_1$ by construction and is not an independent endpoint prediction.

### 4.4.3 Local Depth-Residual Conditioning

RQ3 concerns how node-wise local depth information affects particle-level and curve-level reconstruction accuracy when the visible observation is incomplete. This question is tested by adding an explicit local depth discrepancy to the state-dependent image projection. The baseline uses the six-channel mask-only projection, while the depth-conditioned model uses the mutually exclusive eight-channel configuration defined in Section 3.4. The latter appends the normalized difference between the observed depth and the projected particle depth, together with its validity indicator. Both configurations retain the same velocity parameterization, ordered prior, unordered observation set, training protocol, and $K = 10$ evaluation setting.

Table 4.6 reveals a trade-off between the two error measures. Across all 59 validation sequences, depth-residual conditioning increases the index error from $3.102$ cm to $3.283$ cm, while reducing the discrete Fréchet distance from $6.749$ cm to $6.477$ cm. The added signal therefore improves aggregate curve-level proximity without improving ordered particle correspondence.

The effect is strongly configuration dependent. On the complicated-straight group, depth conditioning reduces the index error from $14.180$ cm to $12.359$ cm and the Fréchet distance from $27.751$ cm to $24.185$ cm. Conversely, both errors increase substantially on the complicated C-shape. Several intersection groups show the same metric-dependent behaviour as the overall result: one measure improves while the other deteriorates. Consequently, the experiment does not support a uniform advantage for local depth residuals; rather, the additional cue changes how error is distributed along the reconstructed curve.

Table 4.6: RQ3 comparison of mask-only local conditioning and conditioning with a local depth residual. Each entry reports mean index error / discrete Fréchet distance in centimeters at $K = 10$. The overall row is computed across all 59 validation sequences.

| Scenario | $n$ | Mask only | Mask + depth residual |
|---|---|---|---|
| C-shape | 6 | 0.222 / 0.463 | 0.213 / 0.471 |
| S-shape | 6 | 0.262 / 0.526 | 0.261 / 0.496 |
| Straight | 2 | 0.286 / 0.381 | 0.265 / 0.364 |
| Intersection | 8 | 1.474 / 2.092 | 1.267 / 2.231 |
| C-complicated | 7 | 0.472 / 1.043 | 1.186 / 1.986 |
| S-complicated | 8 | 0.604 / 1.385 | 0.604 / 1.187 |
| Straight-complicated | 2 | 14.180 / 27.751 | 12.359 / 24.185 |
| Intersection-complicated | 8 | 8.015 / 17.793 | 9.052 / 17.680 |
| Intersection-new | 12 | 5.596 / 13.209 | 5.827 / 12.046 |
| Overall | 59 | **3.102** / 6.749 | 3.283 / **6.477** |

### 4.4.4 Plausibility Gate

The plausibility gate is intended to prevent geometrically implausible learned outputs from replacing a usable prior. Synthetic validation data permit this intended safety function to be tested against three-dimensional ground truth. It uses the same 59 representative frames, four fixed Gaussian draws per frame, and $K = 10$ velocity predictions, yielding 236 frame–draw trials. An accepted trial returns the raw flow prediction; a rejected trial returns the path-lifted prior. The resulting system output is denoted *hybrid* in Table 4.7. The acceptance decision itself remains independent of ground truth and uses only the hard geometric and mask-consistency conditions from Section 3.6.

The gate accepts 145 of the 236 predictions, corresponding to $61.4\%$. The plausibility gate's decisions are strongly associated with scene complexity. Across the six non-intersection groups, 115 of 124 predictions ($92.7\%$) are accepted. Across the three intersection groups, only 30 of 112 predictions ($26.8\%$) are accepted, so the path-lifted prior supplies most returned states in these cases.

Table 4.7: Synthetic-validation analysis of the plausibility gate. The count $n$ denotes frame–draw trials. Acceptance is the fraction for which the raw prediction is returned; otherwise the path-lifted prior is returned. Prediction, prior, and hybrid columns report mean index error in centimeters, not discrete Fréchet distance. The prediction column is computed from the 236 stochastic rows used by this gate evaluation; it is not copied from the independently generated predictions in Tables 4.4 and 4.6.

| | | | Mean index error [cm] | | |
|---|---|---|---|---|---|
| Scenario | $n$ | Acceptance | Prediction | Prior | Hybrid |
| C-shape | 24 | 100.0% | 0.212 | 0.926 | 0.212 |
| S-shape | 24 | 100.0% | 0.275 | 0.906 | 0.275 |
| Straight | 8 | 100.0% | 0.282 | 0.937 | 0.282 |
| C-complicated | 28 | 100.0% | 0.494 | 0.607 | 0.494 |
| S-complicated | 32 | 84.4% | 0.550 | 0.457 | 0.483 |
| Straight-complicated | 8 | 50.0% | 14.222 | 0.640 | 6.490 |
| Intersection | 32 | 62.5% | 1.455 | 6.615 | 3.147 |
| Intersection-complicated | 32 | 9.4% | 7.751 | 12.982 | 12.416 |
| Intersection-new | 48 | 14.6% | 5.570 | 9.873 | 8.863 |
| Overall | 236 | 61.4% | 3.055 | 5.039 | 4.316 |

This selectivity does not imply that the gate minimizes reconstruction error. The raw predictions obtain an overall index error of $3.055$ cm, compared with $5.039$ cm for the prior and $4.316$ cm for the hybrid output. Fallback substantially reduces the complicated-straight error from $14.222$ cm to $6.490$ cm, demonstrating its ability to mitigate a severe learned-model failure. In contrast, it increases the error in all three intersection groups because the prior is less accurate than the prediction there.

These results characterize the gate as a conservative plausibility mechanism, not an oracle for three-dimensional accuracy. A prediction may satisfy the length, continuity, and projected-mask tests while retaining incorrect depth or particle correspondence; conversely, a rejected prediction may be closer to ground truth than its fallback. This distinction also

qualifies the real-recording results in Section 4.3.2: there, fallback maintains observation-consistent output under a simulation-to-reality shift, but the absence of three-dimensional ground truth precludes an accuracy claim.

## 4.5 Summary of Experimental Findings

The external comparison provides evidence at two complementary levels. On the 31 sequences for which both methods return valid estimates, the proposed estimator achieves lower paired index and Fréchet errors than TrackDLO-init on most inputs, while the baseline provides valid output for only 31 of the 59 evaluated configurations. At the same time, the remaining errors and qualitative variation on complicated intersections show that the learned method does not resolve all self-occluded topologies. The real recordings expose a different limitation: the synthetic-trained flow model does not transfer reliably to the observed sensor domain, and the system depends on its geometric fallback whenever a prior can be constructed.

The ablations clarify the contributions behind these system-level results. For RQ1, learned refinement reduces the overall index error relative to the path-lifted prior, and adding $P_{\text{obs}}$ further improves ordered particle accuracy, although the Fréchet results remain scenario dependent. For RQ2, velocity prediction shows a meaningful benefit from multi-step sampling, whereas direct-state prediction behaves approximately as a single-output regressor. For RQ3, the local depth residual improves aggregate curve-level proximity but does not provide a consistent reduction in ordered particle error.

Finally, the synthetic gate analysis shows that plausibility and three-dimensional accuracy are related but not interchangeable. Fallback mitigates a severe failure on complicated straight configurations, yet increases the overall index error because the prior is less accurate than the prediction on several intersections. The resulting evidence supports a geometry-conditioned velocity model as an effective refinement of a coarse observation-derived prior, while identifying complex self-intersection, simulation-to-reality transfer, and upstream prior availability as the principal remaining limitations.

# 5 Discussion and Conclusion

This chapter interprets the experimental findings in relation to the research questions from Section 1.3 and identifies the limits and implications of the proposed approach.

## 5.1 Discussion

The central question of this thesis is whether a complete ordered three-dimensional DLO centerline can be recovered from a single RGB-D observation without an accurate previous state. Compared with the initialisation procedure used in TrackDLO (hereafter TrackDLO-init), the proposed combination of an observation-derived prior and learned refinement improves valid-output coverage and conditional accuracy on the evaluated synthetic configurations. The evidence concerns single-frame initialisation, however, and the remaining errors on complicated self-intersections show that geometric ambiguity is not fully resolved.

The ablations clarify the roles of the main design choices. For RQ1, the prior supplies traversal structure while the learned model uses visible evidence to refine it, but neither is uniformly superior. For RQ2, the benefit from repeated Euler updates supports velocity prediction over direct-state regression. For RQ3, the mixed effect of the local depth residual does not establish a consistent improvement under partial visibility. The components therefore address distinct aspects of the reconstruction problem rather than forming a uniformly improving sequence.

The real failure-case recordings expose the principal limitation of the learned component. The learned component's predictions do not transfer reliably from synthetic training data to real RGB-D measurements, leaving the system dependent on geometric fallback. Fallback preserves consistency with visible evidence, but neither establishes three-dimensional

accuracy without ground truth nor helps when upstream prior construction fails. The plausibility gate is therefore a conservative safeguard rather than an estimator of reconstruction accuracy.

## 5.2 Limitations and Future Work

The principal limitation is the reliance on synthetic training data. The real recordings reveal that the learned refinement does not transfer reliably to measurements outside this domain. Future work should therefore collect real RGB-D observations with ordered 3D annotations, or use them for fine-tuning and domain adaptation.

The current augmentation addresses only in-plane orientation. The target centerline, path-lifted prior, and visible point set are jointly rotated about the camera optical axis, while the SAM3 mask is transformed by the corresponding homography. This operation preserves geometric consistency but does not vary the three-dimensional viewpoint, occlusion pattern, sensor characteristics, or errors in the upstream geometric reconstruction. Future data generation should include broader DLO configurations and camera viewpoints, together with realistic depth noise, missing measurements, outliers, mask-boundary errors, and perturbations of the extracted path and prior. Variations in background, illumination, texture, and material would additionally test the robustness of the SAM3-based segmentation stage, even though the learned flow model itself operates on geometric rather than raw RGB features.

The geometric fallback remains conditional on successful upstream path extraction. In five frames of the real failure-case recordings, no usable ordered path was available and a path-lifted prior could therefore not be constructed. Improving the robustness of trace extraction under severe occlusion, missing endpoints, and fragmented masks is consequently necessary for increasing end-to-end output availability. The plausibility gate also relies on fixed geometric and mask-consistency thresholds. These criteria can reject visibly inconsistent predictions, but they are not calibrated measures of three-dimensional reconstruction error. Future work could instead estimate predictive uncertainty or learn a confidence score from validation data, allowing the decision between the learned prediction and the geometric prior to reflect reconstruction reliability more directly.

Finally, integrating the single-frame estimator with a temporal tracker would extend its role beyond initialisation. The estimator could provide a new state after tracking failure, while valid estimates from adjacent frames could help resolve ambiguities that cannot

be determined from a single observation. Such integration would require an explicit reinitialisation criterion and an evaluation of whether recovery improves subsequent tracking rather than only the current frame.

## 5.3 Conclusion

This thesis presented a geometry-conditioned velocity-flow estimator for recovering an ordered 3D DLO centerline from a single RGB-D observation without accurate initialisation. An observation-derived path prior provides global structure, while learned refinement incorporates unordered and local geometric evidence. The evaluation demonstrates improved synthetic-domain initialisation relative to TrackDLO-init and supports velocity prediction as the main iterative parameterization. At the same time, complex self-intersections, real-domain transfer, and reliable fallback remain open problems. The proposed method should therefore be viewed as a step toward robust single-frame DLO initialisation rather than a complete solution to general DLO state estimation.

# Bibliography


[1] J. Grannen, P. Sundaresan, B. Thananjeyan, *et al.*, "Untangling dense knots by learning task-relevant keypoints", in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 155, PMLR, 2021, pp. 782–800. [Online]. Available: https://proceedings.mlr.press/v155/grannen21a.html.

[2] J. Xiang, H. Dinkel, H. Zhao, *et al.*, "TrackDLO: Tracking deformable linear objects under occlusion with motion coherence", *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6179–6186, 2023. DOI: 10.1109/LRA.2023.3303710.

[3] 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)*, 2023, pp. 7119–7125. DOI: 10.1109/ICRA48891.2023.10160784.

[4] A. Choi, D. Tong, B. Park, D. Terzopoulos, J. Joo, and M. K. Jawed, "mBEST: Realtime deformable linear object detection through minimal bending energy skeleton pixel traversals", *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 4863–4870, 2023. DOI: 10.1109/LRA.2023.3290419.

[5] B. Guler, S. Manschitz, K. Pompetzki, and J. Peters, "AssistDLO: Assistive teleoperation for deformable linear object manipulation", *arXiv preprint arXiv:2605.06323*, 2026. DOI: 10.48550/arXiv.2605.06323.

[6] N. Carion, L. Gustafson, Y.-T. Hu, *et al.*, "SAM 3: Segment anything with concepts", *arXiv preprint arXiv:2511.16719*, 2025. DOI: 10.48550/arXiv.2511.16719.

[7] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling", in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=PqvMRDCJT9t.

[8] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation", *Science Robotics*, vol. 6, no. 54, eabd8803, 2021. DOI: 10.1126/scirobotics.abd8803.

[9] A. Caporali, I. Cuiral-Zueco, G. López-Nicolás, and G. Palli, "Robotic perception and manipulation of deformable linear objects: A survey", *The International Journal of Robotics Research*, 2026. DOI: 10.1177/02783649261432253.

[10] 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. DOI: 10.1109/LRA.2022.3189791.

[11] A. Kirillov, E. Mintun, N. Ravi, *et al.*, "Segment anything", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2023/html/Kirillov_Segment_Anything_ICCV_2023_paper.html.

[12] N. Ravi, V. Gabeur, Y.-T. Hu, *et al.*, "SAM 2: Segment anything in images and videos", *arXiv preprint arXiv:2408.00714*, 2024. DOI: 10.48550/arXiv.2408.00714.

[13] 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 *Computer Vision – ACCV 2018*, Springer, 2019, pp. 662–677. DOI: 10.1007/978-3-030-20890-5_42.

[14] 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. DOI: 10.1109/TII.2023.3245641.

[15] Z. Sun, H. Zhou, N. Li, 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, 2024. DOI: 10.1109/LRA.2023.3337695.

[16] 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)*, 2023, pp. 7104–7110. DOI: 10.1109/ICRA48891.2023.10160437.

[17] A. Myronenko and X. Song, "Point set registration: Coherent point drift", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010. DOI: 10.1109/TPAMI.2010.46.

[18] 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)*, 2019, pp. 6443–6450. DOI: 10.1109/IROS40897.2019.8967827.

[19] 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)*, 2021, pp. 14 199–14 205. DOI: 10.1109/ICRA48506.2021.9561012.

[20] T. Tang and M. Tomizuka, "Track deformable objects from point clouds with structure preserved registration", *The International Journal of Robotics Research*, vol. 38, no. 10–11, pp. 1207–1220, 2019. DOI: 10.1177/0278364919841431.

[21] Y. Yang, J. A. Stork, and T. Stoyanov, "Particle filters in latent space for robust deformable linear object tracking", *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 577–12 584, 2022. DOI: 10.1109/LRA.2022.3216985.

[22] K. Lv, M. Yu, S. Wang, X. Ji, and X. Li, "UniStateDLO: Unified generative state estimation and tracking of deformable linear objects under occlusion for constrained manipulation", *arXiv preprint arXiv:2512.17764*, 2025. DOI: 10.48550/arXiv.2512.17764.

[23] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Póczos, R. Salakhutdinov, and A. J. Smola, "Deep sets", in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[24] C. R. Qi, H. Su, K. Mo, and L. 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*, 2017, pp. 652–660.

[25] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space", in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[26] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need", in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[27] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks", in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97, 2019, pp. 3744–3753.

[28] A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira, "Perceiver: General perception with iterative attention", in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 139, 2021, pp. 4651–4664.

[29] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "PCN: Point completion network", in *2018 International Conference on 3D Vision (3DV)*, 2018, pp. 728–737. DOI: 10.1109/3DV.2018.00088.

[30] L. P. Tchapmi, V. Kosaraju, S. H. Rezatofighi, I. Reid, and S. Savarese, "TopNet: Structural point cloud decoder", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 383–392.

[31] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "PoinTr: Diverse point cloud completion with geometry-aware transformers", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 498–12 507.

[32] P. Xiang, X. Wen, Y.-S. Liu, *et al.*, "SnowflakeNet: Point cloud completion by snowflake point deconvolution with skip-transformer", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5499–5509.

[33] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models", in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 6840–6851.

[34] X. Liu, C. Gong, and Q. Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow", in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=XVjTT1nw5z.

[35] G. Yang, X. Huang, Z. Hao, M.-Y. Liu, S. Belongie, and B. Hariharan, "PointFlow: 3d point cloud generation with continuous normalizing flows", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4541–4550.

[36] S. Luo and W. Hu, "Diffusion probabilistic models for 3d point cloud generation", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2837–2845.

[37] L. Zhou, Y. Du, and J. Wu, "3d shape generation and completion through point-voxel diffusion", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5826–5835.

[38] L. Melas-Kyriazi, C. Rupprecht, and A. Vedaldi, "PC2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 923–12 932.

[39] H. Chung, J. Kim, M. T. McCann, M. L. Klasky, and J. C. Ye, "Diffusion posterior sampling for general noisy inverse problems", in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=OnD9zGAGT0k.

[40] J. Kim, B. S. Kim, and J. C. Ye, "FlowDPS: Flow-driven posterior sampling for inverse problems", *arXiv preprint arXiv:2503.08136*, 2025. DOI: 10.48550/arXiv.2503.08136.

[41] T. Eiter and H. Mannila, "Computing discrete fréchet distance", Christian Doppler Laboratory for Expert Systems, TU Vienna, Tech. Rep. CD-TR 94/64, 1994. [Online]. Available: https://www.kr.tuwien.ac.at/staff/eiter/et-archive/files/cdtr9464.pdf.

[42] J. Xiang, H. Dinkel, H. Zhao, *et al.*, *Data for TrackDLO: Tracking deformable linear objects under occlusion with motion coherence*, University of Illinois Urbana-Champaign, 2025. DOI: 10.13012/B2IDB-2916472_V1.

This thesis was independently written and linguistically revised with the assistance of OpenAI Codex and Anthropic Claude.