أكاديمية كاوست
KAUST ACADEMY

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

OCT-Based Pose Estimation of Surgical Instruments

Final Report

Ammar Alqahtani, **Lueder A. Kahrs**

ammarqaht@gmail.com, lueder.kahrs@utoronto.ca

Abstract

Safe control of articulated surgical instruments in robotic microsurgery requires real-time knowledge of the tool's joint configuration, which is difficult to observe inside the surgical field. Optical coherence tomography (OCT) images tissue and instruments together in 3D, but its volumes are speckled and mostly empty, and dense volumetric processing is too slow for closed-loop use. We present a deep-learning pipeline that estimates the full articulated pose of an instrument directly from OCT. The tool is modelled as a four-link kinematic chain (base, shaft, two jaws) parameterised by eight 3D keypoints, from which we derive the three tip angles the surgeon controls: yaw, pitch, and jaw opening. Each acquisition is treated as a sparse 3D point cloud and processed by VoxelNeXt, a fully sparse convolutional network, chosen because OCT volumes are mostly empty space. Ground truth was annotated with an interactive 3D tool and scaled by a model-assisted human-in-the-loop scheme in which the trained network pre-annotates new frames for rapid correction. We show the *same* pipeline transfers to a second, geometrically different instrument, a pair of surgical **scissors**, with no change to the network. Within a $17 \times 11 \times 26$ mm field of view both tools are localised with **sub-millimetre** keypoint error (0.75 ± 0.53 mm scissors, ~ 0.8 mm dVRK); the scissors recovers its three tip angles to $\mathbf{2.8}^\circ$ and the dVRK tool to 4.7° over all five of its angles, the short-linked jaw opening dominating the residual error in both. Leave-one-recording-out cross-validation shows accuracy varying between 1.0 and 1.6 mm across held-out recordings, so single-split figures should be read as optimistic and a larger labelled set remains the principal need.

Date: August 24, 2026

Contents

1 Introduction 2

2 Related Work 3

3 Methodology 3

3.1 System overview . . . 3

3.2 The two instruments . . . 4

3.3 Pose representation . . . 4

3.4 Data acquisition with OCT . . . 5

3.5 Annotating the volumes: building the ground truth . . . 6

3.6 Transferring the pipeline to the scissors . . . 6

3.7 Model: why a fully sparse network is the right choice . . . 7

3.8 Evaluation procedure . . . 8

4 Results 9

4.1 The same pipeline on a second instrument: scissors . . . 10

4.2 Cross-validation across recordings . . . 11

5 Discussion 13

6 Limitations and Future Work 14

7 Conclusion 15

References 16

1 Introduction

Summary of contributions

The central result of this project is that the pipeline is not tied to a single tool. I **extended the exact same method to a second, very different instrument, a pair of surgical scissors**, with no change to the network, and it reaches **sub-millimetre** keypoints (0.75 ± 0.53 mm) and a **2.8**° mean tip-angle error, on par with, and on the individual angles better than, the dVRK tool. Getting there involved four concrete steps: (i) a new labelling strategy that *annotates on the high-resolution scan but trains on the low-resolution preview* that the sensor actually delivers; (ii) running the model-assisted human-in-the-loop loop over several rounds to grow the scissors labels from a handful of frames to 63 verified frames spanning rolling, tilting, translation, and cutting motions; (iii) *simplifying the reported angles from five to the three tip angles* (yaw, pitch, and jaw opening) that match the lab-standard naming and are well defined for both tools; and (iv) running leave-one-recording-out cross-validation on the dVRK tool to measure how much accuracy varies between recordings. I also fixed the training so it keeps keypoints that lie outside the scanned field of view rather than clamping them to the edge, which previously crashed on frames where the tool base leaves the scan entirely.

Motivation and real-world impact.. Cable-driven articulated instruments, such as the da Vinci Research Kit (dVRK) tools, are central to minimally invasive robotic surgery: they provide wrist-level dexterity at the millimetre scale, which is essential for delicate procedures such as microsurgery. Controlling such an instrument safely and precisely requires the controller to know the instrument's full joint configuration, the yaw and pitch of its articulations and how far its jaws are open, in real time. In practice this configuration is hard to observe from the outside: cable stretch, backlash, and hysteresis mean that the commanded joint values do not reliably match the true pose of the tool tip inside the surgical field. A vision system that recovers the pose directly from images of the instrument would close this gap and enable more accurate, image-guided control.

Why OCT.. Optical coherence tomography (OCT) is an emerging volumetric imaging modality that captures the tissue and the instrument together in 3D at high resolution, making it a promising source of intra-operative feedback for microsurgery. Two properties, however, make OCT difficult to use directly for pose estimation. First, OCT volumes are corrupted by speckle noise. Second, the acquired volumes are large and dense, so processing them with conventional dense 3D networks is too slow for the real-time rates that closed-loop control requires.

Problem statement.. Given a single OCT acquisition of an articulated instrument, we want to localize the **eight 3D keypoints** of a four-link tool model, and from them recover the tool's **three tip joint angles** (yaw, pitch, and the jaw-opening angle) accurately and fast enough for feedback. The eight keypoints are the primary quantity the network predicts; our pipeline localizes them with a mean error of 0.75 **mm** on the scissors and ~ 0.8 **mm** on the dVRK tool, i.e. sub-millimetre on both instruments we have tested, within the OCT field of view. We derive three angles rather than the five we reported initially because yaw, pitch, and jaw opening are exactly the degrees of freedom the surgeon controls at the tip, they follow the lab-standard naming, and they are well defined for both a rigid-wrist tool and a symmetric-bladed one.

Key assumptions.. We assume the instrument follows a known four-link kinematic structure (base, shaft, and two jaws) that can be described by eight 3D keypoints, and that the tool is (at least partially) visible in the OCT field of view. We relaxed the earlier single-geometry assumption: the same four-link model and network are shown to transfer across two different tool geometries. All recordings are acquired on a benchtop robotic setup.

Proposed system.. We represent each OCT acquisition as a *sparse 3D point cloud* rather than a dense grid, and estimate the eight keypoints with **VoxelNeXt**, a fully sparse convolutional network that only computes on occupied voxels. Because OCT volumes are mostly empty space, this sparse formulation is a natural fit and avoids the cost of dense volumetric processing. From the eight predicted keypoints we analytically derive the three tip joint angles. Ground truth is produced with an interactive 3D annotation tool, and a model-assisted human-in-the-loop loop lets the trained network pre-annotate new frames so that the labelled set can grow with minimal manual effort. The remainder of this report describes the method, its extension to a second instrument, the evaluation protocol, and the accuracy obtained on held-out recordings for both tools.

2 Related Work

OCT-based pose estimation of surgical tools.. The closest prior work is that of Fredin et al. [1], who use OCT imaging to estimate the pose of an articulated *magnetic* surgical tool and then close the control loop around that estimate, demonstrating sub-millimetre accuracy and high-speed volumetric feedback. That work establishes OCT as a viable sensing modality for millimetre-scale instrument pose and motivates the present project. Our work differs in the instrument being tracked, a cable-driven, fully articulated dVRK tool with two independently moving jaws rather than a magnetic tool, and therefore in the pose representation: we model the instrument as a four-link chain with eight keypoints and recover five joint angles, including the jaw-opening angle, which is not present in a single-body magnetic tool. Methodologically, we also adopt a sparse, point-cloud view of the OCT volume to keep inference cheap.

Sparse 3D backbones.. For the backbone we build on VoxelNeXt [2], a fully sparse convolutional network introduced for LiDAR-based 3D object detection and tracking. VoxelNeXt performs detection directly on sparse voxel features, without dense feature maps or anchor grids, which makes it efficient on point clouds that are dominated by empty space. LiDAR scenes and OCT volumes share exactly this property, most of the volume is unoccupied, so a backbone designed to concentrate computation on occupied voxels transfers well to OCT. The gap we address relative to its original setting is domain adaptation: OCT point clouds have a very different (and much lower) intensity distribution than LiDAR returns, and the network's voxel-occupancy threshold and keypoint head must be retuned accordingly (Section 3).

3 Methodology

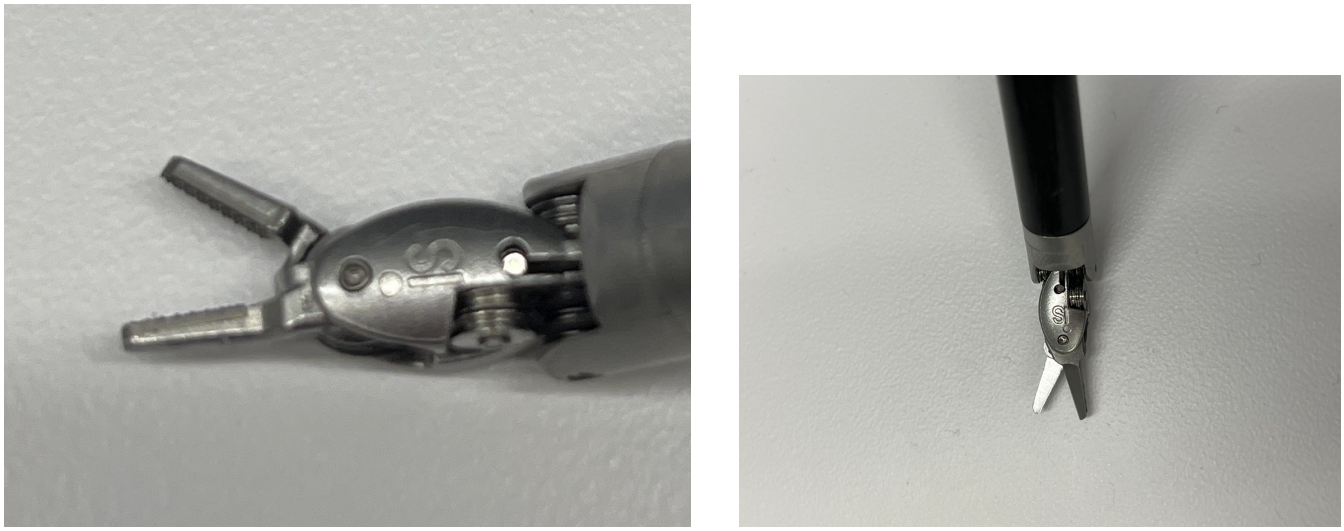
3.1 System overview

The pipeline maps a single OCT acquisition to the instrument's three tip joint angles in four stages: (1) represent the acquisition as a sparse 3D point cloud; (2) voxelize it onto a coarse grid that preserves the fine tool structure (Section 3.4); (3) predict eight 3D keypoints with a fully sparse convolutional network (Section 3.7); and (4) derive the three tip angles analytically from the keypoints. Crucially, these four stages are *instrument-agnostic*: we applied the identical pipeline to two different tools, the dVRK instrument and a pair of surgical scissors.

This section follows the order in which the work was actually carried out. We first introduce the **two instruments** (Section 3.2) and the pose representation used for both (Section 3.3). We then describe how the **data was acquired with OCT** and turned into the sparse input the network consumes (Section 3.4), how that data was **annotated** to produce ground truth (Section 3.5), and only then the **model** trained on it (Section 3.7) and the evaluation protocol (Section 3.8).

3.2 The two instruments

The method was developed and validated on *two* physically different articulated instruments (Figure 1). The first is a **da Vinci (dVRK) instrument**, a cable-driven tool whose wrist bends in yaw and pitch and whose two asymmetric jaws grip. The second is a pair of **surgical scissors**, whose two blades are *symmetric* and interchangeable. The difference matters: for the scissors, roll is geometrically undefined when the blades are closed, and an annotator cannot reliably keep the two blades in a consistent order across frames as the tool rolls. Both consequences are handled in the pose representation below. Using two tools of genuinely different geometry, rather than two views of one tool, is what lets us claim the method is not instrument-specific.



(a) da Vinci (dVRK) instrument

(b) surgical scissors

Figure 1 The two instruments the pipeline was applied to, at the same scale of detail. *(a)* The dVRK tool: a cable-driven wrist bending in yaw and pitch, with two *asymmetric* gripping jaws whose serrated faces differ. *(b)* The surgical scissors: two *symmetric*, interchangeable blades. That symmetry is what makes roll undefined when the blades are closed, and what makes the two jaws impossible for an annotator to order consistently as the tool rolls — both handled in Section 3.3. The same four-link, eight-keypoint model and the same network are used for both.

3.3 Pose representation

The instrument is modeled as a four-link kinematic chain, *base*, *shaft*, *jaw 1* and *jaw 2*, where each link is described by a back point and a front point, giving eight 3D keypoints. From the link direction vectors we compute the three tip joint angles the surgeon controls: *yaw* and *pitch* of the tool tip (left/right and up/down of the mean tip direction) and the *jaw-opening* angle (the angle between the two jaw directions), illustrated in Figure 2. We initially derived *five* angles (shaft yaw/pitch, wrist yaw/pitch, and jaw opening) but consolidated the report to these three: they are the lab-standard tip degrees of freedom, they avoid the redundancy between the shaft and wrist yaw/pitch, and they remain well defined for a symmetric-bladed tool such as the scissors,

where a separate roll angle is ambiguous because the two blades are interchangeable. For the scissors we additionally *canonicalize* the two jaw keypoints (the upper jaw is always assigned to slot 1) so that the interchangeable blades receive a consistent label across frames as the tool rolls.

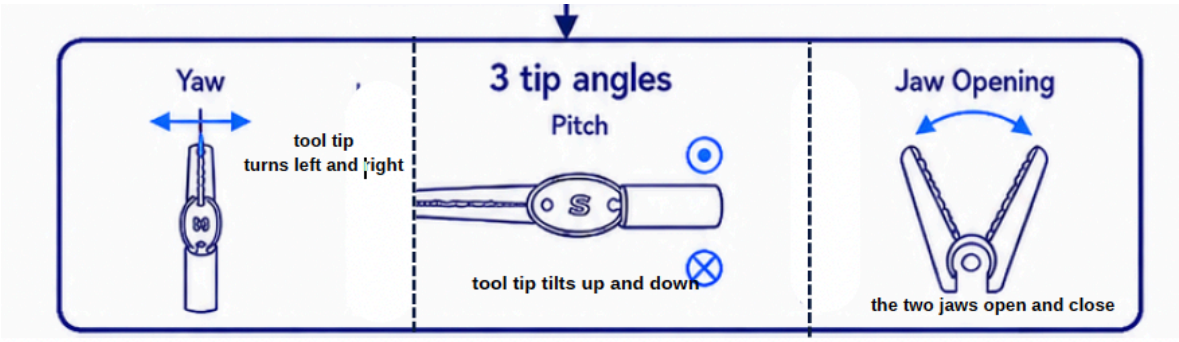


Figure 2 The three tip angles we estimate, shown on the instrument. *Yaw:* the tool tip turns left and right. *Pitch:* the tool tip tilts up and down (the ⊙ and ⊗ marks denote the rotation axis pointing out of and into the page). *Jaw opening:* the angle between the two jaws as they open and close. These are the degrees of freedom the surgeon controls at the tip, and they are well defined for both instruments.

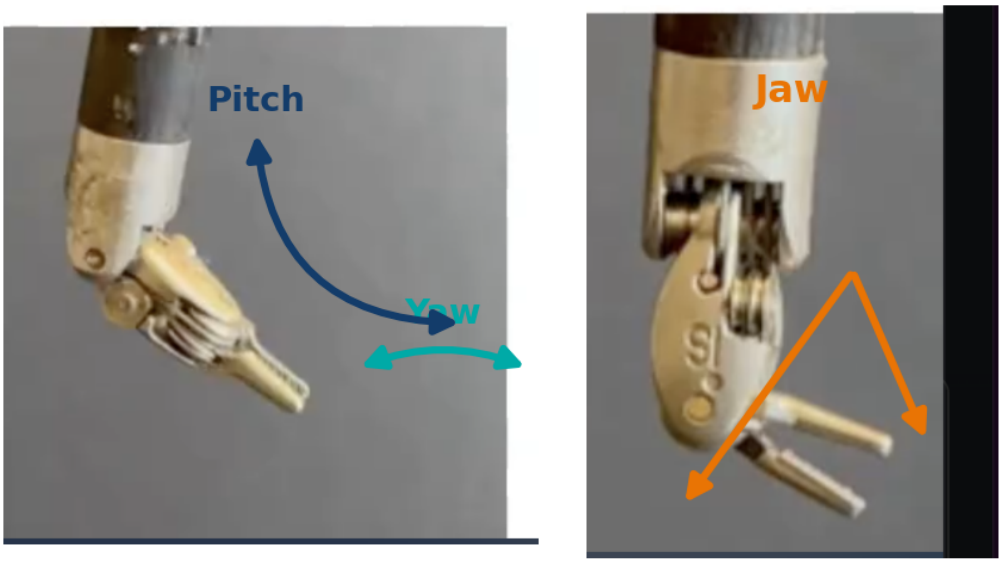


Figure 3 The estimated tip angles shown on real OCT frames of the dVRK instrument, with arrows for the direction of motion: the tip yaw/pitch (left; mean error 3.7° / 2.8°) and the jaw opening (right; 9.2°).

3.4 Data acquisition with OCT

All data used in this project was acquired in-house on the lab's OCT system: there is no external or public dataset behind these results.

What the sensor produces.. Optical coherence tomography builds a volume by sweeping a low-coherence beam across the target and measuring back-scattered light, giving a stack of cross-sectional B-scans. Each acquisition here covers a field of view of approximately $17.4 \times 10.8 \times 26.1$ mm (depth × height × width). The sensor delivers two versions of each frame: a *high-resolution* scan of ~1,100 depth slices, which is visually crisp but slow to acquire, and a *low-resolution preview* of 28 depth slices, which is what is available at the rate a control loop needs. All results in this report are produced on the **preview**, the realistic input.

How the recordings were captured.. The instrument is mounted on a robotic arm and moved through a scripted sequence while the OCT system records continuously, so that a single recording sweeps a wide range of poses rather than repeating one configuration. For the dVRK tool the recordings are grouped by jaw state: one with the jaws *closed*, one *open*, and one *mixed*. For the scissors, eight recordings deliberately span *rolling*, *tilting*, translation and cutting motions, which is what gives the smaller scissors set its pose diversity. Frames are then sub-sampled along each recording so that consecutive labelled frames are visibly different rather than near-duplicates.

From volume to sparse point cloud.. Every voxel above an intensity threshold becomes one row (x, y, z, i) of a point cloud, with coordinates in *millimetres* and intensity normalised to $[0, 1]$; a typical frame yields on the order of 10^5 points. Storing coordinates metrically rather than as voxel indices is a small decision with a large consequence, exploited in Section 3.5. One practical detail: the recordings arrive as a mix of 8-bit and 16-bit volumes, so the threshold is rescaled per data type; applying a single fixed threshold silently empties half the dataset.

Voxelization: bin every point, do not subsample.. Rather than reconstruct a dense high-resolution volume and then subsample it, which discards most of the measured points and erases the thin jaw tips, we *bin every measured point directly into the coarse target grid*, keeping the strongest intensity per cell. On a representative frame this preserves essentially all ~3,900 occupied cells instead of the ~80 that survive the reconstruct-then-subsample path, and at the resulting ~0.1 mm voxel size the two closely spaced jaws remain separable. As Section 4 shows, this single choice is the largest contributor to the final accuracy.

Keypoints outside the scan.. When part of the tool, most often the base, extends beyond the scanned volume, we *keep* its true out-of-view keypoints and let the regression head predict beyond the edge, rather than clamping them to the boundary, which would corrupt the ground truth. This is essential for the scissors, whose base is routinely outside the scan; clamping a link that lies entirely outside the volume also crashes outright.

3.5 Annotating the volumes: building the ground truth

Ground-truth poses are created with an interactive 3D annotation tool that we use frame by frame. Each OCT recording of the robot-manipulated dVRK tool is loaded as a 3D point cloud, and the annotator overlays the four-link tool model on it: for every link (base, shaft, jaw 1, jaw 2) a box is positioned and oriented so that its two ends coincide with that part of the instrument in the volume, which fixes the link's back and front keypoints. The links are colour-coded and the eight keypoints are shown in green, so a correct annotation visibly traces the tool through the volume (Figure 4). The jaw state (open, closed, or mixed) is recorded with each frame. A query-based interface lets us jump to any recording, frame, or annotation status (unlabelled, draft, or verified) and step through frames quickly.

To make annotation scale, we use a **model-assisted human-in-the-loop** scheme: once a model is trained, it *pre-annotates* new frames, and the annotator only *corrects* the predicted links rather than placing them from scratch. Corrected frames are marked as verified and folded back into training; unverified pre-annotations are automatically excluded from training and evaluation. This loop lets the labelled set grow with far less manual effort as the model improves.

3.6 Transferring the pipeline to the scissors

To test whether the method is specific to the dVRK tool or genuinely general, we applied the *identical* pipeline, the same four-link eight-keypoint model, the same network, loss, and training code, to the scissors introduced in Section 3.2. No architectural change was required; only the

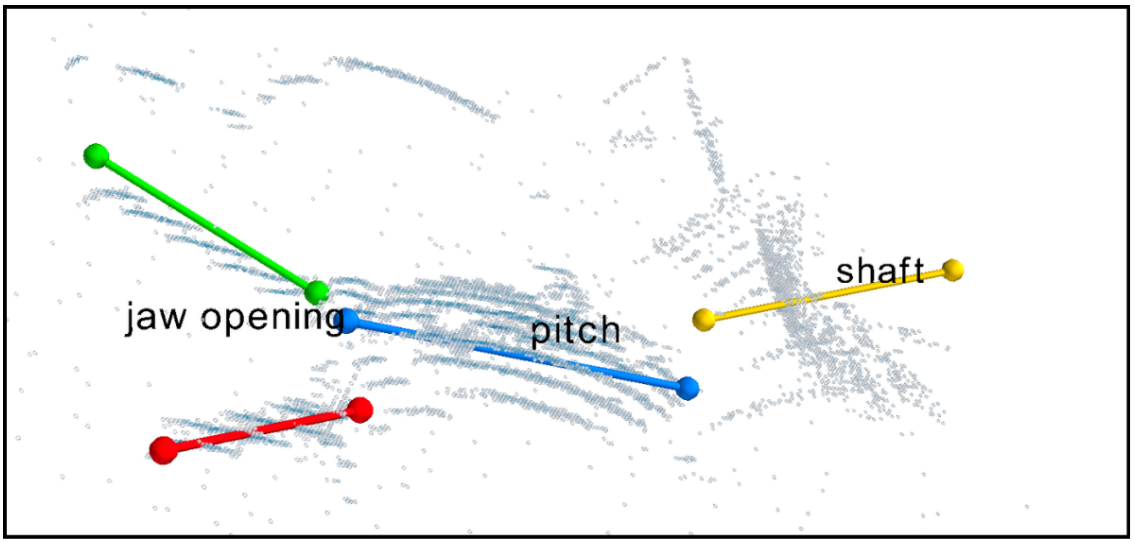


Figure 4 A completed annotation on an OCT point cloud (grey). The four links of the tool model are placed by hand, each defined by its two endpoints: the **shaft** (yellow), the wrist link whose direction gives the **pitch** (blue), and the two jaws (red and green) whose relative angle is the **jaw opening**. The eight endpoints are exactly the keypoints the network is trained to predict, and every reported angle is derived from them. A correct annotation visibly traces the instrument through the volume.

data and two tool-specific settings differ (jaw canonicalization, Section 3.3, and keeping out-of-view keypoints, Section 3.4).

Annotate on the high-resolution scan, train on the low-resolution preview.. For the scissors the sensor provides two versions of every frame: a *high-resolution* scan (~1,100 depth slices), which is visually crisp, and a *low-resolution preview* (~28 depth slices), which is what is actually available for fast, near-real-time use. Annotating the blurry preview would place the keypoints imprecisely, so we annotate on the high-resolution scan, where the blades are clearly resolved, but *train and evaluate on the low-resolution preview* that the system will receive in practice. This is possible because our labels are stored in physical millimetres rather than voxel indices, so a single annotation transfers to any resolution unchanged: we annotate once on the clear scan, then down-sample and train on the realistic input. Figure 6 shows an annotated scissors sample.

Human-in-the-loop growth.. We grew the scissors labels with the same model-assisted loop as the dVRK tool: annotate a first batch, train, let the model pre-annotate the next batch, and only *correct* its predictions. Over several rounds this produced **63 verified frames** drawn from eight recordings that deliberately span very different motions, the scissors *rolling*, *tilting*, translating, and cutting, so the model sees a wide range of poses rather than one repeated configuration.

3.7 Model: why a fully sparse network is the right choice

The backbone is **VoxelNeXt** [2], a fully sparse 3D convolutional network introduced for LiDAR object detection. It is the most suitable choice here for three reasons. *First, it matches the data:* OCT volumes are almost entirely empty space, and a fully sparse network computes only on occupied voxels, avoiding the large cost a dense 3D CNN would spend on background, the same property that makes it fast enough to be a step toward real-time feedback. *Second, it is purpose-built for this sparsity regime:* it detects directly on sparse voxel features without dense feature maps or anchor grids, which is exactly the regime LiDAR and OCT share. *Third, it is easily retargeted:* its keypoint head can be configured to output our eight tool keypoints as per-keypoint heatmaps

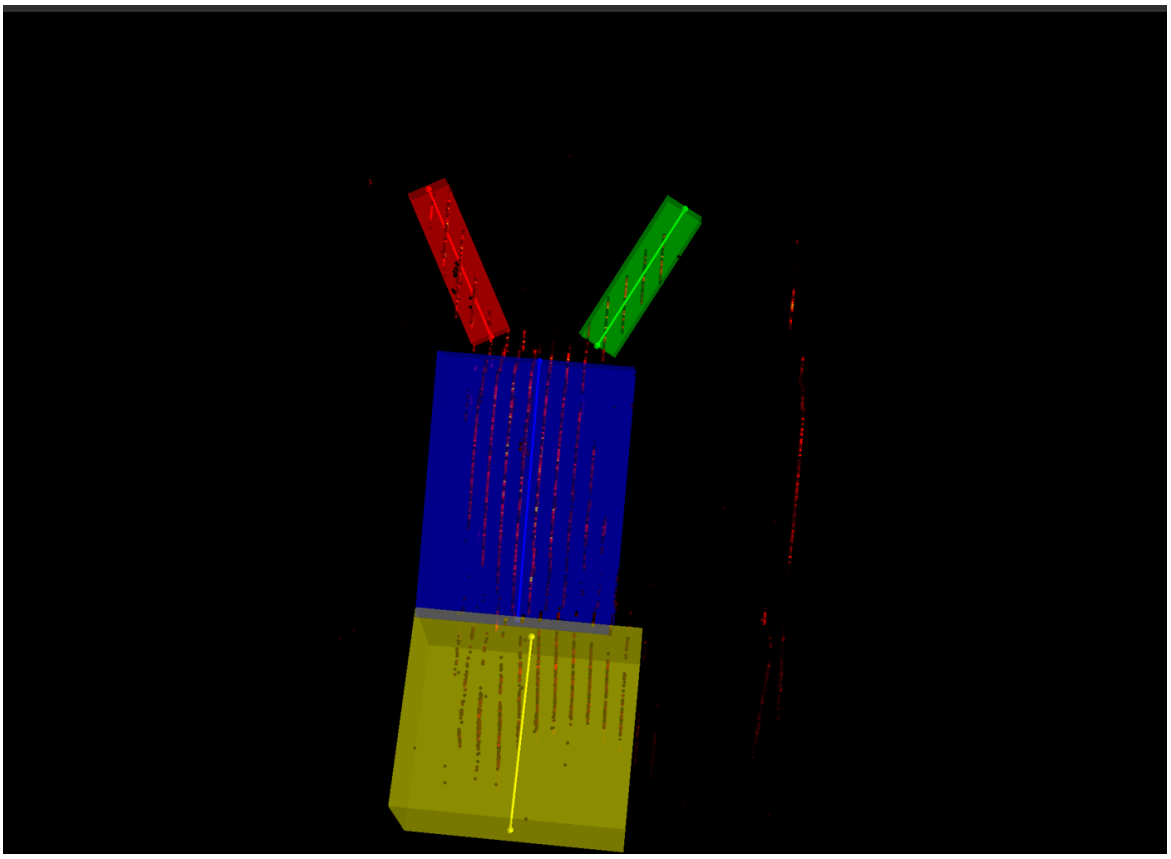


Figure 5 The annotation interface. The OCT point cloud is shown with the four-link model overlaid as colour-coded oriented boxes, one per link. For each link the annotator sets the box centre, size, and orientation, which fixes that link's two endpoints. The jaw state (here *open*) is recorded per frame.

plus sub-voxel offsets. Adapting it from LiDAR to OCT required two changes: lowering the voxel-occupancy threshold by more than an order of magnitude to match the low OCT intensity range (the LiDAR default discards every OCT voxel), and up-weighting the four jaw keypoints in the loss, because the very short (~6 mm) jaw links turn a small keypoint error into a large angular error.

3.8 Evaluation procedure

Tools.. The pipeline is implemented in PyTorch with sparse 3D convolutions (spconv); training and inference run on a GPU compute cluster (Alliance Canada), and the annotation and visualization tools run locally.

Data.. For the **dVRK** tool the labelled set consists of robot-manipulated OCT recordings grouped by tool state, one *closed*, one *open*, and one *mixed* recording; we hold out the mixed recording entirely for evaluation and train on the other two (80 verified training frames and 30 verified held-out frames). For the **scissors** we have 63 verified frames from eight recordings covering rolling, tilting, translation, and cutting; we train on 49 and hold out 13. Holding out data by recording (dVRK) or by a stratified spread across recordings (scissors), rather than shuffling adjacent, near-duplicate frames, prevents leakage and tests true generalization. For the dVRK tool we additionally adopt *leave-one-recording-out* cross-validation, in which each of the three recordings is held out in turn, so that the reported accuracy does not depend on the luck of a single split.

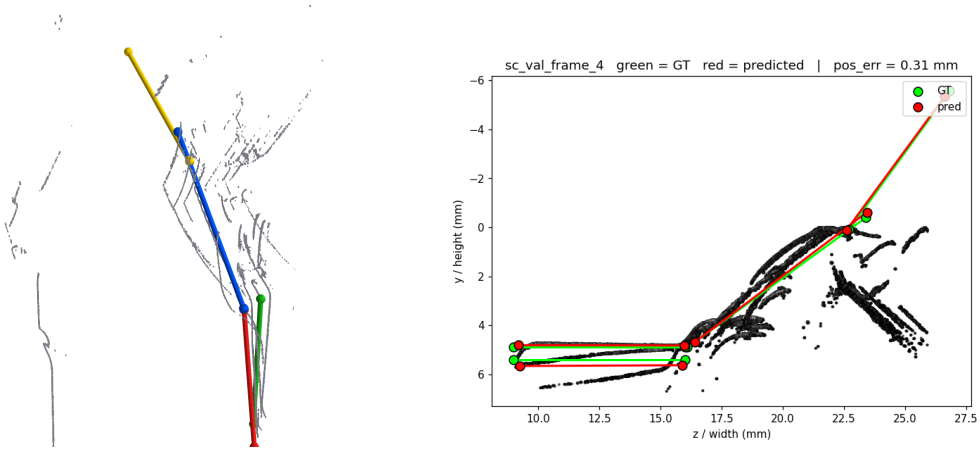


Figure 6 Surgical scissors in OCT. *Left:* an annotated 3D sample; the four-link tool model is overlaid on the point cloud (yellow base, blue shaft, red and green jaws), fixing the eight keypoints. *Right:* a held-out prediction (green = ground truth, red = predicted); the two nearly coincide. The frames come from recordings with rolling and tilting motions.

Metrics.. We report (i) the *keypoint distance error*, the mean Euclidean distance in millimetres between predicted and ground-truth keypoints, and (ii) the *joint-angle errors*, the absolute error in degrees for each of the three tip angles (yaw, pitch, jaw opening) and their mean. Baselines are an earlier, un-adapted version of the pipeline and the successive ablations of our own design choices (Section 4).

4 Results

Results for the dVRK tool are reported on its *held-out* recording (the mixed-jaw recording, 30 verified frames, trained on 80 frames from the two other recordings); results for the scissors are reported on its 13 held-out frames, trained on 49. Neither held-out set is seen during training, and the scissors train/validation/draft splits were checked to be frame-disjoint. Section 4.1 compares the two instruments directly and Section 4.2 reports cross-validation.

A note on the two angle averages.. The dVRK tool exposes five joint angles (base–shaft yaw and pitch, wrist yaw and pitch, jaw opening) whereas the scissors exposes three (yaw, pitch, jaw opening), because roll is undefined for symmetric blades. We therefore quote two averages for the dVRK tool and keep them explicitly distinct: a **three-angle tip mean of** $5.2°$, which is the like-for-like comparison against the scissors, and a **five-angle overall mean of** $4.7°$ across every angle the tool reports. Both come from the same model on the same held-out recording.

Keypoint localization (primary output).. The core output of the network is the set of **eight 3D keypoints**; every joint angle is derived from them, so their accuracy sets the ceiling for the whole system. The final pipeline localizes these eight keypoints with a **sub-millimetre mean error** within a $17.4 \times 10.8 \times 26.1$ mm field of view — 0.75 ± 0.53 mm on the scissors and ~ 0.8 mm on the dVRK tool — i.e. the predicted tool sits within about one voxel of the ground truth. Figure 7 shows a representative held-out frame in which the predicted (red) and ground-truth (green) keypoints nearly coincide (0.81 mm). As a baseline, the initial adapted pipeline (dense reconstruction then subsampling) reached only about **1.4 mm**, so preserving all measured points during voxelization roughly halves the keypoint localization error.

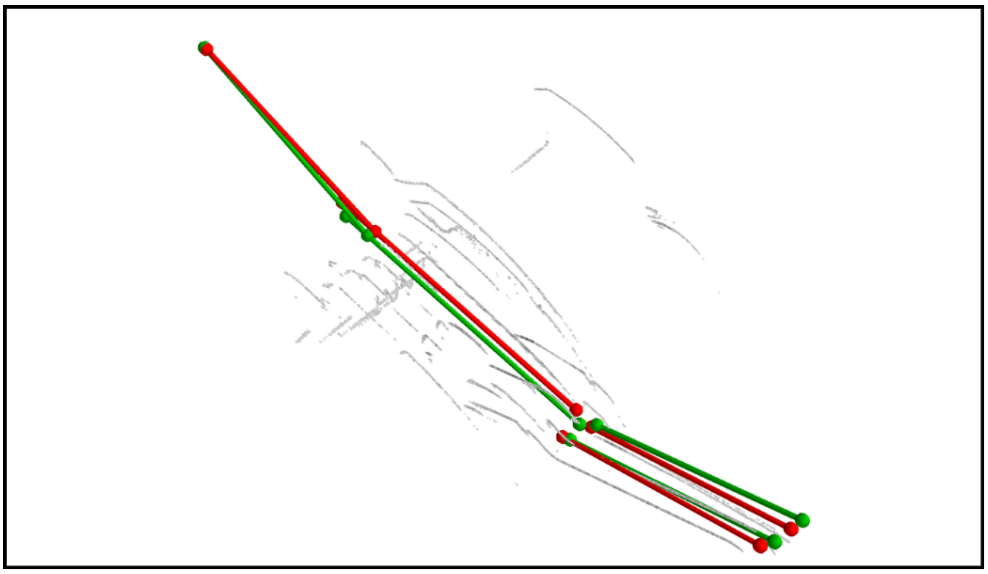


Figure 7 Model output on a *held-out* frame: predicted pose (red) against the manually annotated ground truth (green), over the OCT point cloud in grey. All four links — shaft, wrist, and both jaws — track the instrument, and the two poses very nearly coincide along the whole tool, including at the thin jaw tips on the lower right where the error is hardest to control.

Joint-angle accuracy.. Table 1 reports all five dVRK joint angles. The three tip angles the surgeon controls — yaw, pitch, and the jaw opening — are recovered to a mean of $5.2°$, and the mean over all five angles is $4.7°$. Wrist yaw and pitch are within about $3°$ and the two base–shaft angles within about $4°$, while the jaw opening, whose two ~6 mm links amplify any keypoint error, is by far the largest at $9.2°$. Figure 3 shows these angles on real frames of the tool.

Table 1 Held-out results for the final dVRK model (30 frames): the 8-keypoint distance error and every joint angle the tool reports. The tip mean covers the three angles that are also defined for the scissors; the overall mean covers all five.

Metric	Mean	Std
Keypoint distance	~0.8 mm	—
Base–shaft yaw	3.7°	2.9°
Base–shaft pitch	4.3°	2.9°
Wrist yaw	3.7°	3.6°
Wrist pitch	2.8°	2.6°
Jaw opening	9.2°	8.9°
Tip mean (3 angles)	5.2°	—
Overall mean (5 angles)	**4.7°**	—

4.1 The same pipeline on a second instrument: scissors

Applying the identical pipeline to the surgical scissors, with no change to the network, produces results *on par with or better than* the dVRK tool (Table 2). The scissors keypoints are localized to 0.75 ± 0.53 **mm**, essentially the same sub-millimetre accuracy as the dVRK tool, and its three tip angles are recovered to a mean of **2.8**°, better than the dVRK tool on every tip angle: yaw $2.0°$ vs. $3.7°$, pitch $2.2°$ vs. $2.8°$, and jaw opening $4.2°$ vs. $9.2°$. It reaches this with *fewer* labelled

frames (49 training / 13 held-out vs. 80 / 30), which we attribute partly to the highly varied scissors recordings (rolling and tilting) and partly to the crisper high-resolution annotations. The main message is that the method is not specific to one instrument: the same sparse-keypoint formulation transfers cleanly across tool geometries.

Table 2 The same pipeline on two instruments, each on its own held-out frames. Keypoint accuracy is essentially identical; the scissors is better on every tip angle, with fewer labelled frames.

Metric	dVRK	Scissors
Keypoint distance	~0.8 mm	**0.75 mm**
Yaw (tip)	3.7°	**2.0°**
Pitch (tip)	2.8°	**2.2°**
Jaw opening	9.2°	**4.2°**
Tip mean (3 angles)	5.2°	**2.8°**
Frames (train / val)	80 / 30	49 / 13

Figure 8 shows the scissors training and validation loss, which decrease smoothly and plateau together, with no sign of overfitting despite the smaller labelled set.

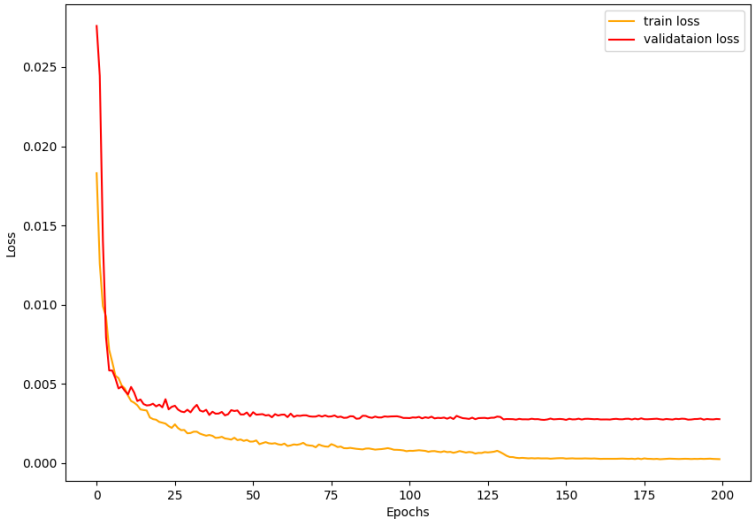


Figure 8 Scissors training and validation loss over 200 epochs. The validation loss tracks the training loss and plateaus, indicating a stable fit on the 49-frame training set.

4.2 Cross-validation across recordings

A single held-out recording measures accuracy on one particular scene. To see how much that figure depends on *which* recording is held out, we ran leave-one-recording-out (LOEO) cross-validation on the dVRK tool, training on two recordings and validating on the third, for all three folds (Table 3).

The keypoint error varies from 1.01 to 1.58 mm depending on which recording is held out, a spread of roughly ±0.3 mm around the mean. Two conclusions follow. First, the method de-

Table 3 Leave-one-recording-out cross-validation on the dVRK tool. Each row holds out one whole recording. Run with an earlier configuration (76 labelled frames, no jaw weighting), so the absolute values are higher than the final model's; the informative quantity is the *spread* between folds.

Held-out recording	Train / val frames	Keypoint error
e2000	49 / 27	1.01 mm
e2001	42 / 34	1.58 mm
e2002	61 / 15	1.37 mm
Mean		**1.32 mm**

grades gracefully rather than catastrophically on an unseen scene: no fold collapses. Second, and more importantly for how these numbers should be read, *a single split can easily be optimistic or pessimistic by a few tenths of a millimetre*, so the sub-millimetre single-split figures reported above should be treated as indicative rather than as a tight bound.

We note explicitly that this cross-validation was run on an earlier configuration than the final model — 76 labelled frames rather than 110, and without the jaw-weighted loss — which is why its absolute errors exceed the ~ 0.8 mm the final model reaches on its own split. Cross-validating the final configuration is the most valuable single experiment left to run, and is the first item of future work.

Ablation of design choices.. Table 4 tracks the two headline metrics, the jaw-opening error and the overall mean angle error, as we add each design choice on top of the adapted baseline. The all-point voxelization is the single largest contributor (it makes the jaws separable at all), the jaw-weighted loss together with a full learning-rate schedule roughly halves the error again, and additional labelled data gives a further improvement.

Table 4 Effect of each design choice on the held-out recording. Lower is better; the jaw-opening angle is the hardest joint and is reported separately from the overall mean.

Configuration	Jaw opening (°)	Overall mean (°)
Adapted baseline (reconstruct + subsample)	40.4	16.1
+ all-point voxelization	25.6	9.2
+ jaw-weighted loss & full schedule	10.1	5.0
+ more labelled data (**final**)	**9.2**	**4.7**

Overall, the jaw-opening error is reduced from $40.4°$ to $9.2°$ and the mean angle error from $16.1°$ to $4.7°$, roughly a $3.4\times$ improvement over the adapted baseline, while keypoint localization reaches sub-millimetre accuracy.

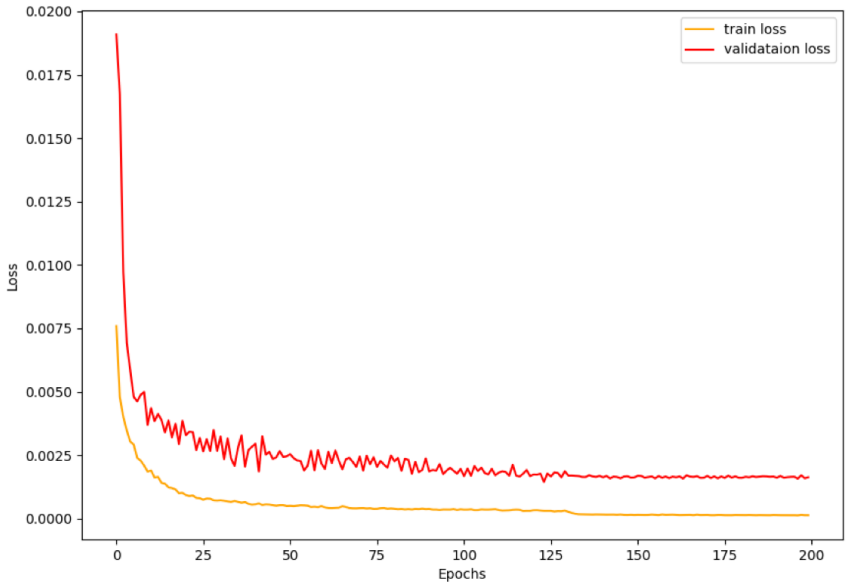


Figure 9 Training and validation loss over 200 epochs for the final model. Both curves decrease smoothly and plateau, with the validation loss tracking the training loss and no sign of divergence or overfitting.

5 Discussion

The method generalizes across instruments.. The most encouraging result of this project is that the pipeline is not tuned to the dVRK tool. Pointing the identical network at a geometrically very different instrument, symmetric scissor blades instead of a cable-driven wrist, reproduced sub-millimetre keypoints and, in fact, *lower* angle errors, with fewer labelled frames. Two design choices made the transfer painless. First, keypoints are stored in physical millimetres, so the same annotation is valid at any resolution; this is what let us annotate on the crisp high-resolution scan yet train on the realistic low-resolution preview. Second, the tool is described only as a generic four-link chain, with no instrument-specific priors, so a new tool needs new data but not new code. The one genuinely tool-specific consideration is the symmetry of the scissor blades, which we resolved by canonicalizing the two jaw keypoints and by reporting the three tip angles rather than a blade-dependent roll.

Why the voxelization step mattered so much.. The largest single jump in accuracy came not from a bigger network or more data but from how the OCT point cloud is turned into the network input. The initial pipeline reconstructed a dense high-resolution volume and then subsampled it; on a typical frame this kept only a small fraction of the measured points and, because the two jaws are thin and closely spaced, frequently erased them entirely. The network then saw the jaws as a single blob and *systematically under-predicted* the jaw opening, for example predicting roughly $13°$ where the true opening was around $50°$. Binning every measured point directly into the coarse grid keeps the jaw tips, makes the two jaws separable at the working resolution, and is what turned the jaw-opening angle from unusable ($40°$ error) into merely difficult ($\sim 9°$). This is the clearest lesson of the project: for thin, articulated tools in a sparse modality, *preserving fine structure in the input matters more than model capacity*.

Why the jaw opening remains the hardest joint.. Even after all improvements, the jaw-opening angle is by far the largest error ($9.2°$ versus under $4.3°$ for every other joint). This is expected from the geometry rather than a failure of the model: the jaw links are only about 6 mm long, so the same ~1 mm keypoint error that is negligible for the long base and shaft links translates into a large angular error at the jaw. The yaw and pitch angles, which depend on longer links, are correspondingly accurate. In other words, the error pattern follows link length, which is what we would predict, and it sets a physical floor that better keypoints, not a different loss, must push against. The pattern holds for the scissors too, but its blades are annotated more precisely from the high-resolution scan, which is part of why its jaw-opening error ($4.2°$) is roughly half that of the dVRK tool.

Why yaw is harder than pitch.. A second, subtler pattern is that the yaw angles are consistently noisier than the pitch angles. This traces directly to the OCT geometry: the volume is sampled with only about **28 slices along the depth axis** (roughly 0.6 mm per slice), versus tens of micrometres along the height and width. Yaw is measured in the depth–width plane and therefore inherits the coarse depth resolution, whereas pitch, measured against the finely sampled height axis, is more accurate. This is a limit of the sensor rather than the model, and it points to depth-weighted supervision as a future refinement.

Why sparse convolutions were the right choice.. Treating the OCT volume as a sparse point cloud, and processing it with a fully sparse backbone, was well matched to the data: OCT volumes are almost entirely empty, so computing only on occupied voxels avoids wasting work on background. The main friction in adopting a LiDAR backbone was the very different intensity statistics of OCT; once the occupancy threshold was retuned to the low OCT intensity range, training was stable.

Frames with the tool out of view.. A further, more practical source of error is field-of-view coverage: in some frames part of the tool falls outside the scanned OCT volume, so a link (most often the base) is only partially imaged or extends beyond the edge. Its keypoints then have to be inferred from incomplete evidence, and these frames carry noticeably larger errors than fully-visible ones. This is a limit of the acquisition rather than the model, and a wider or repositioned scan would reduce it.

Unexpected findings.. Three observations were not obvious at the outset. First, weighting the jaw keypoints in the loss improved the *tip yaw/pitch* angles as well, because those angles are computed from the jaw directions, so a single change helped several metrics at once. Second, the keypoint distance error and the angle error did not always move together: a configuration could have a slightly larger keypoint error yet much better angles, which reinforces that for this task the angles (and especially where the error lands geometrically) are the metric that matters, not the raw keypoint distance alone. Third, a subtle but important lesson from the scissors experiments was procedural rather than algorithmic: stale cross-validation folders left on the compute cluster once silently biased an evaluation (a validation frame had leaked into training), which is why we now verify the exact train/validation split printed at the start of every run.

6 Limitations and Future Work

Small labelled dataset.. Both instruments are trained on modest labelled sets, $80/30$ frames for the dVRK tool and $49/13$ for the scissors. This limits how tightly accuracy and, especially, generalization can be established, and the 13-frame scissors validation set in particular makes single-split numbers noisy. The cross-validation in Section 4.2 quantifies this directly: holding out a different recording moves the keypoint error by roughly ± 0.3 mm. *Next step:* continue the model-

assisted human-in-the-loop loop — the trained network already pre-annotates new recordings for rapid correction, and 14 such drafts are awaiting review at the time of writing — to grow both labelled sets.

The final configuration has not been cross-validated.. The leave-one-recording-out study was run on an earlier configuration (76 labelled frames, no jaw-weighted loss); the final model's sub-millimetre figures come from a *single* held-out split each. The two therefore cannot be combined into one headline number, and the single-split figures should be read as indicative. *Next step:* re-run LOEO on the dVRK tool and k-fold on the scissors with the final configuration and the current labels, and report mean and spread as the primary result.

Correlated validation frames in the scissors split.. The 13 scissors validation frames were sampled *across* recordings rather than by holding out a whole recording. The splits are frame-disjoint — verified explicitly — but neighbouring frames of the same recording are highly correlated, so a validation frame may sit only a few frames away from a training one. The scissors figure is therefore optimistic as an estimate of performance on a genuinely unseen recording, which is precisely the protocol the dVRK tool already uses. *Next step:* adopt a whole recording holdout for the scissors as soon as the labelled set is large enough to afford it.

Jaw-opening accuracy.. The jaw-opening angle remains the dominant error source because of the short jaw links. *Next steps:* (i) collect more frames that cover the full range of jaw states, including closed and partially open jaws, since the held-out recording is biased toward open jaws; and (ii) explore a supervision signal on the jaw-opening angle directly, in addition to the keypoint-based supervision, to push against the geometric error floor.

From two instruments toward many.. We extended the method from one tool geometry to two (the dVRK instrument and surgical scissors), which is strong initial evidence that the four-link, eight-keypoint formulation is not tool-specific. *Next step:* test further instrument geometries and jaw designs, and study how much labelled data a genuinely new tool needs, whether a model trained on several tools can bootstrap a new one with very few annotations.

Robustness and real-time evaluation.. Robustness to speckle noise and to partial occlusion, both common in real OCT, has not yet been systematically quantified, and although the sparse backbone was chosen for speed, inference latency has not been benchmarked against a real-time control budget. *Next steps:* evaluate accuracy under controlled noise and occlusion, add matching data augmentation during training, and measure end-to-end inference time so the pipeline can be placed inside a closed-loop controller in the spirit of prior OCT-based feedback work [1].

7 Conclusion

We presented a deep-learning pipeline that estimates the articulated pose of a surgical instrument directly from OCT volumes, by representing each acquisition as a sparse point cloud, localizing eight keypoints of a four-link tool model with a fully sparse convolutional network (VoxelNeXt), and analytically deriving the three tip joint angles. The most important methodological finding remains that, for a thin articulated tool in a sparse imaging modality, how the volume is voxelized dominates accuracy: binning every measured point into the coarse grid, rather than reconstructing and subsampling, preserves the jaw tips and is what makes the jaw-opening angle recoverable. The key result is that the same pipeline transfers to a second, geometrically different instrument, a pair of surgical scissors, with no change to the network: both tools are localized with sub-millimetre keypoint accuracy, and the scissors recovers its three tip angles to a mean of

2.8° (matching, and on the individual angles beating, the dVRK tool) with fewer labelled frames. A simple but crucial enabler was storing labels in physical millimetres, which let us annotate on a crisp high-resolution scan yet train on the low-resolution preview the sensor actually delivers. The overall takeaway is that OCT, processed sparsely, is a viable and *instrument-general* route to fast, accurate articulated-pose feedback for robotic microsurgery, with the short-linked jaw remaining the main frontier for further accuracy and larger cross-validated datasets the main next step. The honest boundary of the present evidence is the size of the labelled sets: with 30 and 13 held-out frames respectively, and with the final configuration not yet cross-validated, the numbers reported here establish feasibility rather than a tight bound on accuracy.

References

[1] Erik Fredin et al. "OCT Imaging for Pose Estimation and Feedback Control of an Articulated Magnetic Surgical Tool". In: *IEEE Robotics and Automation Letters* 11.2 (2026), pp. 1818–1825. URL: https://medcvr.utm.utoronto.ca/ral2025-oct-pose.html.

[2] Yukang Chen et al. "VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 21674–21683. URL: https://openaccess.thecvf.com/content/CVPR2023/papers/Chen_VoxelNeXt_Fully_Sparse_VoxelNet_for_3D_Object_Detection_and_Tracking_CVPR_2023_paper.pdf.