

# Point Cloud Segmentation for Autonomous Clip Positioning in Laparoscopic Cholecystectomy on a Phantom

Balázs Gyenes<sup>1,2</sup>, Nikolai Franke<sup>1</sup>, Paul Maria Scheikl<sup>3</sup>, Pit Henrich<sup>3</sup>, Rayan Younis<sup>4</sup>,  
Gerhard Neumann<sup>1</sup>, Martin Wagner<sup>4</sup>, Franziska Mathis-Ullrich<sup>3</sup>

**Abstract**—High-risk applications in robotics, such as robot-assisted surgery, present unique challenges. These systems must be both highly precise and interpretable in order to be deployed in environments with very low tolerance for error or unsafe exploration. We present the first robotic system to demonstrate autonomous clip positioning on a physical phantom in laparoscopic surgery, one of the most common interventions in general surgery. After segmentation of a colorless point cloud from a single camera, target positions for the clips are extracted using spline interpolation, and can then be adjusted by the human operator. The segmentation model is trained on only 60 hand-labeled real point clouds, reflecting data scarcity in the surgical domain. We overcome this with a combination of pre-training on 128,000 synthetic point clouds and two novel data augmentation techniques. The motion of the end-effector to each target is visualized for the operator, satisfying the unique motion constraints of minimally-invasive surgery while ensuring that the robot’s actions are verifiable and interpretable. In real robot experiments, our system localizes targets with the required precision of 0.75 mm at a 95% success rate and executes autonomous clip positioning with a 100% success rate. We provide insights that are applicable to many other surgical and non-surgical tasks that require identifying and navigating to a precise target. Our source code is available at <https://github.com/balazsgyenes/kirurc>.

**Index Terms**—Computer Vision for Medical Robotics; Surgical Robotics; Laparoscopy; Transfer Learning

## I. INTRODUCTION

RECENT years have seen many machine learning applications in robotics, especially with large generative models. Although these systems can generalize across many simple tasks, high-risk applications require movements that are precise, yet interpretable and predictable, two characteristics

Manuscript received: January, 27, 2025; Revised May, 15, 2025; Accepted June, 25, 2025.

This paper was recommended for publication by Editor Pietro Valdastrì upon evaluation of the Associate Editor and Reviewers’ comments. This work was supported by the Ministerium für Wirtschaft, Arbeit und Wohnungsbau, Baden-Württemberg, Germany, the Helmholtz Association under the joint research school “HIDSS4Health – Helmholtz Information and Data Science School for Health”, and by the German Research Foundation (Deutsche Forschungsgemeinschaft) as part of Germany’s Excellence Strategy – EXC 2050/1 – Project ID 390696704 – Cluster of Excellence “Centre for Tactile Internet with Human-in-the-Loop” (CeTI) of Technische Universität Dresden.

<sup>1</sup>Institute for Anthropomatics & Robotics - Karlsruhe Institute of Technology, Germany

<sup>2</sup>HIDSS4Health - Helmholtz Information and Data Science School for Health, Karlsruhe/Heidelberg, Germany

<sup>3</sup>Department of Artificial Intelligence in Biomedical Engineering, Friedrich-Alexander-University, Erlangen-Nuremberg, Germany

<sup>4</sup>University Hospital Carl Gustav Carus and Centre for Tactile Internet with Human-in-the-loop (CeTI), Dresden University of Technology, Germany

Digital Object Identifier (DOI): see top of this page.

©2026 IEEE

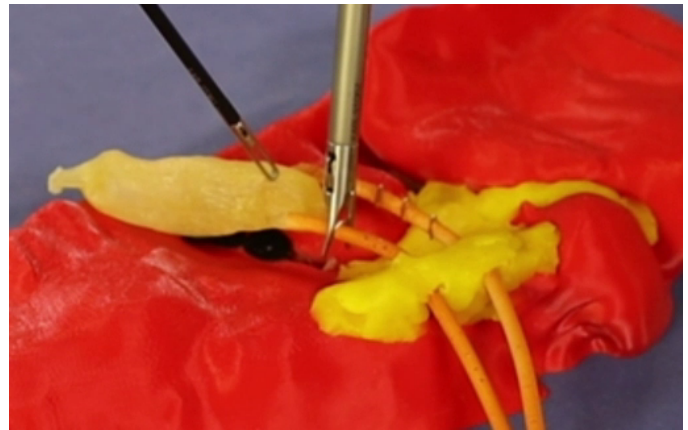


Fig. 1: We present a robotic system for autonomous clip positioning in laparoscopic cholecystectomy. At this stage of the surgery, the cystic duct and cystic artery (■) have been dissected from the surrounding fatty tissue (■) and are ready for ligation. Ligation is performed using clips applied by a laparoscopic clip applier (right tool). Next, the cystic duct and artery are cut to allow the gallbladder (■) to be resected from its bed in the liver (■). Gripper (left tool) provides tension.

not typical of large generative models. One such high-risk application is Robot-Assisted Surgery (RAS), where (semi-) autonomous policies have the potential to address the shortage of skilled personnel. Semi-autonomous RAS systems must be interpretable, since surgeons are unlikely to trust a system they cannot anticipate, and also extremely precise due to the small scale of relevant anatomical structures. Furthermore, data scarcity is a wide-spread problem in medical contexts, so algorithms must generalize to unseen anatomies. Laparoscopic cholecystectomy (LC) is one of the most common interventions in general surgery [1], so it provides a good case study for challenges that arise in RAS and in other high-risk robotic applications. During the clipping phase of LC, 3 clips are placed on the cystic artery and the cystic duct each. Accurately positioning the clips is a challenging task and a critical step for a successful surgery.

We present an autonomous learning-based robotic system to carry out clip positioning in laparoscopic surgery, demonstrating our method in the clipping phase of LC on a realistic 3D-printed phantom (see Figure 1). In our phantom, the cystic artery and cystic duct are 5 mm in diameter, but each clip is only 6.5 mm wide, such that it must be placed with sub-

millimeter precision. Generalist end-to-end approaches, such as reinforcement learning (RL), imitation learning (IL), or end-to-end prediction of the target positions, lack the required precision and may also not be interpretable. To achieve both these requirements, we instead opt for a point cloud-based vision system that segments target regions and predicts targets in the form of B-splines. A motion planner can then generate a complete trajectory to each target point in an interpretable way. This corresponds to task-level autonomy as defined by Haidegger et al. [2], as the system independently generates plans to perform clip positioning, with human input only required to approve the plan and optionally adjust it.

The majority of image-based RAS systems that use 3D information treat the depth map as an additional channel to the RGB image (RGBD). Due to hardware limitations and the homogeneous and specular environment, the depth estimate is often noisy or inaccurate, so the RGBD image is rarely converted to a point cloud. However, point clouds have several advantages over RGBD images, despite nominally containing the same information. Point cloud models are more robust to changes in camera perspective [3], and tend to exhibit better sim-to-real transferability because point clouds focus on geometry rather than texture [4]. This allows us to mitigate the data scarcity by pre-training on a large amount of synthetic data generated in simulation and fine-tuning using a small, manually labeled, real-world dataset. Approaches that rely on RGB data require accurately rendering the colors and textures of the real scene, which is notoriously difficult.

In summary, we present the first system to demonstrate autonomous clip positioning in laparoscopic surgery on a physical phantom. Rather than regressing 3D target positions directly, our key insight is to segment the target regions, skeletonize them, and fit them to B-splines, leveraging the accuracy of the 3D camera. This results in a system which is precise, while remaining interpretable and human-correctable. A human operator is able to visually validate or modify the target positions and, because the trajectory is computed in advance, they are able to confirm its safety before execution. Despite using only 60 hand-labeled data points for training, reflecting the data scarcity in the surgical domain, we demonstrate a 95% success rate at localizing targets within a tolerance of 0.75 mm. To achieve this, we pretrain the segmentation model on a large synthetic dataset and leverage a set of straightforward but essential algorithmic contributions: i) We demonstrate that pre-training with extra segmentation classes that are not present in the fine-tuning data help the model learn the semantic structure of the scene, but only if the extra classes are merged in *logit space* rather than probability space. ii) We introduce two new forms of point cloud data augmentation: `VariableJitter`, allowing random jitter to be applied without introducing a distribution gap; and `RandomPatches`, which provides robustness against missing points due to occlusions. We validate our design choices post-hoc on a disjoint, real-world test set that was unavailable at training time.

## II. RELATED WORK

### a) *Deep Learning for Segmentation & Control in RAS:*

Point clouds are under-explored in RAS, partly because of the limited accuracy of 3D cameras with a form factor that can fit through a conventional 10-12 mm trocar. However, a number of works use RGB or RGBD information to identify a structure of interest and control an end-effector to interact with it. Tanzi et al. [5] segment an endoscopic image to align it to an ad-hoc 3D virtual model, achieving a system with an error of 4.1 mm. Attanasio et al. [6], [7] demonstrate tissue segmentation in abdominal surgery using RGBD images from a stereo endoscope. They construct a custom step-based planning algorithm to grasp and retract a tissue flap. Qi et al. [8] use a 3D U-Net architecture to segment preoperative CT images, after which the key points are detected. A surgical path is pre-planned using a hand-crafted algorithm. Schwaner et al. [9] learn a policy for autonomous surgical suturing from demonstrations and achieve a mean needle insertion error of 3.8 mm. Movements are encoded as Dynamic Movement Primitives, which offers a degree of interpretability compared to step-based inference. Most similarly to our work, Lu et al. [10] develop a system to autonomously grasp a suture thread after identifying its 3D structure. They fine-tune a model pre-trained on diverse data and achieve an average error between 2.4 mm and 4.5 mm.

b) *Point Clouds in RAS:* A limited number of prior works operate directly with point clouds in the context of RAS. There are several examples of vision systems that use point cloud features to estimate 3D state, such as point-wise displacements of facial bone models for the planning of corrective jaw surgery [11] or estimating the 6D pose of a needle [12]. Li et al. [13] create a stereo vision-based perception system that estimates the 3D state of the environment as surfels, which can easily be converted to point clouds. Thach et al. [14] learn a servoing policy from demonstrations using an observed point cloud and a goal point cloud as inputs, but only achieve an accuracy of  $\sim 40$  cm. The STAR system [15] autonomously performs intestinal anastomosis in vivo on porcine models, relying on added visual markers and several task-specific algorithms and datasets, including 9294 labeled data points for learning tissue motion tracking.

c) *Point Cloud Segmentation:* Point cloud segmentation is a well-studied task going back to the advent of PointNet [16]. Since then, a number of architectures have been proposed for segmentation as well as classification of point clouds. These architectures make use of several paradigms, including graph neural networks [17], graph convolutions [18], and transformers [19], [20]. PointTransformer [17] is a graph-based model that makes use of a sequence of downsampling and upsampling layers, and achieves state of the art performance on segmentation of standard point cloud datasets.

## III. METHODS

During laparoscopic cholecystectomy (LC), the cystic duct and cystic artery (the *target structures*) must be transected (cut)

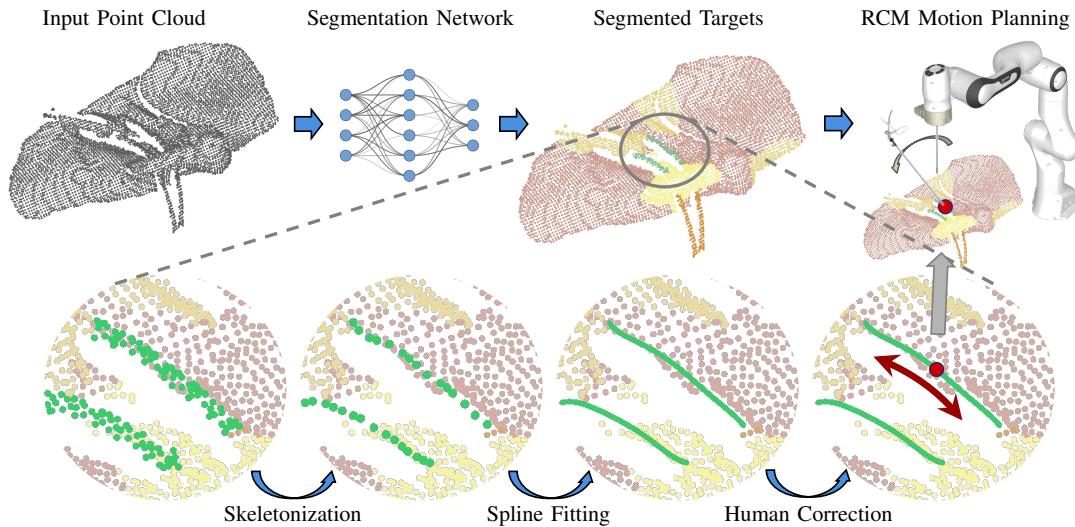


Fig. 2: Schematic of inference pipeline, showing how the input point cloud is segmented to isolate the target regions. Liver (■), cystic duct and artery (■), gallbladder (■), and fatty tissue (■) are colored as in the real scene, while target regions (■) are bright green for clarity. To increase robustness against outliers in the segmentation, each target region is then skeletonized and fitted to a B-spline. To ensure safe execution and accommodate a surgeon’s preference, the human operator is able to customize the target position (■) along the interpolated spline.

before the gallbladder can be removed. To prevent postoperative complications caused by bleeding or bile leaking into the abdominal cavity, these structures must be sealed, where best practice is to place 3 surgical clips on each vessel prior to cutting. While guidelines for clip placement vary across surgical centers, the areas of the cystic duct and artery where clips may be applied are referred to as *target regions*.

We develop a robotic system to autonomously position surgical clips (see Figure 2), while addressing several unique challenges. First, the target positions must be predicted with sub-millimeter accuracy, otherwise the clip would not intersect with the target structure. However, the accuracy along the longitudinal axis of the target structures is less critical, provided the clip is a safe distance from the common bile duct. The ordering of the six clips is also not strictly defined, allowing for flexibility in execution. Furthermore, the system must provide explicit plans that the surgeon can review to ensure safe operation. Finally, due to the absence of publicly available datasets for this task, and the overall data scarcity in the surgical domain, it must be highly data efficient.

While it is possible to directly predict the coordinates of the six targets with an end-to-end vision model, this method is presented as an ablation and fails to reach the required accuracy. Instead, our approach leverages the accuracy of the point cloud camera by training a model to segment the target regions within the point cloud. Each target region is fitted to a B-spline, from which three target points are selected by varying the spline parameter. Compared to selecting a single point as the target, spline interpolation can be thought of as averaging over multiple points, resulting in a more robust estimation of the true surface. Using splines as targets also allows the system to predict an optimal angle of approach for the clip by computing the tangent to the spline at the target location.

During execution, the surgeon retains the ability to decide which clip to target next and adjust the target position along the spline as needed. This approach shifts the model’s task from direct target prediction to a more well-defined segmentation problem, while allowing the surgeon to control task-specific degrees of freedom and make necessary corrections.

*a) Experimental Setup:* A phantom scene is constructed containing a liver, a gallbladder, the cystic duct and artery, and fatty tissue (see Figure 1). This scene represents the phase of the surgery after the surgeon has dissected the tissue covering the cystic duct and artery (Calot’s triangle) and positively identified these structures. The liver is based on an anonymized CT scan of a real patient at Heidelberg University Hospital and is 3D printed from red ABS at 1.15X scale. A gallbladder mold is printed based on a scan of the same patient from SLA material. Several gallbladder phantoms are then produced by painting liquid latex rubber onto the mould and peeling off the dried material. The cystic duct and artery are made of silicone tubing with outer diameter 5 mm. For attaching the tubing to the gallbladder, angled *rodlets* are 3D printed from SLA, where the angle determines the position and shape of the target structures.

Point clouds are taken with a 3D camera (Zivid One+ M, Zivid AS) which achieves a precision of  $<0.1$  mm. A laparoscopic clip applicator (model 30444LR, Karl Storz SE & Co. KG) is mounted to a 7-DoF robot arm (Franka Panda, Franka Robotics GmbH) such that the axis of the tool is perpendicular to the axis of the end-effector. A laparoscopic grasper (model 33356ON, Karl Storz SE & Co. KG) is additionally used to grasp the neck of the gallbladder to apply tension, and is fixed in place during clipping. A 3D-printed ring is used to visually mark the remote center of motion (RCM), where in a surgery the trocar would be located.

b) *Synthetic and Real Datasets*: To address the data scarcity for this specialized task, we pre-train the model on synthetic data and fine-tune it using a small, manually labeled, real-world dataset. To minimize the sim-to-real gap, we use point clouds without color features.

A synthetic dataset is created in Blender, an open-source 3D modeling and animation software. Rather than attempt to replicate the setup in Blender exactly, we deliberately simplify the scene to demonstrate the robustness of our method across a significant sim-to-real gap. We reuse the scans of the patient’s liver and gallbladder for the virtual scene. To introduce variability, random deformations are applied to the liver, gallbladder, and fatty tissue using a displacement field. The pose of the gallbladder neck is randomized, with the target structures following B-spline curves. Additionally, the diameters of the target structures are varied between 2 mm and 4 mm. For simplicity, collisions between objects are ignored.

To ensure robustness to changes in the camera pose, we randomize the virtual camera with an incline from the range  $[30^\circ, 70^\circ]$  and distance from the scene from the range  $[30, 40]$ cm. Segmented depth images are rendered using the virtual camera and converted to point clouds, and each point is labeled with its class of origin: **Liver/Background, Gallbladder, Fatty Tissue, Cystic Duct, or Cystic Artery**. Points on the cystic duct or cystic artery located within a distance of 30 mm from the middle of each structure are relabeled as **Target Duct** or **Target Artery**, respectively. Although we are only interested in the target regions, we show in our ablations that adding extra classes helps the network to learn a fine-grained understanding of the scene and promotes generalization. We generate 128,000 point clouds with roughly 4650 points each. The point clouds undergo preprocessing through voxel downsampling with a 3 mm grid.

For the fine-tuning dataset, we record 60 point clouds from the real experimental setup. For each point cloud, we vary the liver pose, the distribution of fatty tissue, the grasping position of the gripper, and the gallbladder pose, while maintaining the same camera angle. The point clouds are segmented manually in CloudCompare, an open-source software. To minimize labeling effort, the real dataset only contains the **Background, Target Duct, and Target Artery** labels. Each point cloud is pre-processed by cropping, removing statistical outliers, removing the plane of the table using RANSAC, and voxel downsampling with a 3 mm grid.

c) *Pre-Training*: We train a neural network to segment point clouds into the seven classes defined above. We choose the segmentation variant of PointTransformer [17] because of its state-of-the-art performance on standard benchmark datasets. The neural network is trained for 30 epochs with a batch size of 48 and an exponentially decaying learning rate of  $\alpha(t) = \alpha_0 \cdot \beta^t$ , where  $\alpha_0 = 5 \cdot 10^{-4}$ ,  $\beta = 0.98$ , and  $t$  is the epoch. The class weights in the cross-entropy loss are adjusted according to the frequency of each class in the dataset, to ensure there is no bias towards large classes.

On each forward pass, we apply data augmentation to the

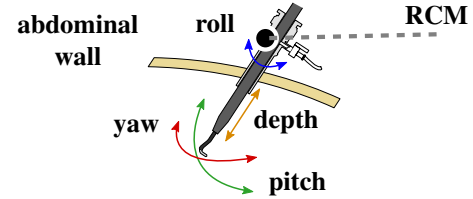


Fig. 3: In laparoscopic surgery, instruments are inserted into the abdomen through trocars, which imposes a pivoted motion constraint. This reduces the end effector’s independent degrees of freedom to 3 rotations (roll, pitch, & yaw) and one translation (depth) around the remote center of motion. Illustration adapted from [22].

point clouds to increase the ability of the network to generalize. Since we found the real scene to contain several occlusions due to the RCM ring or the grasper tool, we apply a novel form of data augmentation called *RandomPatches* inspired by similar methods in the image domain [21]. For each point cloud, this process chooses a random number of patches to remove between 0 and 30, and a random patch size  $k$  between 0 and 200 points. The patch centers are sampled uniformly from the point cloud, and the patches are the  $k$  nearest neighbors to the patch center. The patches (including the patch centers) are removed from the point cloud. Next, a random rotation from the range  $[0^\circ, 90^\circ]$  is applied in each of the three axes, followed by a random shear with magnitude from  $[0, 0.3]$ , and a random stretch with magnitude from  $[0, 0.3]$ . Finally, we apply a novel augmentation method called *VariableJitter*. While adding random jitter to a point cloud tends to improve model robustness, standard jitter with a fixed scale always creates point clouds with the same expected noise, creating a challenging trade-off. Using a larger jitter scale increases robustness (up to a certain point), but also creates a significant domain gap compared to point clouds recorded by the Zivid camera, which are highly precise and orderly. To address this, *VariableJitter* dynamically adjusts the jitter scale for each point cloud. It samples the jitter scale  $d_i$  for the entire point cloud from the range  $[0, d_{\max}]$ , where  $d_{\max} = 0.04$ , and the displacement of each point is then sampled from  $[0, d_i]$ . This ensures that some augmented point clouds maintain low noise, better matching the characteristics of the real data.

d) *Fine-Tuning*: After pre-training, the neural network is fine-tuned on the real dataset for an additional 50 epochs with a batch size of 8 and a reduced initial learning rate of  $\alpha_0 = 2.5 \cdot 10^{-4}$ . In order not to overfit on the limited fine-tuning data, only the MLP head and the three upsampling layers (each composed of a TransitionUp layer and a transformer block) are optimized during fine-tuning. Real point clouds undergo weaker data augmentation. We use random rotations from the range  $[0^\circ, 45^\circ]$ , random shear and random stretch from the range  $[0, 0.2]$ , and *VariableJitter* with maximum displacement  $d_{\max} = 0.01$ .

Since the fine-tuning dataset has fewer classes (three) than the neural network has outputs (seven), we investigate two

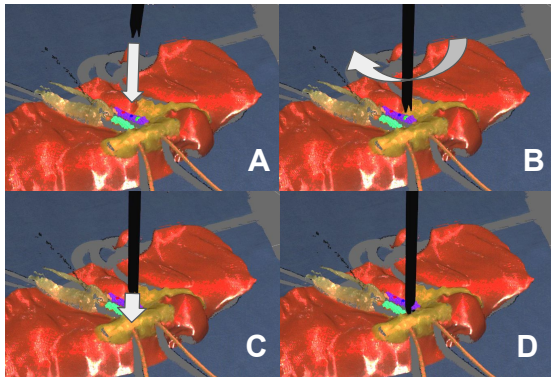


Fig. 4: The clip is applied onto the target structure in three phases. A) navigate to within  $\sim 2$  cm of the target, B) rotate about clipper such that insertion is not obstructed, and C) final insertion, placing clip around target structure.

methods for calculating the cross-entropy loss despite this limitation. All classes except for the two target regions must be merged into a single background class. One approach is to combine the logits in probability space using  $\log\text{-sum-exp}$ , which we call *Probability-Merging*. A simpler approach is to directly sum the logits before converting to probabilities, which we call *Logit-Merging*. Depending on the absolute scale of the logits, this can distort the loss landscape. For example, with negative logit values, the probability of the merged class would be less than the probability of any of the individual classes. Counterintuitively, we show in our ablations that *Logit-Merging* results in the highest accuracy.

e) *Target Localization*: The points belonging to each target region are skeletonized using Laplacian-Based Contraction [23]. Before interpolation, the points must be sorted along the major axis of the skeleton, the direction of which is calibrated based on the layout of the scene. The sorted points are then fitted to a B-spline of degree 3 using a smoothing factor of  $5 \cdot 10^{-4}$ .

A simple user interface allows a human operator to approve the computed target positions and adjust them if needed. This allows the surgeon to adjust the desired distance between the two distal clips that remain in the body, or to correct minor errors resulting from the segmentation. The UI also allows the operator to decide the order of the clips.

f) *Metrics*: We report the accuracy of target localization using five metrics. All metrics are computed across an unseen real-world test set collected during experiments on the real phantom, as described in Section IV.

To measure segmentation performance, we report mIoU, the mean Intersection over Union across all classes. We also report *Target mIoU*, the mean Intersection over Union across only the two target classes, since they are the most relevant for target localization. The *Target mIoU* allows for a direct comparison between synthetic and real data, which is not possible with mIoU since they have different numbers of classes. If too few points are segmented in a target region ( $< 15$  points), the target is considered not to have been found, and no spline

is generated. The *Spline Found Rate* expresses the fraction of target regions for which a spline is successfully found.

To quantify the similarity of the predicted spline  $S_{\text{pred}}$  to the reference spline  $S_{\text{ref}}$ , we discretize each and compute the asymmetrical chamfer distance between them. First, we evaluate  $N = M = 200$  evenly-spaced points along each spline to create point clouds  $P_{\text{ref}} = \{x_i \in \mathbb{R}^3\}_{i=1}^N$  and  $P_{\text{pred}} = \{x_i \in \mathbb{R}^3\}_{i=1}^M$ . We remove 10% from each end of the reference spline to focus on likely clip targets. The asymmetrical chamfer distance between two point clouds is

$$\text{chamfer}(P_{\text{ref}}, P_{\text{pred}}) = \frac{1}{N} \sum_{i=1}^N \|x_i - \mathcal{NN}(x_i, P_{\text{pred}})\|, \quad (1)$$

where  $\mathcal{NN}(x, P) = \operatorname{argmin}_{x' \in P} \|x - x'\|$  is the nearest neighbor to  $x$  in the point cloud  $P$ . We choose the asymmetrical chamfer distance because it quantifies the deviation of the reference spline from the predicted spline, while not punishing the model if parts of the predicted spline are far from the reference. In practice, this measures how close a human operator could get to a desired target point on the reference spline while being constrained to the predicted spline.

*Success Rate* reports the fraction of predicted splines with a chamfer distance below a threshold of 0.75 mm. This threshold is half the clearance between target structure (5 mm) and the clip (6.5 mm), and does not take into account errors in camera calibration, motion planning, or robot control. The success rate is averaged across all target regions, where if no spline is found, the target is considered unsuccessful.

g) *Motion Planning*: First, we estimate the static transformation required to transform the target positions from camera coordinates into the robot’s reference frame. This is obtained via least-squares regression using 12 pairs of corresponding points from both coordinate systems. These correspondences are found by positioning the clip applicator at the corners of a calibration checkerboard visible in the camera frame.

A key aspect of robot-assisted minimally invasive surgery is to satisfy the remote center of motion constraint about the trocar, which reduces the number of independently controllable DoFs from 6 to 4 (shown in Figure 3). Therefore, we opt to use a task space where the degrees of freedom are the same as those of a laparoscopic instrument inside a trocar: three rotations, roll, pitch, yaw; and one translation along the instrument axis, depth. We plan the motion to the target positions in task space and compute the corresponding trajectory in Cartesian space. The resulting trajectory naturally satisfies the constraints of pivoted motion around a remote center of motion, and is densely sampled so that it can be used directly for Cartesian control of the robot.

In order to ensure the clip is properly aligned when it approaches the target structure, the clipping motion is divided into 3 stages, approach, align, and insert, as shown in Figure 4. A single waypoint a distance of 2.3 cm (the length of the clip applicator) from the target ensures that the alignment is step

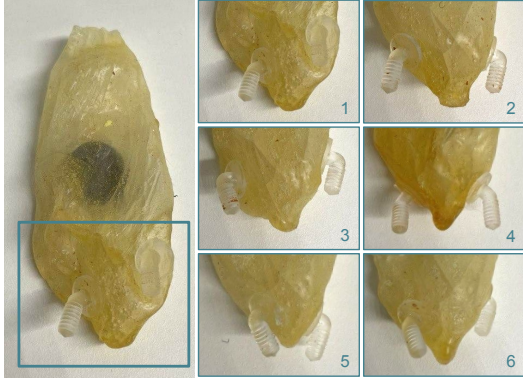


Fig. 5: Different gallbladder phantoms used for real world experiments. The cystic duct and artery are mounted at unique positions and angles on each gallbladder through the use of the clear plastic rodlets highlighted by the teal box. The numbered insets show the variation between gallbladders.

finished and the clip is correctly rotated before the insertion step begins. Each planned motion is verified by the human operator before execution. After insertion, the clipping itself is actuated manually by the human operator.

#### IV. RESULTS AND DISCUSSION

*a) Clipping on Real Phantom:* We demonstrate our method for autonomous clip positioning on the real-world setup described above. A medical student who has completed a laparoscopic training curriculum acts as the operator. We apply six clips each to six novel gallbladder phantoms with different morphologies not seen in the fine-tuning data (see Figure 5). In particular, the angles of the target structures to the gallbladder are varied by using rodlets with different angles, resulting in novel point clouds not seen during training.

Out of the 36 clips, the operator identified possibly incorrect segmentations in 5 cases. In each of these instances, the operator repeated point cloud acquisition and model inference, which then produced a correct segmentation and target localization, partly due to the stochastic nature of model inference. This implies a success rate of  $\frac{36}{36+5} = 87.8\%$  for target localization, though this should be considered a lower bound. It is possible that some of the incorrect segmentations could have still led to successful clipping, especially if the operator had adjusted the target points. In our experiments, the operator never adjusted target points from their default positions. For cases with successful segmentations, we report a 100% success rate for autonomous clip positioning, with 36/36 clips applied correctly. Given the dimensions of the clips and the phantom, this implies a total error of  $<1$  mm across target localization, camera calibration, motion planning, and robot control.

During testing, we record 2 point clouds taken before each clip, which are then manually labeled to create reference segmentations. The reference target regions are skeletonized and fitted to splines to create a test dataset of 72 point clouds. This test dataset is used to evaluate post-hoc how several ablations of our method would have performed during

TABLE I: Ablation Pre-Training & Fine-Tuning

	mIoU	Target mIoU	Chamfer Distance	Success Rate
Pretrained only <sup>†</sup>	0.884	0.809	-	-
<b>Logit-Merging</b>	<b>0.744</b>	<b>0.619</b>	<b>0.3 mm</b>	<b>94.9%</b>
Probability-Merging	0.481	0.239	1.3 mm	41.4%
No Additional Classes	0.534	0.313	0.8 mm	63.3%
No Pre-Training	0.346	0.096	12.0 mm	7.6%

<sup>†</sup> On a held-out synthetic test dataset of 12,800 point clouds

real world experiments, allowing us to evaluate many more variations than would be feasible on the real robot. These *in silico* results are reported as a mean over 20 random seeds for fine-tuning, and no data augmentation is applied during evaluation. When evaluated on this dataset, our method achieves an mIoU of 0.744, a target mIoU of 0.619, a chamfer distance of 0.3 mm, and a success rate of 94.9%, which is indeed slightly higher than the observed success rate.

*b) Target Point Regression:* To validate our segmentation-based approach, we present an ablation where the six target positions are directly predicted by the network. Reference target positions are computed for each point cloud in the synthetic and real datasets by evaluating each reference spline at points 35%, 50% and 65%. We modify the classification variant of PointTransformer to regress 18 values, representing the xyz coordinates of the six target points. The symmetric chamfer distance is used as the loss function for both pre-training and fine-tuning, since the regressed target points are interchangeable. Because this network does not have upsampling layers, we opt to unfreeze the entire network for fine-tuning. We fine-tune for 100 epochs instead of 50, since convergence takes longer.

We report a mean distance between the reference and predicted targets of  $(5.7 \pm 5.2)$  mm. Applying a wider threshold of 1 mm for success, this results in a success rate of 1%. However, this evaluation is simplified, since predicting only target positions would leave the angle of approach undefined. Adding pose prediction to the network is possible but would likely further reduce accuracy. Therefore, target point regression cannot achieve the required accuracy for this task.

*c) Pre-Training Dataset:* We investigate two methods for merging classes when fine-tuning on a dataset with fewer classes than the model has outputs, *Logit-Merging* and *Probability-Merging*. We compare this against pre-training with no additional classes, such that the network only predicts the classes *Background*, *Target Duct*, and *Target Artery*, while the remaining classes are all merged into the *Liver/Background* class. We additionally compare against no pre-training at all. To keep the comparison fair, the network still predicts 7 classes, and we merge additional classes using *Logit-Merging*. We also train for 100 epochs instead of 50, since it takes longer for the network to converge.

The results are summarized in Table I. *Logit-Merging* far outperforms *Probability-Merging*, mainly due to a much lower chamfer distance (0.3 mm vs. 1.3 mm). Pre-training without additional classes results in a chamfer distance of 0.8 mm,

TABLE II: Ablation of Data Augmentation Methods

	mIoU	Target mIoU	Spline Found Rate	Chamfer Distance	Success Rate
<b>All</b>	<b>0.744</b>	<b>0.619</b>	<b>99.6%</b>	<b>0.3 mm</b>	<b>94.9%</b>
w/o Shear/Stretch	0.727	0.593	97.3%	0.6 mm	81.1%
w/o VariableJitter	0.631	0.451	86.6%	1.0 mm	62.5%
w/o RandomPatches	0.635	0.533	89.0%	1.0 mm	58.8%
w/o Jitter	0.396	0.099	11.4%	1.7 mm	4.9%
w/o Rotations	0.330	0.0	0.0%	N/A	0.0%
w/o Augmentation	0.467	0.205	30.0%	1.9 mm	16.8%
PointWOLF instead of VariableJitter	0.632	0.452	76.6%	1.2 mm	61.3%
PointWOLF instead of RandomPatches	0.686	0.531	78.6%	1.2 mm	42.4%

which is better than *Probability-Merging*, although it still only results in a success rate of 63.3%. *Logit-Merging* shows a meaningfully higher Target mIoU of 0.619 compared to *Probability-Merging* (0.239) or training without additional classes (0.313), which is reflected in its high success rate. Training only on the real dataset without any pre-training, in contrast, achieves only a 7.6% success rate, despite training for 100 epochs instead of 50.

d) *Data Augmentation*: Next, we show that data augmentation is essential for allowing the network to generalize across the distribution gap between the real and synthetic data. Since data augmentation can be applied independently during fine-tuning and pre-training, the space of augmentation combinations is large. We choose to disable individual augmentation steps in both pre-training and fine-tuning simultaneously, testing the overall contribution of each method. Additionally, we validate our novel augmentation methods *VariableJitter* and *RandomPatches* by replacing them with PointWOLF [24], a general-purpose augmentation method that applies a random deformation field to each point cloud. Note that *RandomPatches* is never enabled during fine-tuning. The ablation “w/o VariableJitter” substitutes standard random jitter instead of *VariableJitter*, whereas “w/o Jitter” disables all forms of jitter completely.

The results are summarized in Table II. Replacing *VariableJitter* with standard jitter decreases the success rate to 62.5%, making this an essential element of the overall system success. We hypothesize that this was particularly important due to the extremely low noise of the Zivid camera, which produces point clouds that look very dissimilar to typical jittered point clouds. Removing *RandomPatches* reduces the success rate to only 58.8%, likely because the model is less robust to missing regions of the point cloud caused by occlusions. PointWOLF does not address either of these challenges, and is therefore not able to achieve the same level of success.

Removing either random jitter or random rotations, while keeping all other augmentation methods unchanged, renders the model completely unable to learn the fine-tuning data. During pre-training, these models achieve Target mIoUs of 0.889 and 0.736, respectively, on the training data; however, during fine-tuning, their Target mIoUs on the training data drop

TABLE III: Ablation of Network Architecture

	mIoU	Target mIoU	Chamfer Distance	Success Rate
PointNet++ [25]	0.489	0.249	1.1 mm	40.3%
Stratified Transformer [26]	<b>0.796</b>	<b>0.696</b>	0.4 mm	83.1%
<b>PointTransformer [17]</b>	0.744	0.619	<b>0.3 mm</b>	<b>94.9%</b>

TABLE IV: Ablation of Size of Synthetic Pre-Training Dataset

Dataset Size	mIoU	Target mIoU	Chamfer Distance	Success Rate
10%	0.714	0.574	1.3 mm	60.7%
50%	0.732	0.601	0.5 mm	88.2%
<b>100%</b>	<b>0.744</b>	<b>0.619</b>	<b>0.3 mm</b>	<b>94.9%</b>

to 0.063 and 0.0. This indicates an inability to generalize across the sim-to-real gap. Notably, these configurations perform even worse than training without data augmentation at all, which yields a 16.8% success rate.

e) *Network Architecture*: We investigate the effectiveness of several common neural network architectures for point cloud segmentation. PointNet++ [25] introduced hierarchical processing of point clouds. Stratified Transformer [26] is able to capture long-range contexts by sampling both nearby and distant points as keys for each query point. Stratified Transformer appears to perform better if the entire network is fine-tuned, so this is the configuration we report here.

The results are summarized in Table III. PointNet++ is hindered by poor segmentation performance (e.g. Target mIoU of 0.249), leading to low success. Although Stratified Transformer performs well under the mIoU and Target mIoU metrics, which indicate good segmentation performance, its success rate is lower than Point Transformer. We observe that this is due to a lower Spline Found Rate (91.1% vs. 99.7%), which is not captured by the mIoU, in which empty classes are ignored when taking the mean. Overall, we hypothesize that PointTransformer may be better at generalization when the fine-tuning dataset is extremely small.

f) *Fine-Tuning*: Since PointTransformer consists of downsampling and upsampling layers in a U-Net-like configuration, we investigate different configurations of freezing or training network weights during fine-tuning. We test unfreezing 1, 2, or all 3 of the upsampling layers, as well as fine-tuning the entire network. We find that fine-tuning all 3 upsampling layers is substantially better than 1 or 2 layers, which achieve success of 2.2% and 8.1%, respectively. Fine-tuning the entire network results in a chamfer distance of 1.9 mm and the success rate drops to 59.3%.

g) *Size of Synthetic Dataset*: Finally, in Table IV, we describe the effect of reducing the amount of synthetic data for pre-training. Greater amounts of synthetic data are indeed beneficial, but the returns are diminishing. There is a small but meaningful improvement between 64,000 and 128,000 samples. Using only 12,800 samples results in noticeably worse performance, with only an 60.7% success rate.

## V. CONCLUSIONS

We present the first system to achieve autonomous clip positioning in laparoscopic surgery on a physical phantom using point clouds. Our segmentation-based approach leverages the precision of the 3D camera, making it possible to localize targets with the required accuracy of  $<1$  mm, while retaining interpretability. We use motion planning to ensure that the remote center of motion (RCM) constraint imposed by the trocar is satisfied and that the system remains interpretable. We demonstrate our method with a real robot on six gallbladder phantoms and report success rates of 95% for target localization and 100% for autonomous clip positioning. We validate our algorithmic contribution post-hoc on the real robot data: pre-training with additional semantic classes, and *VariableJitter* and *RandomPatches* for data augmentation. Our contributions can be applied to other robotic applications where accuracy and interpretability are core challenges, and represent a significant step towards translating the promise of machine learning for surgical robotics to the clinical reality.

*a) Limitations and Future Work:* While the proposed robotic system demonstrates high accuracy and interpretability, certain limitations remain. Notably, the Zivid camera, selected for its exceptional accuracy, is not suitable for clinical deployment due to its physical size. Initial experiments with a more compact and widely available stereo camera (ZED Mini, StereoLabs Inc.) revealed that its limited accuracy was insufficient for our method, which currently cannot compensate for systematic offsets in observed point positions. Future work could explore integrating 3D reconstruction methods [27] to infer true point positions from noisy observations, potentially reducing the accuracy required of the 3D camera. Additionally, the motion planner does not take collision avoidance into account or respond to changes in the environment, so the human operator is responsible for verifying the safety of the planned motions and initiating re-planning when necessary.

## VI. ACKNOWLEDGMENTS

Beyond simple editing and grammar enhancement with ChatGPT, no generative AI tools were used to create this manuscript, including text, figures, and images.

## REFERENCES

- [1] E. A. Shaffer, "Epidemiology of gallbladder stone disease," *Best Practice & Research Clinical Gastroenterology*, pp. 981–996, Jan. 2006.
- [2] T. Haidegger, "Autonomy for Surgical Robots: Concepts and Paradigms," *IEEE Transactions on Medical Robotics and Bionics*, 2019.
- [3] S. Peri, I. Lee, C. Kim, L. Fuxin, T. Hermans, and S. Lee, "Point cloud models improve visual robustness in robotic learners," *ICRA*, 2024.
- [4] Y. Qin, B. Huang, Z.-H. Yin, H. Su, and X. Wang, "DexPoint: Generalizable Point Cloud Reinforcement Learning for Sim-to-Real Dexterous Manipulation," in *CoRL*, Mar. 2023.
- [5] L. Tanzi, P. Piazzolla, F. Porpiglia, and E. Vezzetti, "Real-time deep learning semantic segmentation during intra-operative surgery for 3D augmented reality assistance," *IJCARS*, Sept. 2021.
- [6] A. Attanasio, B. Scaglioni, M. Leonetti, A. F. Frangi, W. Cross, C. S. Biyani, and P. Valdastrì, "Autonomous Tissue Retraction in Robotic Assisted Minimally Invasive Surgery – A Feasibility Study," *RAL*, Oct. 2020.
- [7] A. Attanasio, C. Alberti, B. Scaglioni, N. Marahrens, A. F. Frangi, M. Leonetti, C. S. Biyani, E. De Momi, and P. Valdastrì, "A Comparative Study of Spatio-Temporal U-Nets for Tissue Segmentation in Surgical Robotics," *IEEE Trans. on Medical Robotics and Bionics*, Feb. 2021.
- [8] X. Qi, J. Meng, M. Li, Y. Yang, Y. Hu, B. Li, J. Zhang, and W. Tian, "An Automatic Path Planning Method of Pedicle Screw Placement Based on Preoperative CT Images," *IEEE Trans. on Medical Robotics and Bionics*, May 2022.
- [9] K. L. Schwaner, D. Dall'Alba, P. T. Jensen, P. Fiorini, and T. R. Savarimuthu, "Autonomous Needle Manipulation for Robotic Surgical Suturing Based on Skills Learned from Demonstration," in *CASE*, Aug. 2021.
- [10] B. Lu, W. Chen, Y.-M. Jin, D. Zhang, Q. Dou, H. K. Chu, P.-A. Heng, and Y.-H. Liu, "A Learning-Driven Framework with Spatial Optimization For Surgical Suture Thread Reconstruction and Autonomous Grasping Under Multiple Topologies and Environmental Noises," in *IROS*, Oct. 2020.
- [11] D. Xiao, C. Lian, H. Deng, T. Kuang, Q. Liu, L. Ma, D. Kim, Y. Lang, X. Chen, J. Gateno, S. G. Shen, J. J. Xia, and P.-T. Yap, "Estimating Reference Bony Shape Models for Orthognathic Surgical Planning Using 3D Point-Cloud Deep Learning," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 8, pp. 2958–2966, Aug. 2021.
- [12] M. Zhou, X. Hao, A. Eslami, K. Huang, C. Cai, C. P. Lohmann, N. Navab, A. Knoll, and M. A. Nasserì, "6DOF Needle Pose Estimation for Robot-Assisted Vitreoretinal Surgery," *IEEE Access*, 2019.
- [13] Y. Li, F. Richter, J. Lu, E. K. Funk, R. K. Orosco, J. Zhu, and M. C. Yip, "SuPer: A Surgical Perception Framework for Endoscopic Tissue Manipulation With Surgical Robotics," *RAL*, vol. 5, Apr. 2020.
- [14] B. Thach, B. Y. Cho, A. Kuntz, and T. Hermans, "Learning Visual Shape Control of Novel 3D Deformable Objects from Partial-View Point Clouds," in *ICRA*, May 2022.
- [15] H. Saeidi, J. D. Opfermann, M. Kam, S. Wei, S. Leonard, M. H. Hsieh, J. U. Kang, and A. Krieger, "Autonomous robotic laparoscopic surgery for intestinal anastomosis," *Science Robotics*, p. eabj2908, Jan. 2022.
- [16] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *CVPR*, 2017.
- [17] H. Zhao, L. Jiang, J. Jia, P. H. S. Torr, and V. Koltun, "Point Transformer," in *CVPR*, 2021.
- [18] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic Graph CNN for Learning on Point Clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 146:1–146:12, Oct. 2019.
- [19] Y. Pang, W. Wang, F. E. H. Tay, W. Liu, Y. Tian, and L. Yuan, "Masked Autoencoders for Point Cloud Self-supervised Learning," in *ECCV*, 2022.
- [20] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-BERT: Pre-Training 3D Point Cloud Transformers With Masked Point Modeling," in *CVPR*, 2022.
- [21] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.
- [22] P. M. Scheikl, B. Gyenes, R. Younis, C. Haas, G. Neumann, M. Wagner, and F. Mathis-Ullrich, "LapGym - An Open Source Framework for Reinforcement Learning in Robot-Assisted Laparoscopic Surgery," *JMLR*, 2023.
- [23] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su, "Point Cloud Skeletons via Laplacian Based Contraction," in *2010 Shape Modeling International Conference*, June 2010, pp. 187–197.
- [24] S. Kim, S. Lee, D. Hwang, J. Lee, S. J. Hwang, and H. J. Kim, "Point Cloud Augmentation With Weighted Local Transformations," in *ICCV*, 2021.
- [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 *NeurIPS*. Curran Associates, Inc., 2017.
- [26] X. Lai, J. Liu, L. Jiang, L. Wang, H. Zhao, S. Liu, X. Qi, and J. Jia, "Stratified transformer for 3d point cloud segmentation," in *CVPR*, 2022.
- [27] R. Peter, S. Moreira, E. Tagliabue, M. Hillenbrand, R. G. Nunes, and F. Mathis-Ullrich, "Stereo reconstruction from microscopic images for computer-assisted ophthalmic surgery," *IJCARS*, Mar. 2025.