2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Helical Control in Latent Space: Enhancing Robotic Craniotomy Precision in Uncertain Environments

Yuanyuan Jia[1], Jessica Ziyu Qu[2], Tadahiro Taniguchi[3]

***Abstract*— In this paper, we introduce a double-stage transfer learning framework based on expert data. It employs probabilistic graphical models to effectively capture helical periodic features in the latent space, integrating Bayesian variational inference and neural networks for implementation. Compared to traditional methods, it achieves high precision and stable control even in environments with limited observation signals and high noise levels. We have successfully applied this method to a biomedical task of a simulated cranial window procedure. Preliminary results show promising performance comparable to those of human experts with only image information, further validating the efficacy of the proposed method.**

I. Introduction

Robotic surgery significantly reduces the workload of surgeons and provides more precise surgical control, gaining substantial attention in recent years [1]. Traditional methods rely on deterministic mathematical models to describe robot behavior, making them less suited for environments with high noise and limited observations [2]. Therefore, effectively modeling and robust surgical control in uncertain environments has become a crucial and challenging research issue..

Over the past decade, deep learning has seen tremendous growth and is now used in robotics as an improvement to traditional controls [3]. However, unlike areas like computer vision or language processing where data is abundant, gathering data in robotics is time-consuming and expensive. Training models is more difficult with limited data, especially in robotic surgery where high accuracy and stability are demanded [4]. Thus, applying deep learning to robotic surgery is still emerging, and effectively combining prior knowledge for better optimization remains unresolved.

Craniotomy is a pivotal step in neurology, neurobiology, and brain-computer interface research, especially within mouse models [5]. Given the small size and intricate anatomical structure of mice, their study can address numerous prevalent challenges in the realm of robotic surgery. However, the inherent uncertainties in the surgical environment, combined with the thinness (0.2-0.3 millimeters) and complex structure of the mouse skull, make it difficult for traditional methods to achieve precise and stable control. Moreover, the limited precision of robotic arms, vibrations, and the restricted observational capabilities of cameras further complicate the procedure. Out of ethical considerations for animals, to minimize the need for real mouse experimental data, researchers widely use chicken eggshells, which bear a resemblance to the anatomical structure and physical properties of mice, as models for simulating automated craniotomies [6]. However, to date, the existing attempts are either tailored for specific experimental settings or lack broad applicability. When compared to human experts, these automated methods still fall short in terms of stability and accuracy.

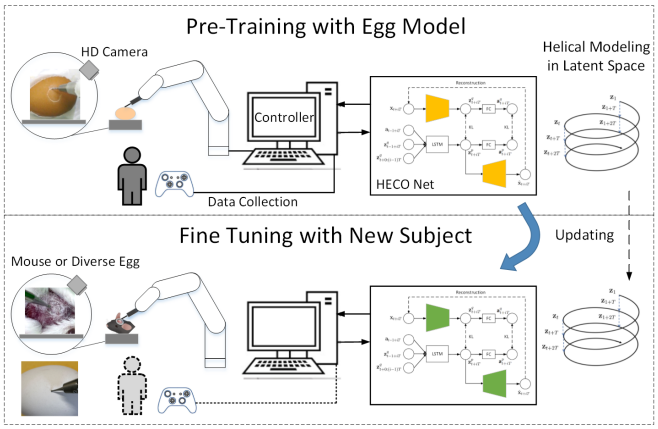

Fig. 1. The two-stage system architecture for helical control in latent space. The modeling of latent space spiral control is initially pre-trained on simulated eggshell data, followed by fine-tuning and transfer learning on different experimental objects. The obtained control policy can then be employed for the automation of skull trepanation procedures. More details in Section III.

We introduce a two-stage cyclical control framework based on imitation learning within latent space, as depicted in Fig. 1. The system platform integrates a 6-axis robotic arm equipped with a drill bit, an observational camera, and a controller. During the pre-training phase, human experts maneuver the robotic arm to simulate operations on chicken eggshells, collecting relevant data. This data is then fed into the proposed Helical Control (HECO) model where neural networks effectively capture latent patterns embedded in the data. In the fine-tuning phase, the trained model is transferred to novel experimental subjects, be it diverse eggshells or actual mice, and parameter adjustments are made through few-shot learning. Experimental outcomes underscore the main contributions of our study: 1) The latent space helical feature modeling significantly bolsters control precision and stability in uncertain environments; 2) The introduced multi-regularization loss function effectively mitigates overfitting. Even with limited training data, the proposed method can achieve efficient transfer learning performance.

The rest paper is organized as follows: Section II provides a review of related work; Section III details our proposed

[1,3]Yuanyuan Jia and Tadahiro Taniguchi are with Ritsumeikan University, 525-8577, Japan, jiayuanusa@gmail.com, [2] Jessica Ziyu Qu is with Canadian Academy, 658-0032, Japan

method; Section IV presents experimental results and comparisons; finally, Section V gives a conclusion.

II. Related Work

Deep Learning in Robotic Surgical Control: As deep learning technology continues to advance, its applications in the field of robotic surgery have expanded, providing a new perspective on traditional deterministic control methods [3]. Among them, the Variational Autoencoder (VAE) stands out in robot perception and decision-making, effectively learning and extracting key features or patterns from surgical data [4]. World models [7] help robots understand their environment to optimize control strategies [8] [9]. In particular, NewtonianVAE [10] [11] enhances control precision. Predictive coding [12] has made significant progress in robot manipulation. However, these studies have not specifically discussed control problems in strong noisy environments and have not specifically modeled the periodic characteristics of robots.

Uncertainty and Periodic Control: The environment of robotic surgery is filled with uncertainties. For example, in mouse skull drilling surgeries, there are many uncertainties caused by vibrations, friction, and slippage. Some researchers try to address these problems through various improved control methods [2]. Periodic control is one of the important ones, effectively improving precision and stability in uncertain environments. For example, Lee et al. [13] designed a VAE and LSTM-based control model for bipedal robot movement, and Zhao and colleagues [14] adopted a similar strategy for robotic arm motion control. These early studies validated the effectiveness of periodic control, but still face challenges in capturing repetitive pattern features, accurately modeling system dynamics, and selecting appropriate control strategies.

Progress in Mouse Skull Drilling Surgery: Mouse skull drilling is crucial in neuroscience, drug delivery, and brain tumor treatment [5]. Some studies explore automated or semi-automated methods [15] [16] to improve accuracy and minimize harm. The procedure involves creating a small skull opening for brain access. Due to the small size of the mouse and complex anatomy, these surgeries still pose considerable challenges. Researchers have proposed techniques such as robotic systems [5], laser ablation [17], and customized drills [18]. To enhance precision, micro-computed tomography [19] is also used to visualize skull structures. Although these methods have improved success rates, they often have high requirements for both equipment and operators, lack generalizability, and still face stability issues.

III. Methodology

The proposed Bayesian spiral control method aims to effectively capture the underlying patterns in the data and achieve high accuracy and stability in automated robotic trepanation, even in the presence of uncertainty and noise.

A. Probabilistic Graphic Model

In skull trepanation surgery, circular interpolation of the grinding head is a highly effective machining method that provides more uniform grinding, better contact between the grinding head and skull surface, and reduces heat damage and cutting forces. Moreover, it can produce a smoother surface and improve machining accuracy.

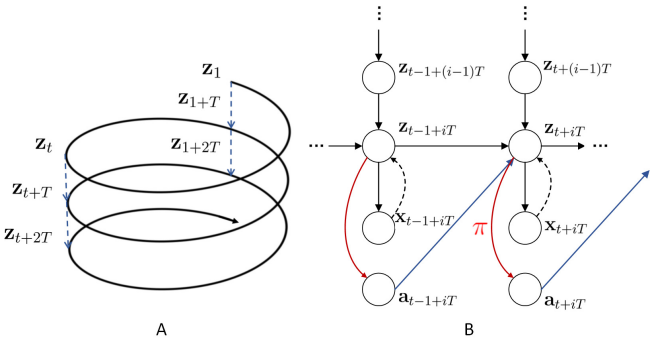

Fig. 2. Probabilistic graphic models, where A shows the spiral trajectory of the end effector of a robotic arm operating in three-dimensional space, and B the unfolded probabilistic graphical model.

Fig. 2A shows the trajectory of the grinding head making periodic circular motions on the workpiece surface. Here, $\mathbf{z}$ denotes the system state in latent space, T denotes the ordinal number of a period, and t is the time stamp within each period. For the sake of simplicity, the model only displays two phases' interrelation among different periods. To represent the relationships between variables more clearly, we unfold Fig. 2A along the time axis and use a probabilistic graphical model (PGM) to depict it, as shown in Fig. 2B, which is also a partially observable markov decision process (POMDP). i is the index of each period. The agent makes control decisions $\mathbf{a}$ based on a probability distribution over the possible states of the system $\mathbf{z}$ given partial observations $\mathbf{x}$. The observations are images captured by a camera. The latent state $\mathbf{z}$ encodes the whole environmental information including the robot itself and its interactions with the skull surface. The arrow from $\mathbf{z}_t$ to $\mathbf{x}_t$ represents the generative relationship where the dashed arrow indicates the inference. The rightward line from $\mathbf{z}_{t-1+iT}$ to $\mathbf{z}_{t+iT}$ shows the state transition, which models the system evolution over time and captures temporal dependencies. The downward line from $\mathbf{z}_{t+(i-1)T}$ to $\mathbf{z}_{t+iT}$ models the dependency between the same phase across contiguous periods. The control input $\mathbf{a} = (u, v, w)$ is defined as the 3D coordinate of grinding head. The red line from $\mathbf{z}$ to action $\mathbf{a}$ indicates the control policy π. During controlling process, the action $\mathbf{a}$ is dynamically estimated and then applied to the robot, affecting the latent state of the next time step as shown by the blue links. In particular, we exploit four different neural networks: encoder, decoder, dynamics and controller, to recursively estimate the probability densities in the above PGM as discussed next.

B. Objective Function

The goal of control is to estimate the latent variables $\mathbf{z}$ and action $\mathbf{a}$ based on the observation $\mathbf{x}$. To achieve this, the joint posterior is first derived based on historic observations,

$$
\begin{aligned}
& p(\mathbf{a}_{1:(t+iT)}, \mathbf{z}_{1:(t+iT)} | \mathbf{x}_{1:(t+iT)}) \\
= \; & \frac{p(\mathbf{a}_{t+iT} | \mathbf{a}_{1:(t-1+iT)}, \mathbf{z}_{1:(t+iT)}, \mathbf{x}_{1:(t+iT)})}{p(\mathbf{x}_{1:(t+iT)})} \\
& \cdot p(\mathbf{a}_{1:(t-1+iT)}, \mathbf{z}_{1:(t+iT)}, \mathbf{x}_{1:(t+iT)}) \qquad (1) \\
= \; & \frac{p(\mathbf{a}_{t+iT} | \mathbf{z}_{t+iT}) p(\mathbf{a}_{1:(t-1+iT)}, \mathbf{z}_{1:(t+iT)}, \mathbf{x}_{1:(t-1+iT)})}{p(\mathbf{x}_{1:(t+iT)})} \\
& \cdot p(\mathbf{x}_{t+iT} | \mathbf{x}_{1:(t-1+iT)}, \mathbf{a}_{1:(t-1+iT)}, \mathbf{z}_{1:(t+iT)}) \qquad (2) \\
= \; & \frac{p(\mathbf{a}_{t+iT} | \mathbf{z}_{t+iT}) p(\mathbf{x}_{t+iT} | \mathbf{z}_{t+iT})}{p(\mathbf{x}_{t+iT} | \mathbf{x}_{1:(t-1+iT)})} \\
& \cdot p(\mathbf{z}_{t+iT} | \mathbf{z}_{t-1+iT}, \mathbf{z}_{t+0:(i-1)T}, \mathbf{a}_{t-1+iT}) \\
& \cdot p(\mathbf{a}_{1:(t-1+iT)}, \mathbf{z}_{1:(t-1+iT)} | \mathbf{x}_{1:(t-1+iT)}) \qquad (3)
\end{aligned}
$$

where the Bayes rule is applied in (1). The Markov property $p(\mathbf{a}_{t+iT} | \mathbf{a}_{1:(t-1+iT)}, \mathbf{z}_{1:(t+iT)}, \mathbf{x}_{1:(t+iT)}) = p(\mathbf{a}_{t+iT} | \mathbf{z}_{t+iT})$ is applied in (2). In (3), there are five densities where $p(\mathbf{a}_{t+iT} | \mathbf{z}_{t+iT})$ is the controller policy, $p(\mathbf{x}_{t+iT} | \mathbf{z}_{t+iT})$ the observation likelihood, $p(\mathbf{z}_{t+iT} | \mathbf{z}_{t-1+iT}, \mathbf{z}_{t+0:(i-1)T}, \mathbf{a}_{t-1+iT})$ the state dynamics, $p(\mathbf{x}_{t+iT} | \mathbf{x}_{1:(t-1+iT)})$ the conditional evidence, $p(\mathbf{a}_{1:(t-1+iT)}, \mathbf{z}_{1:(t-1+iT)} | \mathbf{x}_{1:(t-1+iT)})$ the posterior in the previous time step.

As computing the exact posterior probability in (1) is impractical, we resort to variational inference to estimate it by introducing an approximate posterior $q_{t+iT} = q(\mathbf{a}_{1:(t+iT)}, \mathbf{z}_{1:(t+iT)} | \mathbf{x}_{1:(t+iT)})$. By defining the targeting posterior $p_{t+iT} = p(\mathbf{a}_{1:(t+iT)}, \mathbf{z}_{1:(t+iT)} | \mathbf{x}_{1:(t+iT)})$, we have the KL divergence between these two densities as follows,

$$
\begin{aligned}
& D_{KL}[q_{t+iT} \| p_{t+iT}] \\
= \; & E_q[\log q(\mathbf{a}_{1:(t+iT)}, \mathbf{z}_{1:(t+iT)} | \mathbf{x}_{1:(t+iT)}) \\
& - \log p(\mathbf{a}_{t+iT} | \mathbf{z}_{t+iT}) - \log p(\mathbf{x}_{t+iT} | \mathbf{z}_{t+iT}) \\
& - \log p(\mathbf{z}_{t+iT} | \mathbf{z}_{t-1+iT}, \mathbf{z}_{t+0:(i-1)T}, \mathbf{a}_{t-1+iT}) \\
& - \log p(\mathbf{a}_{1:(t-1+iT)}, \mathbf{z}_{1:(t-1+iT)} | \mathbf{x}_{1:(t-1+iT)}) \\
& + \log p(\mathbf{x}_{t+iT} | \mathbf{x}_{1:(t-1+iT)})] \qquad (4)
\end{aligned}
$$

where we substitute the targeting posterior by (3). Furthermore, following the locality assumption in variational inference [20], which assumes that the estimated posterior $q(\mathbf{a}_{t+iT}, \mathbf{z}_{t+iT} | \mathbf{x}_{t+iT})$ depends only on the current observation $\mathbf{x}_{t+iT}$ and is independent of the past observation, we can have,

$$
\begin{aligned}
& \log q(\mathbf{a}_{1:(t+iT)}, \mathbf{z}_{1:(t+iT)} | \mathbf{x}_{1:(t+iT)}) \\
= \; & \log q(\mathbf{a}_{t+iT}, \mathbf{z}_{t+iT} | \mathbf{x}_{t+iT}) \\
& + \log q(\mathbf{a}_{1:(t-1+iT)}, \mathbf{z}_{1:(t-1+iT)} | \mathbf{x}_{1:(t-1+iT)}) \qquad (5) \\
= \; & \log q(\mathbf{a}_{t+iT} | \mathbf{z}_{t+iT}) + \log q(\mathbf{z}_{t+iT} | \mathbf{x}_{t+iT}) \\
& + \log q(\mathbf{a}_{1:(t-1+iT)}, \mathbf{z}_{1:(t-1+iT)} | \mathbf{x}_{1:(t-1+iT)}) \qquad (6)
\end{aligned}
$$

where we utilize Markov property that action $\mathbf{a}_{t+iT}$ is conditionally independent of $\mathbf{x}_{t+iT}$ given the latent variable $\mathbf{z}_{t+iT}$, as stated in (6). By substituting (6) back into (4), we obtain,

$$
\begin{aligned}
& D_{KL}[q_{t+iT} \| p_{t+iT}] \\
= \; & D_{KL}[q(\mathbf{z}_{t+iT} | \mathbf{x}_{t+iT}) \| p(\cdot)] \\
& + D_{KL}[q(\mathbf{a}_{t+iT} | \mathbf{z}_{t+iT}) \| p(\mathbf{a}_{t+iT} | \mathbf{z}_{t+iT})] \\
& - E_q[\log p(\mathbf{x}_{t+iT} | \mathbf{z}_{t+iT})] + D_{KL}[q_{t-1+iT} \| p_{t-1+iT}] \\
& + E_q[\log p(\mathbf{x}_{t+iT} | \mathbf{x}_{1:(t-1+iT)})] \qquad (7)
\end{aligned}
$$

where $p(\cdot) = p(\mathbf{z}_{t+iT} | \mathbf{z}_{t-1+iT}, \mathbf{z}_{t+0:(i-1)T}, \mathbf{a}_{t-1+iT})$. The quantity $E_q[\log p(\mathbf{x}_{t+iT} | \mathbf{x}_{1:(t-1+iT)})]$ is independent of the model parameters and latent state $\mathbf{z}$. Similarly, the loss from the previous time step $D_{KL}[q_{t-1+iT} || p_{t-1+iT}]$ is known at time $t + iT$. Thus, both of these terms can be treated as constants in the optimization of the loss function J_{t+iT} at time $t + iT$, and we have,

$$
\begin{aligned}
J_{t+iT} = \; & D_{KL}[q(\mathbf{z}_{t+iT} | \mathbf{x}_{t+iT}) \| p(\cdot)] \\
& + D_{KL}[q(\mathbf{a}_{t+iT} | \mathbf{z}_{t+iT}) \| p(\mathbf{a}_{t+iT} | \mathbf{z}_{t+iT})] \\
& - E_q[\log p(\mathbf{x}_{t+iT} | \mathbf{z}_{t+iT})] \qquad (8)
\end{aligned}
$$

Each of the two KL divergences and the expectation (reconstruction loss) can be employed as separate losses for different neural networks, and they can work collaboratively to improve the robustness of control performance.

C. Neural Network Architecture

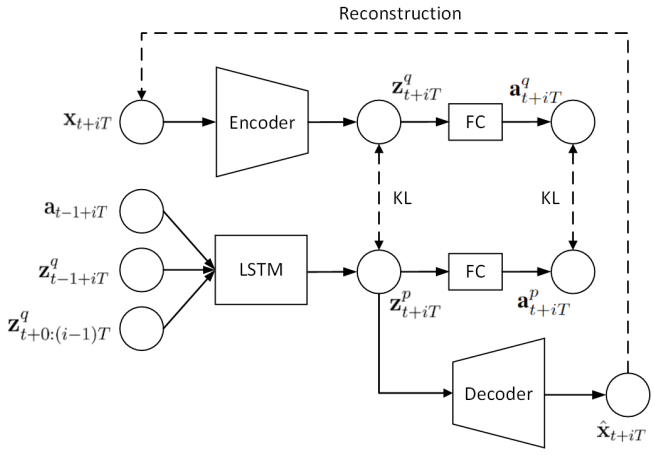

Fig. 3. Neural network structure of the proposed HECO model for control.

As shown in Fig. 3, we exploit four neural networks for the whole system structure. The encoder network is designed to take in an image observation $\mathbf{x}_{t+iT}$. It is composed of a convolutional neural network (CNN) with two hidden layers. Each layer has 32 filters of size 3×3 and a stride of 1, along with a Rectified Linear Unit (ReLU) activation function. The output of this network is a compressed, lower-dimensional latent state $\mathbf{z}^q_{t+iT}$, which represents a compact representation of the input image. We adopt $E_q[\log q(\mathbf{z}_{t+iT} | \mathbf{x}_{t+iT})]$ in (8) as the loss function.

The decoder network of the VAE is a deconvolutional neural network that takes the latent state $\mathbf{z}^p_{t+iT}$ as input, and

generates the reconstructed observation $\hat{\mathbf{x}}_{t+iT}$ using a fully connected layer, two transposed convolutional layers, and a ReLU activation function. Each transposed convolutional layer has 32 filters of size 3×3 and a stride of 1. The decoder network is specifically designed to reconstruct the original input image, which has a resolution of $64 \times 64 \times 3$. The expectation of log likelihood $E_q[\log p(\mathbf{x}_{t+iT}|\mathbf{z}_{t+iT})]$ from in (8) is adopted as the loss function. It compares the difference between the observation $\mathbf{x}_t$ and the reconstructed image $\hat{\mathbf{x}}_t$.

An LSTM network updates latent state $\mathbf{z}^p_{t+iT}$ based on loss $E_q[\log p(\cdot)] = E_q[\log p(\mathbf{z}_{t+iT}|\mathbf{z}_{t-1+iT}, \mathbf{z}_{t+0:(i-1)T}, \mathbf{a}_{t-1+iT})]$ from (8). Particularly, we use $\mathbf{z}^p_{t+iT}$ to distinguish from the predicted latent state $\mathbf{z}^q_t$ by encoder since it represents the targeting "true" state in the latent space.

For the controller, we employ two additional fully connected (FC) networks with inputs $\mathbf{z}^q_{t+iT}$ and $\mathbf{z}^p_{t+iT}$ to predict the action signal. The loss function is derived from equation (8), specifically $D_{KL}[q(\mathbf{a}_{t+iT}|\mathbf{z}_{t+iT})\|p(\mathbf{a}_{t+iT}|\mathbf{z}_{t+iT})]$. The outputs of these networks can both serve as control signals in theory, but in our experiments, we observed that $\mathbf{a}^p_t$ exhibited relatively better performance.

D. Algorithm Pseudo Code

Algorithm 1 displays a pseudocode of the proposed algorithm that outlines the training process for a single period.

Algorithm 1 Helical Control in Latent Space

Input: Training dataset $\mathbf{x}_{1:T}$, $\mathbf{a}_{1:T}$, learning rate η
Set $t = 1$
Initialize the encoder, decoder, LSTM, and FCs
while $t < T$ **do**
  Set $j = t + iT$
  Encoder output: $\mathbf{z}^q_j = \text{Encoder}_{\theta_e}(\mathbf{x}_j)$
  LSTM output: $\mathbf{z}^p_j = \text{LSTM}_{\theta l}(\cdot)$
  Controller: $\mathbf{a}^q_j = \text{FC}_{\theta_{f1}}(\mathbf{z}^q_j)$, and $\mathbf{a}^p_j = \text{FC}_{\theta_{f2}}(\mathbf{z}^p_j)$
  Decoder output: $\hat{\mathbf{x}}_j = \text{Decoder}_{\theta_d}(\mathbf{z}^p_j)$
  Compute KL divergence for state L_z and action L_a
  Compute the reconstruction loss: L_x
  Compute the total loss: $J_j = L_z + L_a + L_x$
  Update network parameters by Adam optimizer:
  $\theta_e, \theta_d, \theta_l, \theta_{f1}, \theta_{f2}$
  $t \leftarrow t + 1$
end while
Output: Trained model weights

IV. EXPERIMENTAL RESULTS

In this section, we present the experimental results of the proposed HECO method in the robotic mouse trepanation project[4], and compare it with two different approaches: the traditional Closed-Loop Controller (CLC) [15] and the VAE-based controller without spiral control strategy (VAE) [4].

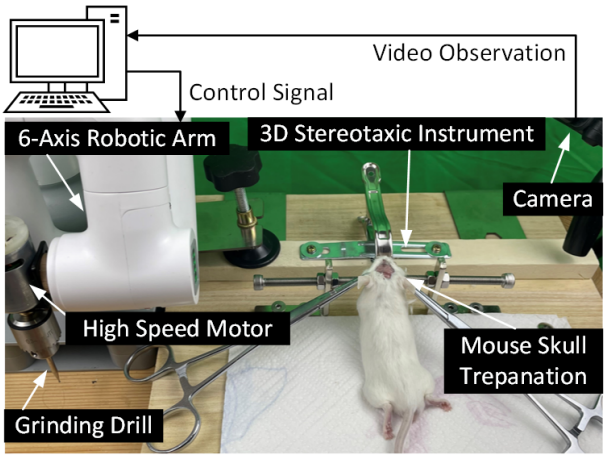

Fig. 4. System platform for automatic trepanation using a 6-axis robotic arm and a monocular camera.

TABLE I
SPECIFICATIONS OF THE ROBOT AND DRILL

Items	Values
Total Arm Length (mm)	300
Weight (kg)	3
Maximum Composite Speed	Default 400 (100 to 1500)
Position Accuracy (mm)	$\pm$0.08mm′2
Rated Load Capacity (kg)	0.3
Diameter of Drill (mm)	1

A. Experiment Setup

Fig. 4 provides an overview of the experimental setup. The ultimate goal is to perform automated craniotomy surgery on mice. Nevertheless, due to stringent considerations of animal ethics and an aim to reduce animal experimentation, the current validation of the proposed methodology has, as is customary [5], been limited to experiments on various types of eggshells. In the future work, we will transit to real mice. A 6-axis robotic arm is used for the craniotomy, and a diamond burr was mounted on the end effector. Compared to traditional surgical scalpels, it can produce more precise cuts and reduce vibrations in the bone tissue. Table I provides specifications of the robot and the burr. A 3D stereotactic device was chosen to stabilize the mouse's head. The cutting angle was set between 30-45 degrees, which facilitates better penetration into the material surface, reduces cutting force and thermal damage, and enhances machining efficiency and surface quality. Craniotomies typically employ a circular incision, as it inflicts minimal damage to the brain tissue compared to other shapes. Moreover, controlling the movement of the burr is relatively straightforward, ensuring uniformity and stability in machining depth. A monocular camera on the right side monitors and analyzes the machining process in real-time. The camera's line of sight is perpendicular to the direction of the grinding rod, facilitating better observation of the surface being machined and avoiding

[4]This work was supported by JST [Moonshot R&D][Grant Number JPMJMS2033].

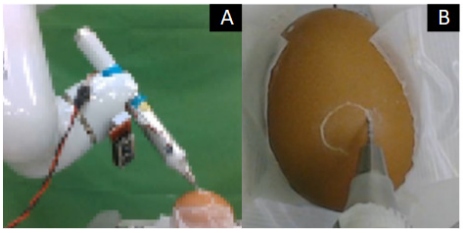

Fig. 5. The eggshell model used for data collection in the pre-training phase. A) Human expert manually tele-operates the robotic arm to conduct craniotomy on an egg. B) The physical properties of the eggshell bear similarities to animal skulls.

errors caused by perspective.

As shown in Fig. 1, we adopted a two-stage strategy of pre-training followed by fine-tuning, learning control policies from expert drilling demonstration data. In the pre-training phase, we selected eggshells as the simulation object for data collection. Subsequently, in the fine-tuning phase, we applied the previously learned policy to new experimental subjects (eggshells of different materials, colors, and types, or future real mouse experiments). The egg model is recognized by researchers as an ideal substitute for training stereotactic surgical protocols [21]. As depicted in Fig. 5, the thickness of the eggshell ($393 \pm 6\mu$m) is extremely close to the skull thickness of mice ($347 \pm 25\mu$m), rats ($1023 \pm 59\mu$m), and macaques ($976 \pm 78\mu$m) [21]. The eggshell consists of multiple different layers [6], with each layer having distinct physical properties such as strength and friction coefficient, offering the possibility to simulate the diversity of animal skulls. Through this two-phase strategy, we effectively addressed the issue of insufficient training samples, achieving a good balance between cost and efficiency.

B. Pretraining with Simulated Data

As depicted in Fig. 5, human experts manually operated the robot to create a circular window on the eggshell, not penetrating the membrane below, thereby collecting the training data. Each successful operation generates a sequence trajectory of observed images and its corresponding control signal sequence. We obtained a total of 110 such trajectories. We also compared the performance of the HECO method (whose loss function (8) includes a "reconstruction term" and two KL "regularization terms") with the standard VAE method [4] that does not model periodic features.

Fig. 6 illustrates the convergence trajectory of the loss functions for both methods. The computations were performed on a PC equipped with an 8-core i7 processor, 32GB of RAM, and an RTX 2080Ti GPU. Using a learning rate set at 0.0002, the model converged within a span of 2000 epochs. From the figure, we can clearly see the following differences: 1) HECO, due to its design that better captures the periodic features in the data, converges faster than the standard VAE. This is mainly attributed to the fact that HECO's structure and loss function are more aligned with the true distribution of the data. 2) Benefiting from HECO's optimized modeling of data's periodic features, its final loss

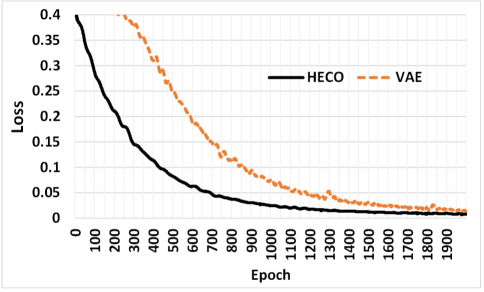

Fig. 6. Convergence process of the loss functions of HECO and VAE with a learning rate set at 0.0002.

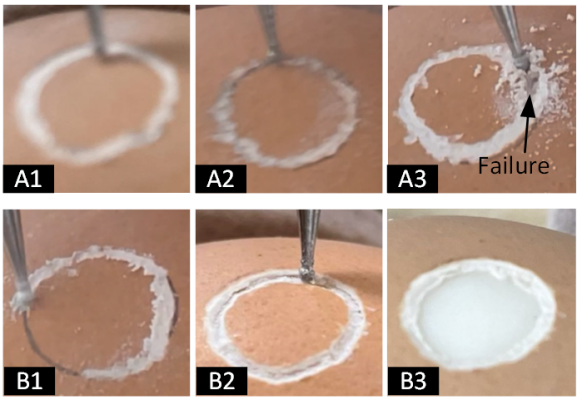

Fig. 7. Comparison of test performance: A1-A3 present randomly selected outcomes from VAE, while B1-B3 illustrate results from HECO.

value is also lower than the standard VAE. 3) HECO exhibits a more stable convergence behavior, thanks to its effective capture of periodic features in the data. In contrast, the standard VAE faces more challenges when trying to capture these periodic features, leading to oscillations in the loss function curve.

TABLE II
TESTING COMPARISON OF LEARNED POLICIES ON SIMULATION DATA

Method	Success Rate	Average Time (s)	Jerk Score
VAE	16.7%	133	19
CLC	33%	91	9
HECO	91.7%	107	4
Manual	100%	148	7

C. Fine-tuning and Testing Analysis

When changing the experimental subjects, such as eggshells of different colors, materials, stages of ripeness, or testing on real mice, it is necessary to adaptively fine-tune this pretrained foundational model. By sharing model parameters across different tasks, transfer learning can significantly speed up the learning process for drilling, reduce training time, and enhance model performance when few samples are available during the fine-tuning phase. In our current experiment, just 10 new trajectories achieved a satisfactory convergence.

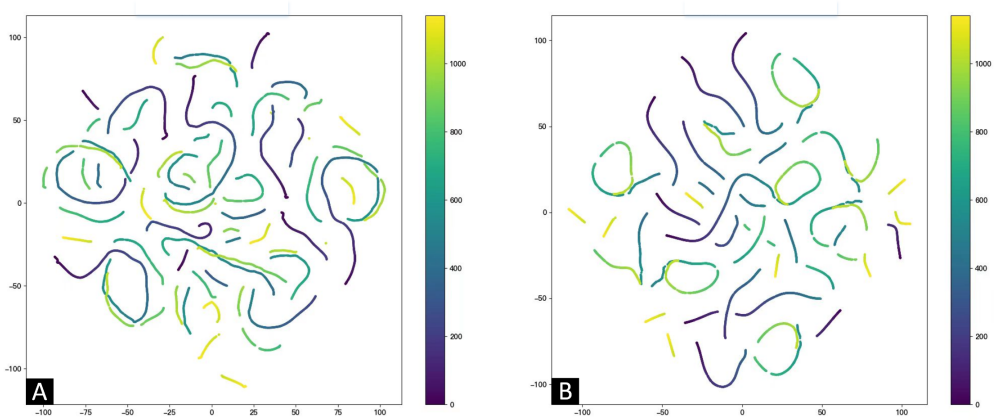

Fig. 8. Latent space visualizations using T-SNE: A depicts results from VAE, while B represents those from HECO.

We conducted a detailed comparison and control strategies using the HECO method versus VAE, CLC, and manual control by experts. Evaluation criteria mainly included the success rate of the drilling operation, average completion time, and the smoothness of the drilling trajectory. We assessed the average abruptness of drilling trajectories by calculating the rate of acceleration changes during drilling. A lower jerk score indicates a smoother procedure. Table II lists the average performance of 12 eggshell drilling experiments. The results show that the HECO method, by effectively simulating spiral features, significantly reduced overfitting. Compared to manual operations by experts, HECO achieved a success rate of 91.7% and displayed substantial improvements in completion time and trajectory smoothness. Experiments also pointed to factors that require further attention and improvement, such as maintaining consistent camera positions and the angle of the drill bit since discrepancies in training image perspectives may slow down model convergence. Larger variations during testing could impact the stability of the control strategy, possibly leading to its failure. In contrast to the HECO method, the success rate of the standard VAE method was not high. Furthermore, while the CLC method performed better in average completion time due to its one-time completion feature [15], it had a higher failure rate. This is primarily because its interpolation strategy relies entirely on the interpolation curve obtained after pre-drilling, rather than utilizing image information for global observation and trajectory analysis. Its stability greatly decreases in high vibration and noisy environments. Additionally, the CLC requires high-precision triaxial robots and measurements of induction signals, complicating the installation process, especially since the high-speed friction generated at the contact points of the roller and the drill bit might increase test errors and limit its applications. When we applied this method to a 6-axis robot arm with more degrees of freedom but lower precision, its accuracy was greatly affected.

To further analyze the HECO method's advantages, we made a detailed comparison of its performance against VAE. This showcases why modeling with spiral features can enhance control stability when both are based on latent variable analysis. Fig. 7 presents representative samples using both VAE and HECO. A1 and A2 represent images reconstructed by VAE, B1 and B2 by HECO, while A3 and B3 respectively display the final result images where VAE failed, but HECO succeeded. Although VAE can reconstruct images, it falls short in capturing details, evident in the blurry edges of the drill rod and eggshell and significant errors in trajectory reconstruction. In contrast, HECO model is more stable, significantly improving the quality of the generated images, reflecting its excellence in capturing input data features through latent variables. Precise latent variable estimation is crucial for accurate and stable motion control, explaining why the HECO method excels in simulated cranial window task.

We further employed the t-SNE [22] technique to visualize the estimated latent vectors in 2D space and compared it with the VAE method. As shown in Fig. 8, different colors represent different sequence orders. It's evident that, compared to VAE, HECO presents a clearer and smoother representation of spiral features in the latent space. At different moments, there's less overlap of features, and their distribution is more uniform and dispersed. This characteristic undoubtedly provides an advantage during the decoding process and aids in achieving more accurate and stable control.

V. Conclusion

A novel precise control algorithm based on probabilistic graphical models has been introduced. This approach models the spiral periodic features of the latent space, significantly enhancing the effective prediction of control actions. Utilizing a pre-training and fine-tuning framework, effective transfer learning of control strategies in small sample scenarios was achieved. Bayesian inference further enhanced the stability of the model's control in noisy environments, such as strong vibrations. The method was applied to simulated cranial window tasks, and its accuracy, efficiency, and stability were verified via comparisons with expert operations and several other methods.

REFERENCES

[1] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot modeling and control*. John Wiley & Sons, 2005.

[2] E. Tagliabue, D. Meli, D. Dall'Alba, and P. Fiorini, "Deliberation in autonomous robotic surgery: a framework for handling anatomical uncertainty," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 11 080–11 086.

[3] K. Abidi, "Robot arm control using a probabilistic neural network," *Journal of Intelligent & Robotic Systems*, vol. 87, no. 1, pp. 115–124, 2017.

[4] S. Hussain, A. Brunetti, G. Lucarelli, R. Memeo, V. Bevilacqua, and D. Buongiorno, "Deep learning based image processing for robot assisted surgery: A systematic literature survey," *IEEE Access*, 11 2022.

[5] H. Li, Q. Li, Y. Liu, and W. Qiu, "A review of robotic systems for skull base surgery: from active constraint to navigation assistance," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 218–236, 2020.

[6] D. Lammie, J. Wilson, V. Saranathan, and G. R. Graves, "Microfocus CT of the avian egg: a virtual approach to the study of bird development," *Journal of anatomy*, vol. 206, no. 5, pp. 447–454, 2005.

[7] T. Taniguchi, S. Murata, M. Suzuki, D. Ognibene, P. Lanillos, E. Ugur, L. Jamone, T. Nakamura, A. Ciria, B. Lara *et al.*, "World models and predictive coding for cognitive and developmental robotics: frontiers and challenges," *Advanced Robotics*, pp. 1–27, 2023.

[8] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg, "Daydreamer: World models for physical robot learning," in *Conference on Robot Learning*. PMLR, 2023, pp. 2226–2240.

[9] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *International conference on machine learning*. PMLR, 2019, pp. 2555–2565.

[10] R. Okumura, N. Nishio, and T. Taniguchi, "Tactile-sensitive NewtonianVAE for high-accuracy industrial connector insertion," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 4625–4631.

[11] M. Jaques, M. Burke, and T. M. Hospedales, "NewtonianVAE: Proportional control and goal identification from pixels via physical latent spaces," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4454–4463.

[12] H. Ito, K. Yamamoto, H. Mori, and T. Ogata, "Efficient multitask learning with an embodied predictive model for door opening and entry with whole-body control," *Science Robotics*, vol. 7, no. 65, p. eaax8177, 2022.

[13] J. Lee, H. Kim, and H. W. Park, "Variational autoencoder-based bipedal motion control," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 697–704, 2019.

[14] J. Zhao, Y. Su, and X. Wu, "Robotic arm motion control using variational autoencoder and LSTM," *Journal of Intelligent & Robotic Systems*, vol. 98, no. 1, pp. 159–169, 2020.

[15] N. Pak, J. H. Siegle, J. P. Kinney, D. J. Denman, T. J. Blanche, and E. S. Boyden, "Closed-loop, ultraprecise, automated craniotomies," *Journal of neurophysiology*, vol. 113, no. 10, pp. 3943–3953, 2015.

[16] G.-B. Bian, B.-T. Wei, Z. Li, P. Ge, Z. Chen, C. Qian, J. Wang, P. Fu, and J. Zhao, "Robotic automatic drilling for craniotomy: Algorithms and in vitro animal experiments," *IEEE/ASME Transactions on Mechatronics*, 2023.

[17] C.-F. J. Yang, J.-M. Lee, and J.-S. Chen, "Robotic surgery for thoracic disease: review of the literature and future perspectives," *Journal of robotic surgery*, vol. 12, no. 1, pp. 1–8, 2018.

[18] G.-S. Choi and H. W. Kim, "Robotics in general surgery: Overview and current status of robotic gastrointestinal surgery," *Journal of Minimal Access Surgery*, vol. 14, no. 4, p. 237, 2018.

[19] M. J. Kim, Y. T. Kim, and S. H. Kim, "Robot-assisted minimally invasive surgery for gynecologic cancer: current status and controversies," *Journal of Gynecologic Oncology*, vol. 30, no. 6, p. 93, 2019.

[20] D. Barber, *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[21] L. Andreoli, H. Simplício, and E. Morya, "Egg model training protocol for stereotaxic neurosurgery and microelectrode implantation," *World Neurosurgery*, vol. 111, pp. 243–250, 2018.

[22] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. 11, 2008.