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

A User-Centered Shared Control Scheme with Learning from Demonstration for Robotic Surgery

Haoyi Zheng[1], Zhaoyang Jacopo Hu[2], Yanpei Huang[1,3], Xiaoxiao Cheng[1,4], Ziwei Wang[5] and Etienne Burdet[1]

***Abstract*— The utilization of shared control in the realm of surgical robotics augments precision and safety by amalgamating human expertise with autonomous assistance. This paper proposes a user-centered shared control framework enabling a robot to learn from expert demonstration, predict operators' intent and modulate control authority to provide natural assistance when needed. We employ deep inverse reinforcement learning (IRL) to enable the robot to learn path planning from expert demonstrations with fast convergence, subsequently enhancing the policy with a potential field method. The control authority is allocated seamlessly between the human operator and the autonomous agent based on the prediction of operators' movement from an adaptive filter and fuzzy logic inference. The proposed method is executed using the da Vinci Research Kit (dVRK) robot in a simulation environment, and its effectiveness is assessed through user performance evaluation in a trajectory tracking task. Compared to direct control and simple shared control, the proposed shared control scheme exhibits superior tracking accuracy and trajectory smoothness under external disturbances. Subjective responses underscore users' perception of the method's efficacy in enhancing their performance.**

I. Introduction

Robotic surgery has revolutionized the landscape of medical interventions, offering enhanced precision, reduced invasiveness, and improved patient outcomes. In most of the existing surgical robotic systems, the surgeon directly controls the remote robotic instruments in teleoperation mode [1]. In this case, surgical safety could be compromised due to human operator errors [2], fatigue [3], or technical failures in system control or communications [4].

Human-robot shared control may solve these issues by exploiting the advantages of both humans and robots in sensing and control [5]. One essential premise to achieve this shared control scheme is to train the robot agent to perform surgical tasks as human operators would. Learning from demonstration (LfD) is the approach wherein robots learn new abilities by observing and emulating the actions of an expert [6]. Many methods have been proposed for LfD, such as the Gaussian Mixture Model (GMM), Dynamic Movement Primitives (DMPs), and inverse reinforcement learning (IRL). IRL attempts to reason the expert's preferences rather than directly mimicking the expert's behaviour [7]. It has been applied in various scenarios of robotic control, including autonomous driving [8], robot navigation [9], and robotic surgical operations [10], [11]. The recent work from Hu et al. [12] has applied maximum entropy IRL to train a robotic agent to perform a surgical training task of peg transfer. However, the aforementioned IRL-based training is computationally demanding, and unable to traverse the state space with limited training samples, leaving space for more efficient and comprehensive path planning.

State fusion shared control (SFSC) is a control scheme where human and autonomy inputs are coupled with an arbitration mechanism [13]. This arbitration mechanism allocates the authority between the human and autonomy, allowing various forms including context detection [14], sensory measurements [15] or bilateral trust model [16]. However, current works lack an efficient scheme for performance regulation, thus an error-based authority allocation would be beneficial

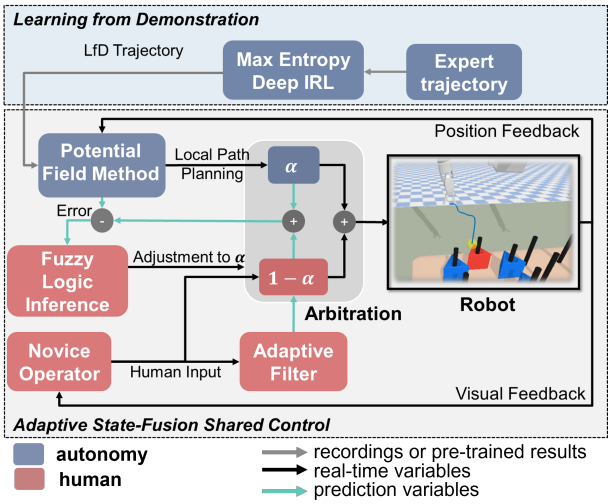

Fig. 1. Overall diagram of the shared control scheme. On the autonomy side, an LfD trajectory is first generated with trained Maximum Entropy Deep IRL, then passed to a potential field for local path planning. On the human side, the operator's and shared control motions are predicted sequentially using an adaptive filter. Errors between the predicted shared control motion and the planned motion are then used for authority adjustment in arbitration.

[1] Haoyi Zheng, Yanpei Huang, Xiaoxiao Cheng, and Etienne Burdet are with the Department of Bioengineering, Imperial College of Science, Technology and Medicine, London, UK. [2] Zhaoyang Jacopo Hu is with the Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, London, UK. [3] Yanpei Huang is also with School of Engineering and Informatics, University of Sussex, Brighton, UK. [4] Xiaoxiao Cheng is also with the Department of Electrical and Electronic Engineering, University of Manchester, Manchester, UK. [5] Ziwei Wang is with the School of Engineering, Lancaster University, Lancaster, UK. {yanpei.huang@sussex.ac.uk, xiaoxiao.cheng@manchester.ac.uk, e.burdet@imperial.ac.uk}

This work was supported by the EC H2020 grants NIMA (FETOPEN 899626), CONBOTS (ICT 871803), EPSRC and Intuitive Surgical in the form of an industrial CASE studentship.

to guarantee safety in robotic-assisted surgery.

By considering these factors, this paper proposes a novel user-centered human-robot shared control framework for robotic surgery, enabling safe control by novice operators. The framework involves a robotic agent learning from expert demonstrations, assisting novice operators in trajectory following tasks, and correcting their actions in case of deviations from the intended path. Instantaneous prediction of shared-control deviations from target paths helps mitigate uncertainties in human control. This framework is characterized by the following key features:

- Enhanced autonomous path planning through combing a faster IRL algorithm and a robust local path reconnection method based on potential field.
- Developed a human motion intent prediction method utilizing adaptive filtering in time series prediction for improved arbitration.
- Allocated control authority in real-time using fuzzy logic inference according to position and direction errors.

To evaluate the proposed framework, an experiment was conducted using the dVRK simulator [12], [17] with five participants. The results show that the proposed human-centered framework could effectively improve task performance especially when the human operator receives an external disturbance.

II. Learning from Demonstration and Path Planning

A. LfD with Maximum Entropy Deep IRL

Maximum Entropy Deep IRL (MaxEntropy Deep IRL) is used as the LfD method. It takes a human expert's trajectories as demonstrations and learns a reward function that best describes the expert's behavior. The recording of one expert trajector ies' (x, y, z) coordinates are discretized in a $22 \times 22 \times 22$ grid space. Each grid is indexed as one state, and the number of states is the feature size. The action space is defined as the next movement, or $(\Delta x, \Delta y, \Delta z)$ coordinate. The IRL agent learns the reward function given demonstrations and a Markov Decision Process (MDP). The reward approximation function [18] is

$$r = g(f, \theta), \tag{1}$$

where r denotes the rewards, θ is the parameters for the neural network, and f represents a binary vector where each state is one-hot encoded. These parameters are updated with the backpropagation process of the neural network. The joint logarithmic likelihood of observing expert demonstrations D and of model parameters could be described as follows:

$$\mathcal{L}(\theta) = \log \mathcal{P}(D, \theta | r) = \log \mathcal{P}(D|r) + \log \mathcal{P}(\theta), \tag{2}$$

where $\log \mathcal{P}(D|r)$ is denoted as $\mathcal{L}_D$, and $\log \mathcal{P}(\theta)$ is denoted as $\mathcal{L}_\theta$. The gradient with respect to θ consists of two terms, gradient with respect to θ of the data $\frac{\partial \mathcal{L}_D}{\partial \theta}$ and the weight decay regulariser $\frac{\partial \mathcal{L}_\theta}{\partial \theta}$:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}_D}{\partial \theta} + \frac{\partial \mathcal{L}_\theta}{\partial \theta}, \tag{3}$$

$$\frac{\partial \mathcal{L}_D}{\partial \theta} = \frac{\partial \mathcal{L}_D}{\partial R} \frac{\partial R}{\partial \theta} = (\mu_D - \mathbb{E}[\mu]) \frac{\partial g(f, \theta)}{\partial \theta}, \tag{4}$$

where μ_D is the visitation frequency observed in the given demonstrations, and $\mathbb{E}[\mu]$ is the observed visitation frequency given a policy (learned with current reward function R). We can then obtain $\frac{\partial \mathcal{L}_D}{\partial \theta}$ with backpropagation.

Our neural network has a simple full connection structure. The input layer contains neurons with the same size as the feature vector, and two hidden layers have 16 and 8 neurons respectively, both of which use ReLU activation functions. The network has one output, with a tanh activation function.

We take the learned rewards R and solve the MDP using the learned rewards from IRL. By using the Q-learning update rule as shown in eq. (5), we record the trajectories observed and calculate the state visitation frequencies.

$$Q(s, a) = (1 - \alpha) Q(s, a) + \alpha (r + \gamma \max_{a'} Q(s', a')), \tag{5}$$

where s is the current state, a is an action, s' is the next state, r is the reward of this step, α is the learning rate and γ is a discount factor ranging from 0 to 1.

The IRL algorithm is implemented with the SurRoL simulator [17] for the laparoscopic surgery. We have collected one demonstration for each trial and trained the model separately for the 4 trials in our experiments (Section IV-B). Both epsilon decay and reward shaping [19] techniques are introduced in the training process to facilitate training. The overall IRL algorithm is shown in Algorithm 1.

Algorithm 1 Maximum Entropy Deep IRL

Input: D, f
Output: R, π^*
  $\mu_D \leftarrow$get_visitation_frequency(D)
  θ, $Q(s, a)$, $\hat{D} \leftarrow$initialize()
  $R \leftarrow$ nn_forward()
  for $n = 1$: episodes **do**
    if $n\%$ update frequency $= 0$ **then**
      $\mathbb{E}[\mu] \leftarrow$ estimate_visitation_frequency$(\hat{D})$
      $\frac{\partial \mathcal{L}_D}{\partial \theta} =$nn_backprop$(\mu_D - \mathbb{E}[\mu])$
      $R \leftarrow$ nn_forward(f)
      $\theta \leftarrow$ update_weights $(\theta, \frac{\partial \mathcal{L}_D}{\partial \theta})$
      $\hat{D} \leftarrow$clear()
    end if
    $s \leftarrow$ reset(), $\zeta \leftarrow s$
    while not done **do**
      $s' \leftarrow$ step $(s, \epsilon$_greedy$(s, Q)), r \leftarrow$ get_reward (R)
      $Q(s, a) \leftarrow (1-\alpha) \cdot Q(s, a) + \alpha \cdot (r + \gamma \cdot \max_{a'} Q(s', a'))$
      $s \leftarrow s', \zeta$.pull (s)
    end while
    $\hat{D}$.pull(ζ)
    $\pi^* \leftarrow \epsilon$_greedy_policy$(Q)$
  end for
  return R, π^*

B. Potential Field Local Path Planning

We take the IRL learned Q-table to generate trajectories from a starting point to reach the target position. As the state

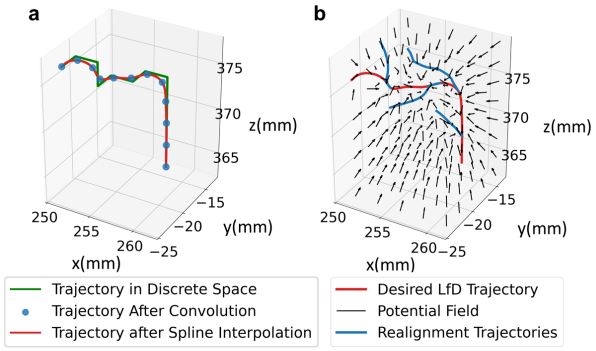

Fig. 2. Illustrations of Potential Field Path Planning (a) Trajectories after Smoothing and Interpolation, (b) Potential Field and Example Realignment Trajectories.

space is discretized in the LfD algorithm, it may lead to jagged trajectories thus reducing the comfort of the human user. Furthermore, due to the large number of states and dimensionalities, the agent is unable to fully explore all state space, leaving insufficiently updated Q-values of unexplored states, which could cause some unexpected behavior from the trained agent.

To solve these issues, post-processing of the planned trajectory is needed by using smoothing and interpolation techniques. The trajectory is firstly smoothed with a 1-D convolution, which convolutes sequences of x, y, z coordinates with a [0.25, 0.5, 0.25] convolution kernel respectively. Then it is interpolated using cubic spline interpolation, as shown in Fig. 2a. The cubic spline interpolation method [20] is then applied to further smooth the trajectory with two times continuously differentiable interpolating functions. As a result, the average minimum curvature radius increases to 1.01, a significant improvement compared to 0.33 for cubic interpolation without convolution smoothing and 0.02 for only linear interpolation.

In case of deviation caused by human-errors or environmental disturbances, we introduce potential fields for local planning, aiding the robot in realigning with the LfD trajectory. Potential field methods are commonly used in shared control systems for obstacle avoidance [21] or lane-keeping [22]. Our force field is defined as follows:

$$\mathbf{F} = \mathbf{F}_{att}(\mathbf{p}) + \mathbf{F}_{acc}(\mathbf{p}) \tag{6}$$

$$\mathbf{F}_{att} = k_{att1} \cdot (\mathbf{p}_{tar} - \mathbf{p}) + k_{att2} \sum_{i=1}^{N} \left(\frac{1}{||\mathbf{p}_i - \mathbf{p}||^2} \cdot \frac{(\mathbf{p}_i - \mathbf{p})}{||\mathbf{p}_i - \mathbf{p}||} \right) \tag{7}$$

$$\mathbf{F}_{acc} = \frac{k_{acc}}{||\mathbf{p}_n - \mathbf{p}||^2} \cdot \frac{(\mathbf{p}_{n+1} - \mathbf{p}_n)}{||\mathbf{p}_{n+1} - \mathbf{p}_n||}, n = \arg\min_i ||\mathbf{p}_i - \mathbf{p}|| \tag{8}$$

where $\mathbf{F}$ is the total combination force, $\mathbf{F}_{att}$ and $\mathbf{F}_{acc}$ are attractive and accelerative forces respectively, k_{att1}, k_{att2} and k_{acc} are handcrafted constant coefficients, $\mathbf{p}_i$ denotes the position of the i-th point on the smoothed LfD trajectory, $\mathbf{p}$ is the current position, $\mathbf{p}_{tar}$ is the target position. The robot moves along the resultant force's direction at a set step length (0.25mm).

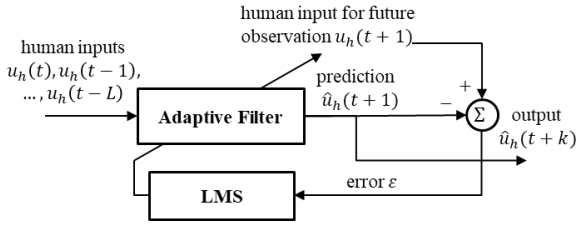

Fig. 3. Block diagram of adaptive filter human input prediction.

The attractive force term comprises two aspects: attracting the robot towards the target and guiding it back on track. The realignment is achieved by applying attractive forces at various points along the trajectory and we introduce an extra attractive force toward the target position to ensure that the robot moves toward the target while rejoining the LfD trajectory. In addition, the accelerative force boosts the robot positively when it is near the desired LfD trajectory to prevent it from inadvertently moving in the opposite direction along the path. This generated potential field is shown in Fig. 2b.

III. Human-Centered Shared Control

We use state-fusion shared control [13] which combines both human intent $\mathbf{u}_h$ and agent's planned path $\mathbf{u}_a$ in an arbitration process. It uses a dominance factor α, ranging from 0 to 1, to determine how much the autonomy (or human) dominates the control:

$$\mathbf{u} = \alpha \cdot \mathbf{u}_a + (1 - \alpha) \cdot \mathbf{u}_h. \tag{9}$$

The human intended motion $\mathbf{u}_h$ is predicted by an adaptive filter (Section III-A) while the dominance factor α is adjusted through fuzzy logic inference (Section III-B).

A. Adaptive Filter Human Input Prediction

Human inputs to the interface are a series of time-varying coordinates noted as $[x(t), y(t), z(t)]$. For convenience, we will use u_h to represent one of the three coordinates of human intent, which could be predicted using an adaptive filter:

$$\hat{u}_h(t+k) = \sum_{i=1}^{L} w_i(t) u_h(t+k-i),\ k = 1, 2, 3, \ldots \tag{10}$$

All weights w_i are initialized as $1/L$ so that the model starts as a moving-average (MA) autoregression model in the beginning. Updating of the weights with the least mean squared (LMS) algorithm in eq. (11), the weighted combination gradually adapts to the latest observations.

$$w_i(t+1) = w_i(t) + 2\mu\epsilon(t) u_h(t+1-i), i = 1, 2, ..., L \tag{11}$$

where μ is a convergence factor, $\epsilon(t) = \hat{u}_h(t+1) - u_h(t+1)$ is the predicted error.

By employing the adaptive filter, upcoming human inputs can be forecasted iteratively, thereby enabling the prediction of shared control movements. The iterative prediction algorithm of the shared control is shown in Algorithm 2.

Algorithm 2 Iterative Shared Control Motion Prediction

Input: $[\mathbf{u}_h(t-l), ..., \mathbf{u}_h(t)]$, $[x(t), y(t), z(t)]$, α

Output: $[[x(t+1), y(t+1), z(t+1)], ...[x(t+m), y(t+m), z(t+m)]]$

$[\mathbf{u}_h(t+1), ..., \mathbf{u}_h(t+m)] \leftarrow$ adaptive_filter_predict($[\mathbf{u}_h(t-l), ..., \mathbf{u}_h(t)]$)

$current\_position \leftarrow [x(t), y(t), z(t)]$

for $i = 1 : m$ **do**

  $\mathbf{u}_a(t+i) \leftarrow$autonomy_policy($current\_position$)

  $\mathbf{u}(t+i) = \alpha \cdot \mathbf{u}_a(t+i) + (1-\alpha) \cdot \mathbf{u}_h(t+i)$

  $[x(t+i), y(t+i), z(t+i)] \leftarrow$robot.step($\mathbf{u}(t+i)$)

  $current\_position \leftarrow [x(t+i), y(t+i), z(t+i)]$

end for

return $[[x(t+1), y(t+1), z(t+1)], ...[x(t+m), y(t+m), z(t+m)]]$

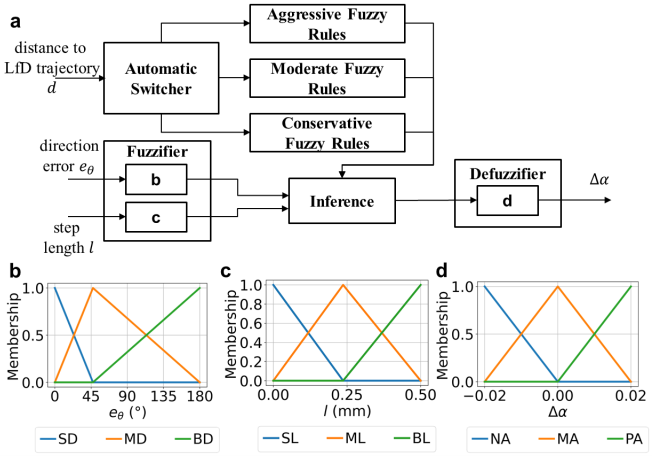

Fig. 4. Proposed automatic switching fuzzy logic for dominance modulation. (a) Block diagram of fuzzy logic dominance modulation with automatic switcher; (b) membership function for direction error e_θ; (c) membership function for step length l; (d) membership function for output $\Delta\alpha$.

B. Fuzzy Logic Dominance Factor Modulation

With the shared control prediction for the next steps, the dominance factor α could be adjusted accordingly. The future predictions are evaluated by the *direction error* and *step length*. *Direction error* e_θ is defined as the angle between the predicted displacement vector and movement vector from the generated trajectory, while *step length* l is simply defined as the Euclidean length of predicted displacement of the end-effector. The change of the dominance factor $\Delta\alpha$ is determined using different fuzzy logics. The fuzzy rules are designed with the following basic ideas:

- Aggressive fuzzy rules should be very 'strict' with human inputs or tend to seize control rights when human control inputs have errors.
- Conservative fuzzy rules tend to be tolerant towards flawed human inputs or tend to believe human inputs are correct and yield control to humans.
- Moderate fuzzy rules are neither too strict nor too permissive. This set of fuzzy rules lies between the conservative ones and the aggressive ones.

TABLE I
FUZZY LOGIC RULES TABLE

Mode	Step Length l	Direction Error e_θ		
		SD	MD	BD
Conservative	SL	NA	NA	NA
	ML	NA	NA	MA
	BL	NA	NA	PA
Moderate	SL	NA	MA	MA
	ML	MA	MA	MA
	BL	MA	MA	PA
Aggressive	SL	NA	MA	PA
	ML	MA	MA	PA
	BL	PA	PA	PA

To ensure tracking accuracy remains within controllable limits, we propose a method that switches between different sets of fuzzy logic rules based on the deviation to the planned trajectory, providing an adaptive autonomy scheme for the human operator to cooperate with. A diagram illustrating the proposed fuzzy-logic switcher is shown in Fig. 4. The switching mechanism is between Aggressive ($d > 0.4$mm), Moderate ($0.4\text{mm} \geq d > 0.2$mm), and Conservative ($0.2\text{mm} \geq d$) based on the distance to the planned path.

For the change of the dominance factor $\Delta\alpha$, we take direction error (ranging from 0 to 180 degrees) and step size (ranging from 0 to 5 mm) as inputs to the fuzzy logic system, which outputs the adjustment $\Delta\alpha$ (ranging from -0.02 to 0.02). Fuzzification is carried out with triangular membership functions (Fig. 4). The linguistic values are listed as follows:

- Direction Error e_θ: small direction error (SD), medium direction error (MD), and big direction error (BD);
- Step Length l: small step length (SL), medium step length (ML), and big step length (BL);
- Adjustment to Dominance Factor $\Delta\alpha$: negative alpha (NA), maintain alpha (MA), and positive alpha (PA).

The three sets of fuzzy logic rules are then designed as shown in Table I. Finally, the defuzzification process is implemented using the centroid approach.

IV. EXPERIMENT

A. Experiment Setup

An experiment was conducted using a robotic platform with the da Vinci surgical simulator and a hand controller (omega.7, Force Dimension). Five subjects (all male, 20-25 years old) with no prior surgical background or experience using surgical robots participated in the experiment. The experiments were approved by the Ethics Committee of Imperial College London (21IC7042). Each participant was informed about the experiment's purpose and protocol and signed a consent form before the experiment.

B. Experiment Modes and Tasks

The experiment compared the proposed shared control framework (auto mode) with simple shared control using constant weight (simple mode) and the conventional direct hand teleoperation (none). The experimental task is a trajectory following and reaching task and it simulates the teleoperated surgical operation under a camera view.

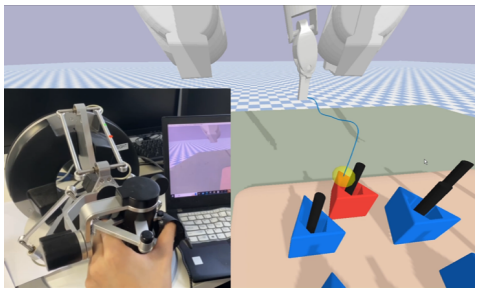

Fig. 5. Experiment setup. Left: Hand controller; Right: SurRoL simulation environment.

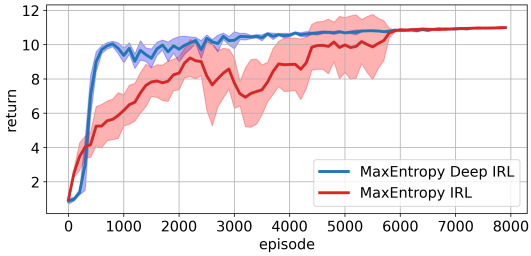

Fig. 6. Learning curves of Maximum Entropy IRL and Maximum Entropy Deep IRL.

The participants were first given time to familiarise with the robotic system and practice the task using different modes. They continued practicing until they felt confident in completing the tasks using the teleoperated system. In the test phase, two tasks were conducted. Task 1 involved participants following trajectories to reach a target using three modes, with four trials per mode. Each trial has a different trajectory to follow. Task 2 required participants to perform the same operation, with a step disturbance (10 mm, random direction) occurring at $t = 3$s in each trial. This step disturbance simulates a scenario when the operator is affected by the environmental noise and accidentally move the robotic end-effector out of the tracking path.

C. Evaluation Metrics

We used three metrics to evaluate the user performance: (1) *Integrated Path Error* is the integral of error along the trajectory [23]. (2) *Completion Time* is the time from the start of the trial to the gripper reaching the target. (3) *Motion Jerkiness* [24] evaluates the jerkiness of the trajectory.

In addition, we asked the participants to fill in a questionnaire when they finished the experiment to evaluate the subjective responses on operation efficiency ('I could reach the target in one go') and performance ('I could follow the path') using 5-Likert questions.

V. Results

A. LfD Training Results

To evaluate the MaxEntropy Deep IRL algorithm, both the capability of generalization and convergence speed have been taken into account. Comparison has been made between MaxEntropy Deep IRL, and our previous work using Maximum Entropy IRL (MaxEntropy IRL) [12]. The convergence is evaluated with a hand-crafted return value that represents the similarity between the current trajectory and the demonstration trajectory, this value becomes stable when the learned policy is close enough to the expert's behavior and could finish the task. This return is the sum of a handcrafted reward function without discounting, and it is given as:

$$return = \sum_{s \in \hat{\zeta}} r'(s), \quad r'(s) = \begin{cases} 1 & s \in \zeta \\ -1 & s \notin \zeta \end{cases} \tag{12}$$

where ζ and $\hat{\zeta}$ are demonstrated and learned trajectory.

Here, we define task completion success as reaching the target from a given starting state. Both algorithms demonstrate adequate success rates starting from states within space ($x \in [252.5, 262.5], y \in [-22.5, -12.5], z \in [363.0, 380.5]$, unit: mm). Among these 200 starting states, MaxEntropy IRL's learned policy has an average success rate of 73.9%, while MaxEntropy Deep IRL has an average success rate of 68.9%. However, as shown in Fig. 6, MaxEntropy Deep IRL has a faster convergence speed (within 1000 episodes) than MaxEntropy IRL (about 5000 episodes).

B. User study results

We have validated our human input prediction using adaptive filtering, achieving an average prediction error of 0.86 ± 0.35mm across the recorded inputs of 5 participants in the no-disturbance tasks.

Then, we carried out experiments to evaluate the whole shared control scheme. The automatic switching mechanism using fuzzy logic rules enables different authority allocation tendencies, which consequently leads to a change in dominance factor α. As shown in Fig. 7, the rules are set to be aggressive when the distance from the target trajectory is high, while when the distance is low, the switching mechanism changes to conservative.

Experimental results with participants were analyzed using Kruskal-Wallis tests with Dunn's multiple comparisons. The results are derived from all trials of participants. As shown in Fig. 8a, the proposed auto shared control method shows fewer path errors than direct hand teleoperation in both disturbance

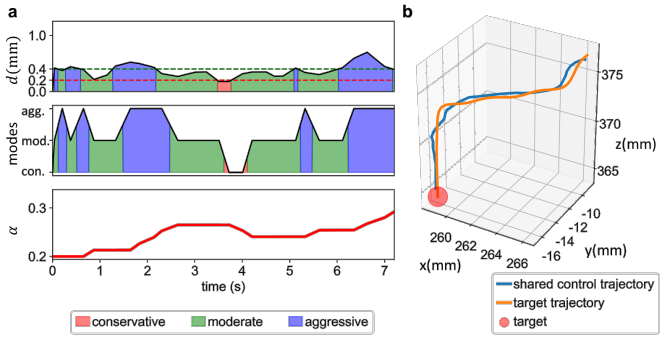

Fig. 7. Automatic switching of fuzzy logic rules and dominance factor modulation of a representative subject. (a) mode switching and α curve. (b) shared control and target trajectories.

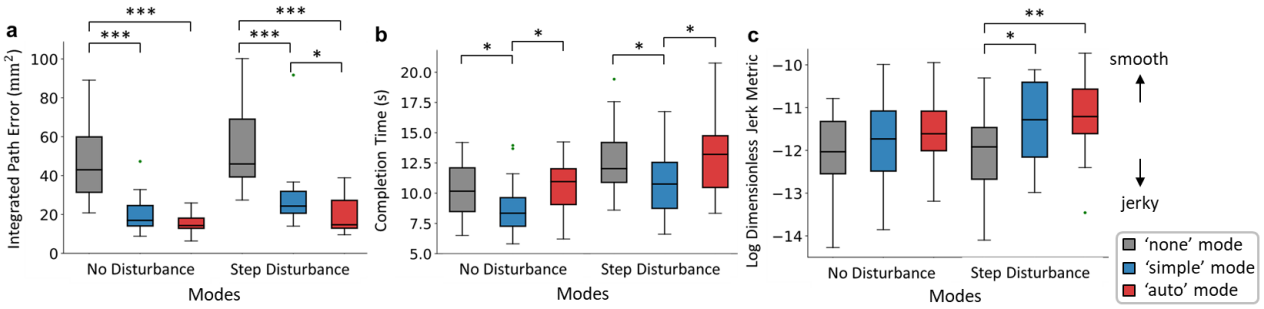

Fig. 8. Experimental results on (a) integrated path error; (b) completion time; (c) log dimensionless jerk metric. Asterisks denote significant effects at *p<0.05, ** p<0.01, *** p<0.001.

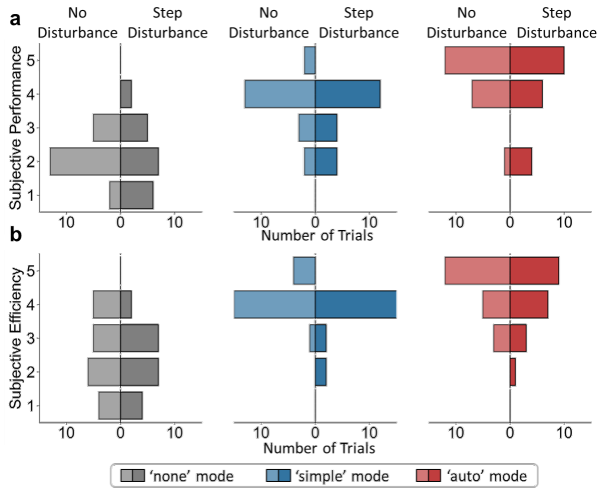

Fig. 9. Questionnaire results on (a) subjective efficiency and (b) performance. The horizontal axis is the number of trials. The vertical axis is the subjective score, the higher score represents positive responses.

($p = 2 \times 10^{-7}$) and non-disturbance ($p = 1 \times 10^{-7}$) tasks. In the non-disturbance task, the automatic shared control is also more accurate than the simple shared control mode ($p = 0.0349$).

Although the auto shared control mode improves the tracking performance, it is interesting to notice that the simple shared control with constant dominance factor has a shorter completion time than the automatic mode in both tasks with ($p = 0.0186$) and without ($p = 0.0373$) disturbance.

The control schemes do not affect the motion smoothness in task 1 (without disturbance). However, auto mode exerts smoother motion than direct teleoperation ($p = 0.0058$) and simple mode ($p = 0.0349$), when there is a disturbance to the trajectory.

The questionnaire results are shown in Fig. 9. The participants felt that the automatic share control has better performance (mean Likert scale 4.50 ± 0.76) than the direct hand teleoperation (2.15 ± 0.59). Similar to self-evaluated performance, the participants felt they could efficiently complete the task using the automatic shared control (4.45 ± 0.76) which is higher than the direct hand control (2.55 ± 1.10).

VI. Discussion

In this study, we proposed a human-centered shared control framework that enables the robotic agent to provide adaptive assistance to the operators. We applied MaxEntropy Deep IRL to train the robot with the expert's demonstrated trajectories. Compared to conventional maximum entropy IRL methods [12], MaxEntropy Deep IRL showed faster convergence. The success rates of demonstrative learning decreased by 5%, which might be a result of quicker convergence toward the demonstration trajectory, reducing chances for the agent to explore more states. The IRL policy was further extended into global continuous space by incorporating local path planning with a potential field method, resolving the limitations in discretized state spaces and inadequate exploration. To achieve dynamic authority modulation between human and robot, we developed a shared control scheme using adaptive filtering for the prediction of human movement and fuzzy logic for authority allocation.

The experiment using a haptic interface with the da Vinci surgical simulator showed that our approach could reduce the tracking error compared to direct teleoperation or simple shared control. It also provides superior motion smoothness when the operator receives environmental disturbances. For completion time, there was no clear improvement observed compared to direct teleoperation. Meanwhile, the simple shared control mode took participants the least time to finish the task. This may mean that the manipulation could be accelerated when the human operator has the dominant control with mild assistance from the robot, but when the robot took over the control, the increment step from the autonomy may have slowed down the overall pace. User feedback indicated that our approach is more efficient than direct teleoperation or a simple shared control scheme. In the future, we will extend the adaptivity of the shared control scheme by considering robotic control inaccuracy and incorporating haptic feedback to enhance user performance.

Acknowledgment

We thank the Hamlyn Centre at Imperial College London for providing the haptic devices used to carry out the work. We thank the subjects for participating in the experiment.

REFERENCES

[1] C. Freschi, V. Ferrari, F. Melfi, M. Ferrari, F. Mosca, and A. Cuschieri, "Technical review of the da vinci surgical telemanipulator," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 9, no. 4, pp. 396–406, 2013.

[2] R. Onofrio and P. Trucco, "Human reliability analysis (hra) in surgery: Identification and assessment of influencing factors," *Safety science*, vol. 110, pp. 110–123, 2018.

[3] T. Yamany, S. L. Woldu, R. Korets, and K. K. Badani, "Effect of postcall fatigue on surgical skills measured by a robotic simulator," *Journal of Endourology*, vol. 29, no. 4, pp. 479–484, 2015.

[4] M. S. Yasar and H. Alemzadeh, "Real-time context-aware detection of unsafe events in robot-assisted surgery," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2020, pp. 385–397.

[5] C. J. Payne, K. Vyas, D. Bautista-Salinas, D. Zhang, H. J. Marcus, and G.-Z. Yang, "Shared-control robots," *Neurosurgical Robotics*, pp. 63–79, 2021.

[6] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual review of control, robotics, and autonomous systems*, vol. 3, pp. 297–330, 2020.

[7] Z. Xie, Q. Zhang, Z. Jiang, and H. Liu, "Robot learning from demonstration for path planning: A review," *Science China Technological Sciences*, vol. 63, no. 8, pp. 1325–1334, 2020.

[8] S. Sharifzadeh, I. Chiotellis, R. Triebel, and D. Cremers, "Learning to drive using inverse reinforcement learning and deep q-networks," *arXiv preprint arXiv:1612.03653*, 2016.

[9] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *The International Journal of Robotics Research*, vol. 35, no. 11, pp. 1289–1307, 2016.

[10] K. Li and J. W. Burdick, "A function approximation method for model-based high-dimensional inverse reinforcement learning," *arXiv preprint arXiv:1708.07738*, 2017.

[11] G. Ning, H. Liang, X. Zhang, and H. Liao, "Inverse-reinforcement-learning-based robotic ultrasound active compliance control in uncertain environments," *IEEE Transactions on Industrial Electronics*, 2023.

[12] Z. J. Hu, Z. Wang, Y. Huang, A. Sena, F. R. y Baena, and E. Burdet, "Towards human-robot collaborative surgery: Trajectory and strategy learning in bimanual peg transfer," *IEEE Robotics and Automation Letters*, 2023.

[13] G. Li, Q. Li, C. Yang, Y. Su, Z. Yuan, and X. Wu, "The classification and new trends of shared control strategies in telerobotic systems: A survey," *IEEE Transactions on Haptics*, 2023.

[14] D. Zhang, Z. Wu, J. Chen, R. Zhu, A. Munawar, B. Xiao, Y. Guan, H. Su, W. Hong, Y. Guo *et al.*, "Human-robot shared control for surgical robot based on context-aware sim-to-real adaptation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7694–7700.

[15] R. Balachandran, H. Mishra, M. Cappelli, B. Weber, C. Secchi, C. Ott, and A. Albu-Schaeffer, "Adaptive authority allocation in shared control of robots using bayesian filters," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 11 298–11 304.

[16] H. Saeidi, F. McLane, B. Sadrfaidpour, E. Sand, S. Fu, J. Rodriguez, J. R. Wagner, and Y. Wang, "Trust-based mixed-initiative teleoperation of mobile robots," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 6177–6182.

[17] J. Xu, B. Li, B. Lu, Y.-H. Liu, Q. Dou, and P.-A. Heng, "Surrol: An open-source reinforcement learning centered and dvrk compatible platform for surgical robot learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1821–1828.

[18] M. Wulfmeier, P. Ondruska, and I. Posner, "Maximum entropy deep inverse reinforcement learning," *arXiv preprint arXiv:1507.04888*, 2015.

[19] M. Buhmann, P. Melville, V. Sindhwani *et al.*, "Reward shaping," *Encyclopedia of Machine Learning*, pp. 863–865, 2011.

[20] H. Mayer, I. Nagy, and A. Knoll, "Skill transfer and learning by demonstration in a realistic scenario of laparoscopic surgery," in *Proceedings of the IEEE International Conference on Humanoids CD-ROM*, 2003.

[21] A. Gottardi, S. Tortora, E. Tosello, and E. Menegatti, "Shared control in robot teleoperation with improved potential fields," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 3, pp. 410–422, 2022.

[22] M. Li, H. Cao, G. Li, S. Zhao, X. Song, Y. Chen, and D. Cao, "A two-layer potential-field-driven model predictive shared control towards driver-automation cooperation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 4415–4431, 2020.

[23] H. Boessenkool, D. A. Abbink, C. J. Heemskerk, F. C. van der Helm, and J. G. Wildenbeest, "A task-specific analysis of the benefit of haptic shared control during telemanipulation," *IEEE Transactions on Haptics*, vol. 6, no. 1, pp. 2–12, 2012.

[24] P. Gulde and J. Hermsdörfer, "Smoothness metrics in complex movement tasks," *Frontiers in neurology*, vol. 9, p. 615, 2018.