Evaluating the impact of reinforcement learning on automatic deep brain stimulation planning

Anja Pantovic, Caroline Essert

Evaluating the Impact of Reinforcement Learning on Automatic Deep Brain Stimulation Planning

Anja Pantovic[1] and Caroline Essert[1*]

[1]ICube, University of Strasbourg, Strasbourg, France.

*Corresponding author(s). E-mail(s): essert@unistra.fr;
Contributing authors: anja.pantovica@etu.unistra.fr;

Abstract

Purpose: Traditional techniques for automating the planning of brain electrode placement based on multi-objective optimization involving many parameters are subject to limitations, especially in terms of sensitivity to local optima, and tend to be replaced by machine learning approaches. This paper explores the feasibility of using Deep Reinforcement Learning (DRL) in this context, starting with the single-electrode use-case of Deep Brain Stimulation (DBS).

Methods: We propose a DRL approach based on deep Q-learning where the states represent the electrode trajectory and associated information, and actions are the possible motions. Deep neural networks allow to navigate the complex state space derived from MRI data. The chosen reward function emphasizes safety and accuracy in reaching the target structure. The results were compared with a reference (segmented electrode) and a conventional technique.

Results: The DRL approach excelled in navigating the complex anatomy, consistently providing safer and more precise electrode placements than the reference. Compared to conventional techniques, it showed an improvement in accuracy of 2.3% in average proximity to obstacles and 19.4% in average orientation angle. Expectedly, computation times rose significantly, from 2 to 18 minutes.

Conclusion: Our investigation into DRL for DBS electrode trajectory planning has showcased its promising potential. Despite only delivering modest accuracy gains compared to traditional methods in the single-electrode case, its relevance for problems with high-dimensional state and action spaces and its resilience against local optima highlight its promising role for complex scenarios. This preliminary study constitutes a first step towards the more challenging problem of multiple-electrodes planning.

Keywords: Reinforcement learning, Deep Q-learning, Deep Brain Stimulation, Path planning, Optimization

1 Introduction

Stereotactic electrode implantation encompasses a range of interventions among which Deep Brain Stimulation (DBS) and Stereoelectroencephalography (SEEG). DBS is a surgical treatment aimed at alleviating symptoms in patients with movement disorders like Parkinson's disease and essential tremors. The procedure [1] involves the insertion of one electrode into a deep brain target to stimulate it with high-frequency electrical impulses. Despite its invasive nature, this method has gained in popularity because it offers a flexible and reversible alternative to the permanent removal of functional areas, while effectively alleviating the symptoms of the disease. SEEG aims to accurately locate the epileptogenic zone in patients with pharmacoresistant focal epilepsy before surgical removal. Between 10 and 18 are implanted in selected brain areas to record neuronal activity through metallic contacts evenly spaced along their body [2].

If both implantation procedures suffer from a complex planning, SEEG presents an even greater complexity compared to DBS, due to a higher number of electrodes, a less precise targeting objective, and possible electrode conflicts. Indeed, while DBS targets one tiny nucleus, which may be assimilated to a single point in 3D space, SEEG electrodes have to ensure that a maximal number of metallic contacts are included in the areas selected for exploration in the gray matter.

Consequently, it can take up to 1h30 for DBS and several hours for SEEG of a collaborative effort of multiple clinicians, including neurosurgeons, neurologists, and neuroanatomists, to converge to an optimal and safe implantation strategy. This pivotal step requires a high degree of expertise and experience. For DBS, the standard placement suggested by the existing software has to be fine-tuned through a long trial-and-error approach. For SEEG, the trial and error process has to be repeated multiple times for all electrodes, as each electrode may interfere with the others. In this context, automated preoperative planning assistance is a much-awaited feature that many research teams have been trying to address, with the dual aim of saving valuable time and potentially increasing the efficiency of the procedures.

Most of the related works used traditional techniques based either on brute force or multi-objective optimization. In scenarios with a high number of parameters, such as SEEG, these approaches face a rapid growth of combinatorial possibilities and challenges in terms of convergence and sensitivity to local optima. These techniques tend to be replaced nowadays by machine learning approaches to scale up. However, for both DBS and SEEG, the limited availability of sufficiently large training datasets may hinder the use of such methods. In this context, Reinforcement Learning methods that do not require large training datasets appear to be a promising and viable alternative.

This paper presents a preliminary study initiating an exploration of the potential of using Deep Reinforcement Learning (DRL) techniques for brain electrode placement planning. To start with a simple application, this paper focuses on the single-electrode use-case of DBS.

More complex, multi-electrode planning is left for future work. In the following sections, we detail a DRL method based on deep Q-learning, with states indicating the electrode trajectory, actions representing possible motions, and a reward function prioritizing safety and accuracy. Our findings are benchmarked against a reference

electrode and a conventional technique. Finally we discuss and conclude about the potential of DRL in the context of brain electrode placement planning.

2 Related works

In the past few decades, numerous approaches have been introduced to alleviate the workload of clinicians during the planning phase and enhance the information considered in determining an optimal trajectory. The wide and exhaustive survey on surgical planning assistance dedicated to keyhole and percutaneous surgery published by Scorza et al. [3] details the various techniques proposed to automate the computation of needles or electrodes placement in general. Among other applications, it covers the placement of brain electrodes for stereotactic neurosurgery, including DBS.

The automatic computation of optimal placement for DBS electrodes has been relatively well studied in the literature. If early planning tools were still requiring substantial manual intervention [4–6], subsequent approaches proposed a more automated assistance to the task [7–13]. Some of these methods emphasized more particularly the safety by maximizing the separation between the candidate trajectory and critical structures [8, 11], while others considered a broader range of placement rules, categorized as hard or soft constraints [7, 9, 10, 12, 13]. Regardless of the specific approach, these methods aiming at ensuring feasibility and safety, were exclusively based on conventional methods to find an optimum. Some authors used heuristic-based approaches with gradient-free optimization algorithms [9, 10] or Pareto front optimization [13], while many simply used a brute force method consisting in an exhaustive exploration of the whole search space [7, 8, 11, 12]. The reported execution times ranged from 15 minutes to a few seconds. The approaches accounted for 3 to 6 degrees of freedom per electrode, depending on whether the target point was manually fixed by a clinician prior to the search or part of the explored variables.

As pointed in the introduction however, in multiple-electrode scenarios with a high number of parameters, brute force algorithms are confronted with the rapid growth of combinatorial possibilities, and optimization methods are limited by their sensitivity to local optima. These techniques tend to be replaced nowadays by machine learning approaches in order to scale up, but to the best of our knowledge, no attempt has yet been made to introduce machine learning approaches in this field. Indeed, this problem is limited by the lack of sufficiently large training datasets and the disparity in image parameters, which make it difficult to consider using conventional supervised learning approaches.

In this context, reinforcement learning (RL) seems to be a promising approach for overcoming these limitations despite its longer computation times. This approach doesn't require large training datasets but instead gradually learns how to converge via interactions with its environments, and adapts itself to environment changes.

In the field of medicine, particularly in the context of path planning, RL has, so far, found application in robot-assisted surgery [14, 15]. In neurosurgery, Segato et al. [14] explored GPU-based Asynchronous Advantage Actor-Critic (GA3C) RL-based path planning for steerable catheters. This approach also encountered difficulties with

training efficiency. Guanglin et al. [15] proposed a heuristically accelerated deep Q-learning algorithm for needle insertion that integrates a fuzzy inference system into the framework, combining heuristic policies and RL methods. Simulations in this study showed promising results, including substantial reductions in training episodes.

3 Materials and Methods

3.1 General Objective

In this study, we undertake a comparative analysis between an automated electrode trajectory planning method based on deep reinforcement learning (DRL) and conventional optimization techniques. Our primary objective is to evaluate the effectiveness of the DRL approach in determining the optimal electrode trajectory for Deep Brain Stimulation (DBS), with a focus on safety and efficacy. The DRL-based method is benchmarked against the results obtained in our prior study [9], representing conventional optimization methods in the context of DBS electrode trajectory planning, where the following *constraints* governing the electrode placement for DBS were collected from experienced neurosurgeons:

1. *Ensuring the electrode tip is within the target.*
2. *Proper positioning of the insertion point:* The electrode must be implanted from the upper surface of the scalp, adhering to accessibility and aesthetic considerations.
3. *Restricting the maximal length of the trajectory:* A maximum trajectory length limits the search space avoiding unnecessary complexity.
4. *Avoiding risky structures:* To minimize risk, it's crucial to avoid critical structures such as ventricles and vessels. Avoiding trajectories through cortical sulci serves as a strategy to avoid vessels, therefore sulci are also considered as a critical structure.
5. *Minimizing the path length:* Shorter trajectories reduce the risk of imprecision.
6. *Maximizing the distance between the electrode and risky structures.*
7. *Optimizing the electrode orientation:* Aligning the trajectory axis with the target's principal axis facilitates exploration of different depths within the target.
8. *Placing the tip as close as possible to the center of the target.*

In the prior study, constraints were addressed through an aggregative cost function:

$$f(X) = k_d \cdot f_{depth}(X) + k_r \cdot f_{risk}(X) + k_o \cdot f_{ori}(X) + k_c \cdot f_{center}(X) \tag{1}$$

which combines the cost functions associated to constraints #5 (f_{depth}), #6 (f_{risk}), #7 (f_{ori}) and #8 (f_{center}) respectively with their corresponding weight factors. In the newly proposed method, these constraints are reflected in a reward function of the DRL agent, as explained in detail later in this section.

3.2 Reinforcement Learning approach

3.2.1 Deep Q-learning

Numerous algorithms have been developed in the field of reinforcement learning, each tailored to specific challenges. Our task presented a complex challenge due to its vast

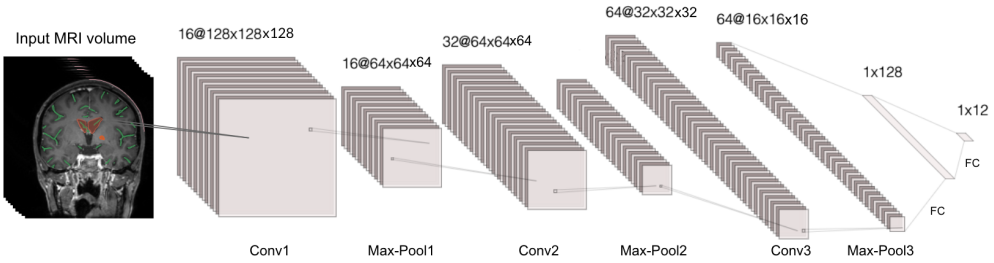


Fig. 1: Deep Q-network architecture. The architecture consists of Convolutional (Conv), MaxPooling (MaxPool) and Fully Connected (FC) layers.

state space derived from MRI data. Deep Q-learning [16] was our methodology of choice due to its adeptness at handling large state spaces.

Deep Q-learning combines the capabilities of DNNs with the principles of traditional Q-learning [17]. Q-learning seeks to determine the expected future rewards for each state-action pairing, enabling an agent to discern optimal actions in varying states. However, vast or complex state spaces make it unfeasible to maintain a Q-value table for every state-action combination. Deep Q-learning addresses this by using DNNs to approximate Q-values, thus efficiently generalizing across expansive state spaces. The agent undergoes interactions with an environment, acquiring experiences that encompass state, action, reward, and the subsequent state. These experiences fill a replay buffer. As training progresses, random experiences are sampled from this buffer, which ensures data decorrelation and augments the stability of the learning process. The network is trained to predict Q-values such that the discrepancy between predicted and target Q-values, based on the Bellman equation, is minimized. In the context of our study, deep Q-learning provides a robust framework to handle the intricate state space defined by neural structures and electrode trajectory parameters, guiding the agent to propose safe and effective trajectories.

3.2.2 Architecture

Our deep reinforcement learning agent is backed by a DNN that processes the state representations and estimates Q-values for the possible actions. The architecture is represented in Fig.1 The model leverages the Huber loss and is optimized using the Adam optimizer with a learning rate of 0.001.

3.2.3 Solution Space: States and Actions

In our deep Q-learning framework, the state is defined by the current electrode trajectory, represented as a cylinder that extends from the entry point (EP), placed within the insertion zone, to the target point (TP) within the predefined target, combined with the positions of critical structures derived from the MRI scans. In each episode, EP is initialized at a random position within the insertion zone, while TP is consistently initialized at the center of the target. Actions are possible motions of the electrode, i.e. translations of the EP and TP across the x, y, and z axes. We allow the

agent to move the EP or TP in any direction, resulting in a total of 12 possible actions. Like all reinforcement learning approaches, our algorithm learns to find the optimal trajectory (EP/TP pair) through trial and error using feedback from its actions and some rewards detailed in the next section.

3.2.4 Reward Function

The immediate reward after taking an action is primarily influenced by the safety and efficacy of the resulting state (i.e., electrode placement) and is designed as:

$$R(s, a, s') = -c_d \cdot R_l(s') - c_r \cdot R_r(s') - c_o \cdot R_o(s') - c_c \cdot R_c(s') + P_i(s') + P_z(s') \quad (2)$$

where s is the current state, a is the action taken and s' is the resulting next state. Despite the reward being a function of s, a and s', it's notable that all terms are exclusively dependent on s'. The components of this function, normalized in line with the previous approach [9], are defined as follows:

- $R_l(s')$ accounts for the length of the trajectory (*constraint #5*):

$$R_l(s') = \frac{\text{distMin(TP, EP) - distTargetScalp}}{\text{maxPathLength - distTargetScalp}} \quad (3)$$

- $R_r(s')$ measures the agent's proximity to critical structures (*constraint #6*):

$$R_r(s') = \max\left(\frac{10.0 - \text{distMin(criticalStructures, electrodeTrajectory)}}{10.0}, 0\right) \quad (4)$$

- $R_o(s')$ denotes the alignment of the electrode's trajectory with the principal axis of the target structure (*constraint #7*):

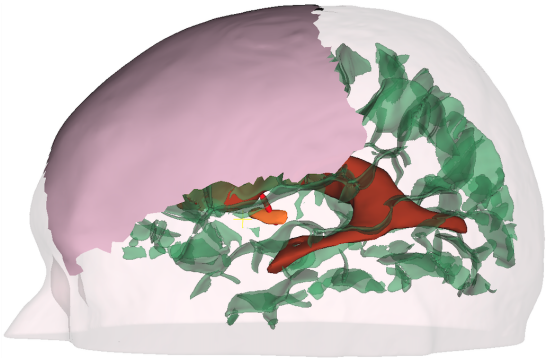
$$R_o(s') = \frac{\text{angle(electrodeTrajectory, mainAxis(target))}}{90.0} \quad (5)$$

- $R_c(s')$ quantifies how close TP is to the center of the target (*constraint #8*):

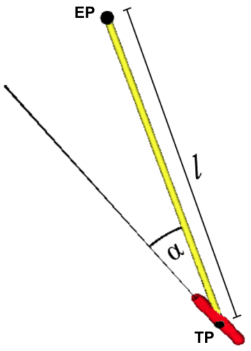
$$R_c(s') = \frac{\text{distMin(TP, center(target))}}{10.0} \quad (6)$$

- $P_i(s')$ is a penalty received when the electrode trajectory is less than 2mm away from critical structures, deterring the agent from unsafe trajectories.
- $P_z(s')$ is a penalty applied if the state chosen is outside the predefined zones, ensuring that EP and TP remain within the insertion zone and target. When an action leads to a state (s') outside these zones, a penalty of -10 is given and the state remains unchanged ($s' = s$).

Constants c_d, c_c, c_r, c_o are used to weight their contributions to the reward function and can be fine-tuned and adjusted based on the preferences of individual surgeons.



(a) Scalp, insertion zone (pink), target (red), GPi (orange), ventricle (brown) and sulci (green).



(b) Target (red), electrode (yellow), path length l and orientation angle α.

Fig. 2: Visualisation of the RL environment (a) and optimization constraints (b).

3.3 Data and Preprocessing

To ensure a meaningful comparison with the prior study [9], our DRL algorithm was evaluated on data from the same cohort of patients. This data consists of co-registered pre-operative 3T T1-weighted MRI images (1 mm x 1 mm x 1 mm, Philips Medical Systems) and post-operative CT scans (0.44 mm x 0.44 mm x 0.6 mm, GE Healthcare VCT 64) obtained from patients who underwent DBS procedure. We focused on the same 30 electrode trajectories, from 18 different patients, that were already segmented and reconstructed, before being planned and evaluated in our previous work.

In the preprocessing stage of this study, to train the DRL agent, we employed data from 13 additional patients, accounting for another 25 trajectories. This additional data was prepared as follows. First, the scalp was segmented from the pre-operative MRI using intensity-based segmentation in 3D Slicer [18]. The resulting scalp segmentation represented the initial broad space within which the electrode could potentially enter. To exclude implausible entry points, we applied a series of constraints. We focused only on the upper surface of the scalp, as the lower regions are avoided for accessibility and aesthetic reasons. Trajectories that would exceed a maximum allowed path length were discarded. The resulting part of the scalp will in this paper be referred to as *insertion zone*, and is represented in opaque pink color in Fig. 2a.

For consistency with the previous study [9], the electrode contacts were segmented from the post-operative CT scans. This oblong structure, shown in Fig. 2 in red, was used as the *target*, ensuring that comparisons were solely influenced by trajectory computation rather than variances in location.

Ventricles were segmented using the FreeSurfer software [19] with atlas-based segmentation techniques applied to the MRI scans. Cortical sulci were identified and segmented with the BrainVisa Morphologist pipeline [20] from the MRI data.

For the purpose of validating the proposed trajectories, co-registration between the post-operative CT scans and the pre-operative MRIs was performed using the 3D

Slicer Registration module. This registration process ensured the accurate alignment of the implanted electrodes with the corresponding pre-operative anatomical data, facilitating a comprehensive assessment of the proposed trajectories against reference trajectories. Proposed trajectories were segmented using the thresholding tool in 3D Slicer, allowing us to isolate DBS electrodes from the surrounding brain tissue.

3.4 Experiments and Validation

In this section, we provide an overview of our experimental setup, encompassing the training and validation procedures for the proposed deep Q-learning agent.

3.4.1 Training the Agent

To ensure the generalization of the agent, we used the data collected from 13 different patients, with 25 distinct trajectories as previously described. This diversity was essential because each patient exhibits unique brain morphology, including variations in sulci positions and ventricle shapes. By training on diverse patient data, the agent learned to adapt better to these patient-specific differences. Pre-operative MRI scans were used for training, while post-operative CT scans were used to validate the proposed electrode placements.

For training the deep Q-learning agent, the preprocessed data were cropped to a size of $128 \times 128 \times 128$, capturing the relevant areas of the head and brain. This input volume contained labels for critical structures, insertion and target zone, and the current electrode position, which was updated as the agent performed actions.

During the training phase, our agent interacted with the environment, accumulating state-action-reward-next state tuples, which were stored in a memory buffer with a capacity of 10,000 experiences. Random mini-batches of 32 experiences were sampled from this buffer for training the network. The state representations contained information about the patient's specific anatomy, electrode placement, and proximity to critical structures,which, together with reward function, allowed the agent to learn how to find the shortest and safest trajectory to the target.

An epsilon-greedy strategy was employed to balance exploration and exploitation. The agent starts with an initial epsilon value of 1. Over time, this epsilon value is adjusted using the formula

$$\epsilon = max(min\_eps, \epsilon \times eps\_decay^{episode}) \tag{7}$$

where min_eps was set to 0.01 and eps_decay to 0.995. This allowed for extensive early exploration, reducing random actions over time while maintaining a slight chance of exploration throughout the learning process. To maintain stable learning and prevent abrupt policy changes, a target network approach was employed. The weights of the primary Q-network were periodically copied to a target network every 100 steps.

Terminal states were defined under two conditions to ensure stability and progress:

1. *Convergence to an optimal trajectory:* over a sliding window of ten consecutive actions, no significant improvement in the average reward, i.e. the agent consistently suggesting the same trajectory, signifies a potential optimal solution.

2. *Reaching maximum episode length:* an episode is terminated when the maximum episode length of 500 steps is reached. As the largest sum of their diameters is 187 voxels, this maximum episode length provided ample room for the agent to explore and reach any electrode position within the episode.

To create a meaningful comparison with the prior work [9], we focused on the same set of weights for reward function (2): $c_d = 1, c_r = 1, c_o = 4, c_c = 4$.

Training was performed on 4 NVIDIA GeForce RTX 2080 Ti GPUs, with 11GB RAM each. The agent was trained for 1000 episodes, taking 18h04' on the whole training dataset.

3.4.2 Validation

Validation was done on the set of 30 trajectories from [9], unseen during training. For each case, the agent required re-training for a certain number of episodes to adapt to the patient's unique morphology. This process continued until the agent converged, consistently proposing the same electrode placement for ten consecutive episodes. On average, it took 21 episodes to ensure adaptation to the differences in the environment, such as variations in the shape and position of sulci unique to each patient.

4 Results and Discussion

The trajectories proposed by DQL (T_{dql}) are compared with those generated using the conventional method [9] (T_{conv}) and the manual trajectories (T_{ref}). Numerical results evaluating proximity to critical structures and orientation angle are shown in Table 1.

When comparing T_{dql} to T_{ref}, we observe that in all cases, T_{dql} consistently maintains a larger distance from the sulci (4.5mm on average), indicating safer trajectories. Furthermore, T_{dql} remains on average further away from the ventricles by 1.7mm compared to T_{ref}. In cases where T_{ref} maintains a greater distance from ventricles, it passes much closer to the sulci than T_{dql}. Figure 3 illustrates the trajectories for Case #11, where T_{dql} (in yellow) can be compared to T_{ref} (in blue), highlighting their proximity to sulci (in green) and a ventricle (in brown).

In comparison to T_{conv}, we observe that T_{dql}, on average, maintains slightly greater distances from both sulci and ventricles while being better aligned with the desired target. The average angle between T_{conv} and the ideal orientation is 11.11 degrees, whereas T_{dql} shows a slightly improved average angle of 8.96 degrees.

As expected, the DRL method requires significantly more computation time. Adapting to individual patient environments and proposing an optimal trajectory took between 14 and 23 minutes, averaging 18.6 minutes per patient. In contrast, the conventional method had an average planning time of 2.3 minutes per patient.

To further improve the efficiency of the DQL approach, several strategies could be considered in future works. Increasing the number of patients in the training dataset could make the agent more generalizable and potentially reduce the adaptation time. Accumulating information learnt after each new case, in the spirit of continual learning, may also help develop the agent's effectiveness. Incorporating heuristically accelerated DQL algorithms proposed, along the lines of [15], could be considered to improve training efficiency but perhaps at the cost of a higher sensitivity to local minima.

The proposed approach can be tailored to suit different imaging modalities and surgical requirements and constraints through modifications of the reward function, provided that all the relevant anatomical structures can be well identified and segmented.

5 Conclusion

Our investigation into the application of Deep Reinforcement Learning for DBS electrode trajectory planning highlights its promising potential. While the study demonstrates modest accuracy gains compared to traditional methods in the case of a single electrode, it constitutes a proof of concept that showed its suitability in the context of brain electrode positioning. This study highlighted the relevance of the DRL

Table 1: Comparison between the trajectory produced by DRL planning, T_{dql} (with $c_d = 1/c_c = 1/c_r = 4/c_o = 4$), trajectory produced using conventional method, T_{conv} (with $k_d = 1/k_c = 1/k_r = 4/k_o = 4$), and the reference trajectory, T_{ref}.

Target	Case	Distance to sulci (mm)			Distance to ventricles (mm)			Orientation (deg.)	
		T_{ref}	T_{conv}	T_{dql}	T_{ref}	T_{conv}	T_{dql}	T_{conv}	T_{dql}
GPi	1	2.052	6.573	5.916	5.422	12.204	6.540	19.22	3.80
	2	6.222	8.206	7.803	8.453	8.461	8.105	5.82	3.64
	3	6.042	6.299	7.720	2.289	2.254	4.209	0.46	7.60
	4	0.054	9.424	7.832	8.849	4.375	6.817	24.17	9.61
	5	4.396	7.491	7.562	9.844	7.120	8.203	7.15	7.00
	6	1.263	8.103	7.623	7.603	7.654	9.937	23.36	10.23
	7	5.714	11.677	9.827	1.805	5.347	7.918	14.97	10.54
	8	0.555	8.918	8.708	9.546	1.981	3.270	16.72	14.75
	9	0.648	7.606	7.600	11.424	14.876	14.035	10.15	10.03
	10	3.157	9.178	8.557	11.272	11.854	11.928	6.38	3.85
	11	1.480	5.961	7.164	8.866	10.424	9.212	5.78	5.53
	12	3.228	6.795	7.102	10.483	17.704	15.302	23.98	20.01
	13	6.140	6.809	7.348	8.614	10.015	9.517	4.40	3.47
	14	3.892	6.446	8.000	9.975	10.049	9.193	3.20	8.06
STN	15	5.746	8.133	6.890	10.289	13.303	13.102	10.02	2.94
	16	2.850	5.379	7.010	9.241	11.225	10.020	7.54	8.92
	17	0.750	8.301	7.427	2.590	7.306	10.810	20.60	8.93
	18	2.790	7.107	7.403	2.932	7.484	7.301	13.53	11.03
	19	3.490	5.753	7.477	3.818	6.375	5.984	8.66	12.84
	20	1.740	4.900	6.324	7.365	7.809	7.538	18.17	19.22
	21	2.425	4.768	5.099	5.327	3.613	5.002	23.98	15.78
	22	2.748	5.365	7.403	9.933	9.069	9.012	3.34	4.32
	23	1.946	4.858	7.708	4.460	7.433	5.016	9.89	5.46
	24	1.184	7.275	6.631	3.917	9.196	7.503	12.53	9.20
	25	2.239	7.958	8.078	7.451	9.751	9.877	4.99	6.62
VLc	26	0.150	6.426	5.107	2.316	0.567	4.280	11.76	9.07
	27	1.715	5.186	5.203	4.893	9.097	8.917	9.18	9.21
	28	1.409	7.356	7.210	3.163	7.652	9.302	20.50	16.80
	29	7.429	10.001	9.428	5.439	6.436	7.273	5.15	4.70
	30	1.366	6.996	7.024	5.728	5.187	8.427	7.78	5.64
AVERAGES		2.827	7.169	7.336	6.777	8.193	8.450	11.11	8.96

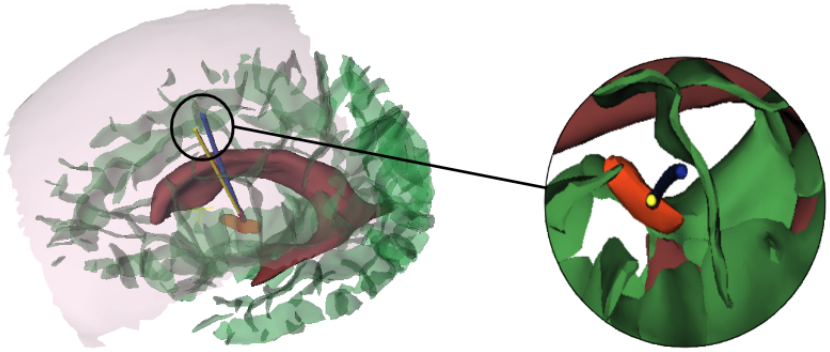


Fig. 3: Visualisation of the results for Case #11. Planned trajectory T_{dql} is represented in yellow, while the reference trajectory T_{ref} is represented in blue.

method for complex scenarios with high-dimensional state and action spaces and its robustness against local optima. This preliminary research represents an initial step towards the more challenging problem of multi-electrode planning. Future work will focus on scaling up and extending the application of DRL to address the high complexity of Stereoelectroencephalography (SEEG) planning, which involves a greater number of electrodes and intricate targeting objectives.

Acknowledgments. The authors would like to thank Pierre Jannin, INSERM Research Director at the University of Rennes 1 / LTSI / INSERM France, and Claire Haegelen, Professor at the University of Lyon 1 UCB / CRNL / INSERM / CNRS and Neurosurgeon at the Civil Hospital of Lyon, France, who kindly allowed us to use the data prepared for [9].

Declarations

Funding: This work was supported by ArtIC "Artificial Intelligence for Care" grant (ANR-20-THIA-0006-01), co-funded by Région Grand Est, Inria Nancy - Grand Est, IHU Strasbourg, University of Strasbourg and University of Haute-Alsace, France.
Conflict of interest: The authors have no conflict of interest to declare.
Ethics approval This research study was conducted retrospectively from anonymised data, in accordance with the ethical standards of the institution and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.
Informed consent: Informed consent was obtained from all individual participants included in the study.

References

[1] Benabid, A.L., Chabardes, S., Mitrofanis, J., Pollak, P.: Deep brain stimulation of the subthalamic nucleus for the treatment of parkinson's disease. The Lancet Neurology **8**(1), 67–81 (2009)

[2] Talairach, J., Bancaud, J.: Lesion, "irritative" zone and epileptogenic focus. Stereotactic and Functional Neurosurgery **27**(1-3), 91–94 (1966)

[3] Scorza, D., El Hadji, S., Cortés, C., Bertelsen, A., Cardinale, F., Baselli, G., Essert, C., De Momi, E.: Surgical planning assistance in keyhole and percutaneous surgery: A systematic review. Medical Image Analysis **67**, 101820 (2021)

[4] Bourbakis, N., Awad, M.: A 3-D visualization method for image-guided brain surgery. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics **33**(5), 766–781 (2003)

[5] Fujii, T., Emoto, H., Sugou, N., Mito, T., Shibata, I.: Neuropath planner-automatic path searching for neurosurgery. In: Proceedings of CARS'03, vol. 1256, pp. 587–596 (2003). Elsevier

[6] Vaillant, M., Davatzikos, C., Taylor, R., Bryan, R.: A path-planning algorithm for image-guided neurosurgery. In: Proceedings of CVRMed-MRCAS'97. Springer LNCS, vol. 1205, pp. 467–476 (1997)

[7] Bériault, S., Subaie, F.A., Collins, D.L., Sadikot, A.F., Pike, G.B.: A multi-modal approach to computer-assisted deep brain stimulation trajectory planning. International Journal of Computer Assisted Radiology and Surgery **7**(5), 687–704 (2012)

[8] Brunenberg, E.J.L., Vilanova, A., Visser-Vandewalle, V., Temel, Y., Ackermans, L., Platel, B., Haar Romeny, B.M.: Automatic trajectory planning for deep brain stimulation: A feasibility study. In: Proceedings of MICCAI'07. Springer LNCS, vol. 4791, pp. 584–592 (2007)

[9] Essert, C., Haegelen, C., Lalys, F., Abadie, A., Jannin, P.: Automatic computation of electrode trajectories for deep brain stimulation: a hybrid symbolic and numerical approach. International journal of computer assisted radiology and surgery **7**(4), 517–532 (2012)

[10] Essert, C., Fernandez-Vidal, S., Capobianco, A., Haegelen, C., Karachi, C., Bardinet, E., Marchal, M., Jannin, P.: Statistical study of parameters for deep brain stimulation automatic preoperative planning of electrodes trajectories. International Journal of Computer Assisted Radiology and Surgery **10**(12), 1973–1983 (2015)

[11] Shamir, R., Tamir, I., Dabool, E., Joskowicz, L., Shoshan, Y.: A method for planning safe trajectories in image-guided keyhole neurosurgery. In: Proceedings of MICCAI'10. Springer LNCS, vol. 6363, pp. 457–464 (2010)

[12] Liu, Y., Konrad, P.E., Neimat, J.S., Tatter, S.B., Yu, H., Datteri, R.D., Landman, B.A., Noble, J.H., Pallavaram, S., Dawant, B.M., D'Haese, P.-F.: Multisurgeon,

multisite validation of a trajectory planning algorithm for deep brain stimulation procedures. IEEE Transactions on Biomedical Engineering **61**(9), 2479–2487 (2014)

[13] Hamzé, N., Voirin, J., Collet, P., Jannin, P., Haegelen, C., Essert, C.: Pareto front vs. weighted sum for automatic trajectory planning of deep brain stimulation. In: Proceedings of MICCAI'16. Springer LNCS, vol. 9900, pp. 534–541 (2016)

[14] Segato, A., Sestini, L., Castellano, A., De Momi, E.: GA3C Reinforcement Learning for Surgical Steerable Catheter Path Planning. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 2429–2435 (2020)

[15] Guanglin, J., Qian, G., Tianwei, Z., Lin, C., Zhenglong, S.: A heuristically accelerated reinforcement learning-based neurosurgical path planner. Cyborg and Bionic Systems **4**, 0026 (2023)

[16] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**, 529–533 (2015)

[17] Watkins, C.J.C.H., Dayan, P.: Q-learning. Machine Learning **8**, 279–292 (1992)

[18] Kikinis, R., Pieper, S.D., Vosburgh, K.G.: 3D Slicer:A platform for subject-specific image analysis, visualization, and clinical support. In: Intraoperative Imaging and Image-guided Therapy, pp. 277–289. Springer, New York (2014)

[19] Fischl, B.: Freesurfer. NeuroImage **62**(2), 774–781 (2012)

[20] Rivière, D., Régis, J., Cointepas, Y., Papadopoulos-Orfanos, D., Cachia, A., Mangin, J.: A freely available Anatomist/BrainVISA package for structural morphometry of the cortical sulci. Proceedings of HBM'03, NeuroImage **19**(2, Supplement), 1825–1826 (2003)