IEEE International Conference on Robot & Human Interactive Communication (RO-MAN 2021)

Interactive Learning of Sensor Policy Fusion

Bart Bootsma[1,2], Giovanni Franzese[1], and Jens Kober[1]

Abstract— **Teaching a robot how to navigate in a new environment only from the sensor input in an end-to-end fashion is still an open challenge with much attention from industry and academia. This paper proposes an algorithm with the name "Learning Interactively to Resolve Ambiguity" (LIRA) that tackles the problem of sensor policy fusion extending state-of-the-art methods by employing ambiguity awareness in the decision-making and solving it using active and interactive querying of the human expert. LIRA, in fact, employs Gaussian Processes for the estimation of the policy's confidence and investigates the ambiguity due to the disagreement between the single sensor policies on the desired action to take. LIRA aims to make the teaching of new policies easier, learning from human demonstrations and correction.**

The experiments show that LIRA can be used for learning a sensor-fused policy from scratch or also leveraging the knowledge of existing single sensor policies. The experiments focus on the estimation of the human interventions required for teaching a successful navigation policy.

I. Introduction

The use of robotics in assisting humans and performing several different tasks is changing our lives. Unfortunately, for the execution of each new task, a time-consuming reprogramming of the robot is necessary. This approach limits the adaptability of robots to new conditions. Ideally, robots should quickly adapt to unknown situations and learn directly from experience or human demonstrations. What if even a child could teach a robot to drive autonomously? Learning from Demonstration (LfD) is the field that studies how humans can teach robots this way [1].

Within LfD, there are different input modalities and learning methods for teaching a new behavior. For example, Behavioural Cloning (BC) collects the demonstration data and then trains the robot policy to clone that desired behavior. It can, however, be challenging to foresee if the demonstrations lead to the desired behavior [2]. A solution is to have the human in the learning loop and supervise the robot with corrective feedback. Instead of providing all demonstrations at the start, the human is asked to supervise the training and provide corrective feedback during multiple iterations. This is the field of Interactive Imitation learning (IIL) methods. Examples like HG-DAgger [3] and COACH [4], have shown promising results in an easy, safe, and fast way of teaching robots. Additionally, IIL methods teach the robot in an online fashion, in contrast to the offline BC.

*This work was supported by European Research Council Starting Grant TERI "Teaching Robots Interactively", project reference 804907.

[1]Cognitive Robotics, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands {g.franzese, j.kober}@tudelft.nl

[2]DOBOTS, Stationsplein 45 - D1.125, 3013AK Rotterdam, The Netherlands bart.bootsma@dobots.nl

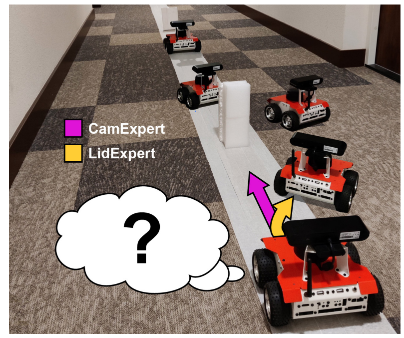


Fig. 1: Example of ambiguous policy fusion: the CAMERA-based policy does not perceive the white obstacle and deems to drive ahead. However, the LiDAR perceives the obstacle and wants to steer on the side. The human is queried to take control. As a consequence, LIRA infers the correct action for the sensor-fused policy.

This work focuses on learning navigation behavior for an autonomous mobile robot. Mobile robots often use a combination of sensors to navigate safely through the environment. As each sensor modality has its strengths, they are often responsible for a specific behavior. The fusion of different sensors has the scope of obtaining better performance than using them individually [5]. Learning how to leverage the strengths of different sensors, under human supervision, is the scope of this paper.

In the context of sensor policy fusion, we define an ambiguous situation as a novel scenario where only one policy can perceive the relevant feature, e.g., an obstacle, and then they would not agree on the control decision. Thus, the source of action disagreement is given from a mismatch in the world perception. For example, Fig. 1 shows an ambiguous situation the robot might encounter. In fact, the camera cannot distinguish the white obstacle from the line and the LiDAR can only detect the presence of an obstacle directly in front of the robot. The single-sensor expert policies may propose different and conflicting actions in controlling of the robot. Which would be the safest (and desired) behaviour?

Thus, in case the sensor-fused policy (called 'novice' in this paper) is uncertain and the sensor fusion is ambiguous (conflicting actions), LIRA queries the user to take control of the robot. The corrective demonstration is used to update the novice, so it learns how to solve the current ambiguity in the future.

To summarize, LIRA proposes an alternative to learning a fused policy only from human demonstrations, allowing the fusion of existing single-sensor policies. However, rather than employing heuristics, in case the single sensor policies do not agree about the desired control action, LIRA solves the ambiguity interactively with queried user corrections.

II. Related Works and Background

This work follows the conceptual reasoning introduced in [6] where "Learning Interactively to Resolve Ambiguity" (LIRA) was employed for solving the ambiguity in the selection of task parametrization, when the kinesthetic demonstrations were not informative enough. Differently, this current work tackles the problem of sensor policy fusion in robot control, investigating how to solve the fusion ambiguity with minimum user effort and how to learn reliable navigation tasks.

Although previous works used Learning from Demonstrations (LfD) with a single sensor input [7], [8], LIRA aims to learn navigation policies in an end-to-end fashion, mapping multiple sensor inputs to robot control actions, learning from human demonstrations and corrections.

However, other methods perform multiple sensor fusion. For example, [9] combines RGB and Depth images as input of a Deep Neural Network. Furthermore, [10] shows that a camera and laser range finder perform better together than alone while learning to drive autonomously. For learning complex tasks, the methods in [11] and [12] fuse sensor input with state measurements (e.g., speed, position) and higher-level goal commands in the same network.

For learning robust sensor policy fusion, [13] applies Sensor Dropout during the training process. In contrast, in LIRA the fusion of two sensor-specific policies is trained on a different novice policy, similarly to the method introduced in [14]. This architecture gives the teacher more control over which sensor-modality is essential in different situations. However, in contrast to previous works, LIRA actively calls for the human teacher's attention in case of uncertainty and fusion ambiguity and infers the right sensor indirectly from the human feedback in the action space. This avoids averaging between conflicting sensor policies with eventual dangerous results, see Sec. IV.

In [15], the use of heuristics is employed in the fusion operation. Since the camera-based network can determine mid/long-term actions, while the LiDAR-based networks are better equipped for close-range obstacle avoidance, the fusion discriminator prioritizes the LiDAR when its measured distance is lower than a threshold. The disadvantage of this approach, aside from the difficulties in scaling to more sensors, is that the camera is disregarded when the LiDAR sensor is active: in ambiguous situations this could result in dangerous decisions. As a consequence, LIRA avoids the use of heuristics and prefers to query the user in uncertain situations.

Since LIRA estimates the epistemic uncertainties with Gaussian Processes, the following section reviews their fundamental mathematical background.

A. Gaussian Process for Robot Learning

A Gaussian Process is a non-parametric regression method that provides the means for inferring prediction and epistemic uncertainty with a clear mathematical formulation [16]. The two equations that govern the mean and the variance of the process are

$$\mu(\boldsymbol{x}) = \boldsymbol{k}_{*}(\boldsymbol{\xi}, \boldsymbol{x})^{\top}(\boldsymbol{K}(\boldsymbol{\xi}, \boldsymbol{\xi}) + \sigma_n^2 \boldsymbol{I})^{-1}\boldsymbol{\Gamma}, \tag{1}$$

$$\Sigma = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_{*}^{\top}(\boldsymbol{K} + \sigma_n^2 \boldsymbol{I})^{-1}\boldsymbol{k}_{*}, \tag{2}$$

where k is the kernel evaluation at the current robot state $\boldsymbol{x}$, $\boldsymbol{k}_*$ is the covariance between $\boldsymbol{x}$ and the training state inputs $\boldsymbol{\xi}$, $\boldsymbol{K}$ is the covariance matrix of the training inputs, σ_n^2 is the process noise, and $\boldsymbol{\Gamma}$ denotes the collection of the desired actions during the demonstration. The covariance matrices are a function of the chosen kernel and its hyper-parameters which are optimized in the initial fitting of the data. It is possible to explicitly compute the value of the maximum and minimum predicted variance, respectively, on a remote unvisited region, i.e., an independent process, and close to the demonstration samples, i.e., the process noise. Thanks to this property, the epistemic variance threshold that discriminates certain against uncertain regions is chosen as a percentage of the maximum variance minus the minimum variance. This would automatically adjust the variance threshold in different applications and does not require any manual tuning. This requirement is necessary for an end-user accessibility without any knowledge in robot learning. Differently to HG-DAgger [3] where the safety threshold of the estimated epistemic uncertainty of the Neural Network Ensemble is tuned based on the human interventions in the training phase, for a Gaussian Process, we tuned the percentage threshold and update the variance threshold according to the optimized kernel parameters. This summarizes with

$$\frac{\Sigma_{\mathrm{tr}} - \Sigma_{\min}}{\Sigma_{\max} - \Sigma_{\min}} = \text{constant}$$

that is solved for Σ_{tr}. Therefore LIRA does not need the distinction between a training and test phase like HG-DAgger. The next section will explain details of the framework and how it provides a valuable tool in the field of interactive imitation learning.

III. Framework: LIRA

For ground robot navigation, two inputs need to be directly or indirectly controlled: the wheels velocity and the steering angle. However, for making a robot autonomous, the input control needs to be a function of the perceived sensor inputs. Nevertheless, when the controller is learned, it is of fundamental importance to measure the confidence of its decisions: unconfident situations should result in the active querying of the supervisor, i.e., another policy or the human, without attempting possible unsafe behaviours.

The presented algorithm, named LIRA, enhances this idea and it also allows the querying of multiple single sensor policies when a novel sensor-fused policy is uncertain. Each

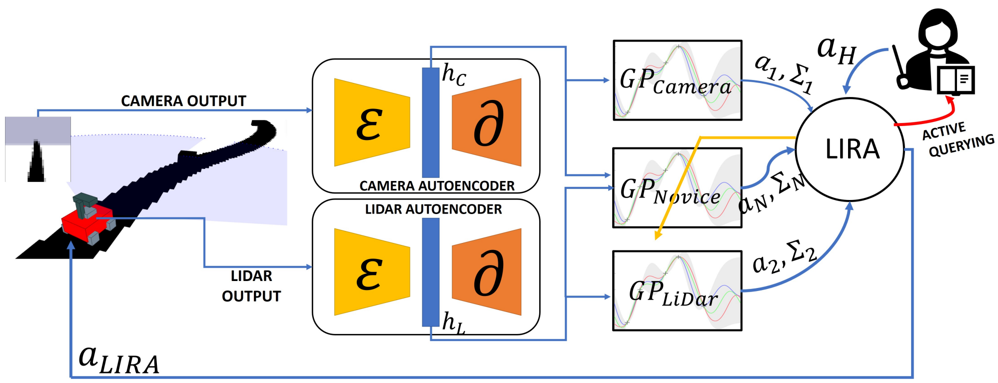


Fig. 2: LIRA takes as input the action and the confidence of the experts (Camera and LiDAR), the fusion policy and the user inputs. It performs the disambiguation based on the user corrections and teaches the novice on how to act in future similar situations.

control variable, i.e., steering velocity and wheel velocity is modelled as an independent Gaussian Process. Fig. 2 illustrates how the human, the novice policy, and the single-sensor policies integrate in the learning loop and how their desired action and confidence is used by LIRA for inferring the current decision. Its objective is to infer the best action to take and keep updating the novice sensor-fused policy, see Alg. 1. Additionally, the high dimensional input of the sensor is encoded in a latent variable using a Deep AutoEncoder trained on the database of each single-sensor expert. Furthermore, the lengthscales in each of the latent dimension for the fitting of the GP are optimized with an Automatic Relevance Determination algorithm.

As LIRA is conceptualized for the fusion of multiple sensor policies, LIRA requires at least two expert behaviours, Π_1 and Π_2, that use different sensors. The nature of these policies does not matter: they can be hard coded or data-driven. The only requirement is that they provide their desired robot action a and a measure of the confidence Σ on the decision. This information is required in l. 12 of Alg. 1 when the fusion policy is not confident and LIRA has to query the single-sensor policies. Details on how LIRA performs the sensor fusion, solving the ambiguity is detailed in Sect. III-B.

When eventually the human takes control of the robot, because they were queried by the algorithm or because they would like to modify the observed robot behaviour, LIRA gives them full control of the robot (l. 6, similar to [3]) but at the same time it learns from the corrections in a data-efficient manner (l. 7, see Sec. III-A). In future encounters of the same situation the sensor-fused policy will know how to act and will remember the user correction in that circumstance. Details on how LIRA uses Gaussian Process Regression for active and interactive learning of the final policy are summarized in the following section.

A. *Interactive Learning with Gaussian Process*

LIRA exploits the use of epistemic uncertainty for understanding whether the combined sensor reading is novel for the novice policy or not. However, since the user can take

Algorithm 1: LIRA

1: **Get input:**
2: $[a_{1,\dots,K}, \Sigma_{1,\dots,K}] = \Pi_{1,\dots,K}(\mathcal{O})$
3: $[a_N, \Sigma_N] = \Pi_N(\mathcal{O})$
4: a_H = HumanInput()
5: **if** Human in Control **then**
6: $\quad a = a^H$
7: $\quad$ **Update** Π_N
8: **else**
9: $\quad$ **if** $\Sigma_N < \Sigma \mathrm{tr}$ **then**
10: $\quad\quad a = a_N$
11: $\quad$ **else**
12: $\quad\quad$ [a,Ask]=**ResolveAmbiguty**($[a, \Sigma]_{1,\dots,K}$, **input**)
13: $\quad\quad$ **Update** Π_N
14: $\quad$ **end if**
15: **end if**
16: **Execute** a

control of the robot even when the novice policy is confident (l. 5 of Alg. 1) it is crucial to determine how to modify the existing database according to the user corrections rather than keep aggregating it with the database.

It is reasonable to leveraged again the use of epistemic uncertainty as a discriminator for either *adding samples* to the database or *correct* already existing ones. Thus, when $\Sigma \geq \mathrm{tr}$ the *Update* rule (l. 12 of Alg. 1) of the policy simply aggregates the sample to its database. Otherwise, the existing database is modified according to:

$$\boldsymbol{y}^{\mathrm{new}} = \boldsymbol{y} + \boldsymbol{k}_{*}\left(\boldsymbol{x}, \hat{\boldsymbol{x}}\right)\left(a - \mu\left(\hat{x}\right)\right), \tag{3}$$

where a is the output of the desired control action labelled from the human control (l. 6); $\boldsymbol{k}_{*}(\boldsymbol{x}, \hat{\boldsymbol{x}})$ automatically modulates how much the correlated elements in the database should be modified for matching the user's desired corrections. This way of spreading corrections on the database is an easy solution to the possible collection of conflicting labels that would generate poor behaviour of the policy and proved to also be effective in rapidly adjusting mistaken labels provided in the previous policy roll-outs without the accumulation of them as noise in the database.

B. Ambiguity Resolution

When the robot encounters a novel situation for the novice policy, and the user is not in control, LIRA queries the expert policies on their desired actions. If only one policy is significantly more confident than others, LIRA simply solves the ambiguity by giving priority to it, updating the novice policy with the selected confident action. If multiple policies are confident and they do not agree on the control action, LIRA slows down the robot and solves the ambiguity by asking the user to take control. The algorithm goes back to l. 6-7 and the novice policy is updated for that fused sensor input. This feedback modality is named *local disambiguation*.

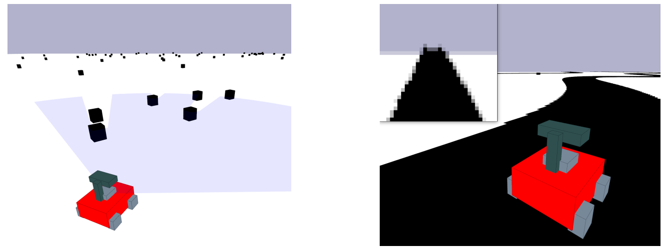
For the use of *long-term disambiguation*, LIRA allows to give priority to the sensor policy that has actions similar to the human correction. This option captures the idea that in some circumstances only one sensor perceives the relevant feature. For example, in the proximity of an obstacle the user wants to rely more on the LiDAR or Camera policy according to their perceived features, remember Fig. 1. This expert prioritization is activated when a specific button is kept pressed (see Fig. 3), the selected expert policy remains confident, and the novice policy is still *not* confident. When one of these conditions is not respected anymore, LIRA would restart the disambiguation procedure with a possible query of the user when a novel uncertain and/or ambiguous situation is encountered again. In case that explicit actions are provided in this modality, the expert selection is recomputed. This button's availability gives the user the complete freedom to choose between local demonstrations or long-term expert selection.

Fig. 3: Teleopertion device for giving continuous action feedback. For safety reasons, the user can give corrections only if the blue button is kept pressed. The green one is used for activating the long-term disambiguation as described in Sec. III-B.

IV. Experimental Validation

The goal of the experiments is to compare the performance of LIRA with:

- A simplified version that queries the user in case of uncertainty but it is not meant for the fusion of expert policies. It is equivalent to learning the novice policy from scratch only from user interactive corrections, similarly to [3];
- a non-ambiguity-aware version that always averages the experts also when they disagree. The human has the responsibility to take control and correct the policy when the robot attempts dangerous actions. This shows that the user can correct existing undesired actions.



(a) LiDAR Policy Training (b) Camera Policy Training

Fig. 4: Collection of demonstrations of the single sensor policy for training the AutoEncoders and the Gaussian Process with Behavioural Cloning.

The comparison will focus on the successfulness of the learning and the direct number of provided user labels. In all the experiments, the single sensor experts are trained using Behavioural Cloning in different environments. The demonstration data were used for training an autoencoder for learning a latent representation of the high-dimensional sensor input. This dimensionality reduction can be seen as deep unsupervised kernel learning [17]. Then, the latent state and the record actions are used for fitting a Gaussian Process where the hyper-parameters of the Radial Basis Functions (RBF) kernel were optimized with an Expectancy Maximization algorithm. Automatic Relevance Determination (ARD) is employed for selecting different lengthscales in each dimension of the latent space. The same encoders were also used for obtaining the concatenated latent representation input of the sensor-fused policy. Moreover, the optimized hyper-parameters of the kernel were re-used for the initialization of the novice policy. This choice is reasonable because the correlation of latent inputs can be considered invariant.

The player stage environment is used for simulating the ROSbot, also employed in the physical validation experiment. The Robot Operating System (ROS) and Tensorflow with GPFlow libraries are used for the code implementation. The robot is equipped with a 2D-LiDAR and a 32x32 pixel RGB camera (later converted to gray-scale), as visualized in Figure 4b. The 2D-LiDAR, placed at a height of $26\,\mathrm{cm}$, produces 130 distance measurements, evenly spaced in a forward-facing angle of 130 degrees. The control variables are the linear velocity (maximum set to $0.20\,\mathrm{m/s}$), and the steering angular velocity (maximum set to $210\,{}^{\circ}/\mathrm{s}$). The control frequency of the robot is $5\,\mathrm{Hz}$.

*A. **Simulation**: Line-Following with Collision Avoidance*

The LiDAR policy (LidExpert) is trained in an obstacle forest to drive straight or steer away from the obstacle while

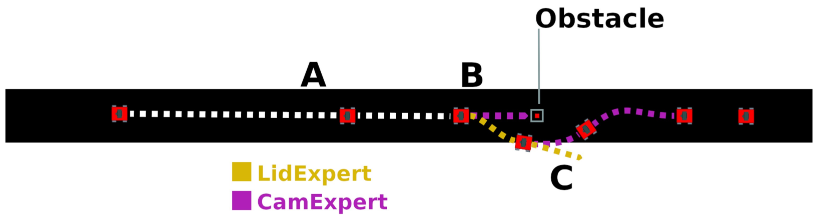


Fig. 5: Training of the Novice Policy with LIRA: A) The novice is confident to drive straight. B) The robot encounters an obstacle: the novice is uncertain and the CamExpert disagrees with the LidExpert. LIRA queries the human that indirectly selects the LidExpert for a long-term disambiguation. C) The human gives new feedback to steer back to the black line, selecting the CamExpert.

Expert Training Labels	Simulation	Real-world
LidExpert	350	406
CamExpert	455	378

TABLE I: The CamExpert and LidExpert required a similar amount of labels to learn how to navigate in their training environment.

the camera policy (CamExpert) is trained to follow a black line, see Fig. 4. The first goal was to show that the proposed architecture with autoencoder for dimensional reduction and GP for policy learning would successfully learn to perform the navigation task in single sensor configurations. Both the learning policies were successful and data efficient, See Table I.

For the testing of LIRA, the robot is placed in an environment with a 120 m long black line. However, 21 black obstacles are placed along the line, as visualized in Fig. 5. The environment's design is such that the camera cannot distinguish the black obstacles from the black line while the LiDAR can only perceive the obstacles. An ablation study is conducted repeating the experiments 5 times for each version of the algorithm.

The first validation was on the feasibility of learning a fusing policy from scratch without the use of single-sensor expert policies. It results in successful learning but with the necessity of many user inputs. The second validation consists of always averaging the expert actions (even in conflicting situations). Because this could result in dangerous teaching of the novice, the goal was to test if the update rule of Eq. (3) was reactive enough to correct the policy when the user takes control (alerted by the robot taking the wrong decision). Finally, the full LIRA version was used for checking the successful learning but with a reduced number of required labels and actively querying the user in case of ambiguity. The box plot of Fig. 6 shows that LIRA requires less direct human labels (thanks to the long-term disambiguation modality). Moreover, Table II also shows that LIRA learns a policy that deviates the minimum from the center of the road for performing the collision avoidance. This can be read as an index of higher safety.

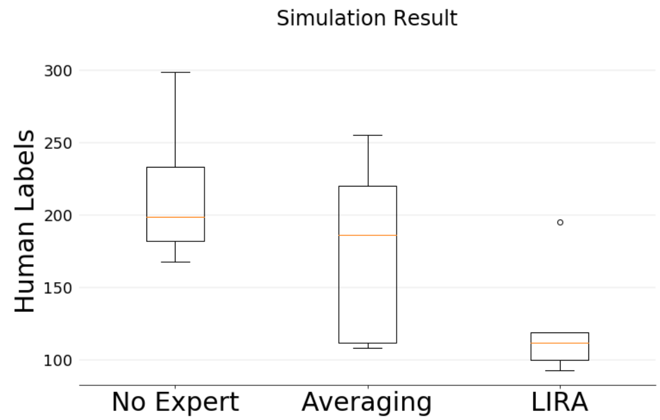


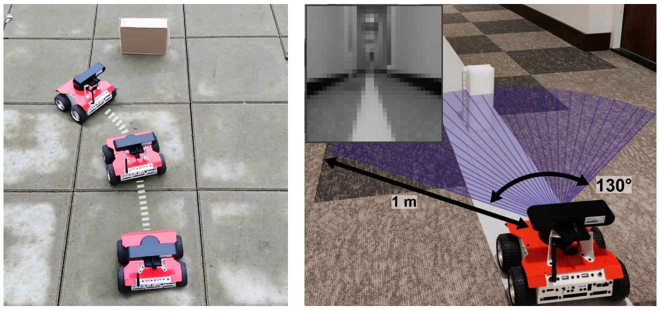
Fig. 6: Comparison of the human label required for accomplish a successful (simulated) navigation task when the novice policy is learned only from human corrections, similarly to HG-DAgger [3] (No Expert), when the expert policy are used but simply averaged (Averaging), and the LIRA algorithm.

Distance to the Center [m]	Min	Mean	Max
No Expert	0.23	0.29	0.33
Expert Averaging	0.22	0.28	0.43
LIRA	0.18	0.23	0.28

TABLE II: Root mean square of the distance to the centerline is computed for measuring the performance of the policy.

B. ***Real-word Validation****: Corridor navigation*

The real-world validation is similar to the simulation experiment, with a white line in the corridor's middle. Three white obstacles are positioned on the white line, making it hard to distinguish based on the camera sensor. The CamExpert is trained to follow the white line in a corridor without obstacles. The LidExpert is trained to avoid obstacles in an open space, see Fig. 7. With LIRA, we successfully trained a policy that avoids the obstacles and follows the white line. The novice required 344 labels from the human and 64 from the experts. It is worth noting that the quality of the expert's policies determines how much LIRA benefits in the training process in terms of performance and data efficiency. Compared to the simulation, the complexity of the experiment either requires better experts or more human



(a) LiDAR Policy Training (b) LiDAR and Camera perspective when facing a white obstacle

Fig. 7: Different set-up for the training of the LidExpert (a), the CamExpert without obstalces (b), and Novice with obstacles (b).

input for the fusion task.

V. Conclusions and Future Work

This study investigates the problem of sensor policy fusion where the confidence of the novice policy and the disagreement between the single sensor policies is used for actively querying the help of the user demonstrator. We showed that the use of interactive correction helps in preventing the fusion of conflicting labels avoiding to learn dangerous navigation behaviours. The expert policies used during the experiments are trained using Behavioural Cloning in a combination of Deep AutoEncoder for kernel learning and a Gaussian Process for the modelling of desired control action and policy uncertainty. As this expert training is done offline, it assumes that the collected database does not contain conflicting labels. Following investigations will implement the possibility of detecting this conflicting situation with a measure of heteroscedastic noise of the process [18]. This other measure of data-uncertainty could additionally help in calling the demonstrator, avoiding dangerous action selection in robot control and sensor fusion. However, we showed that having a measure of epistemic uncertainty and a well designed update rule is already sufficient to avoid collecting conflicting labels in the novice policy. It is worth noting that the update rule avoids the exponential growth of the policy database: data efficiency results in computational efficiency without the necessity to use approximation methods.

Because of the successful learning of ground navigation tasks from high dimensional inputs, immediate future implementations will focus on drone collision avoidance [19] and force tasks in robot manipulation [20] when the robot state will be fused with a camera (or LiDAR) input. The online and interactive learning of temporal features [21] for modelling the observation dynamics will be tested combined with the proposed GP framework. Finally, future investigations will involve non-skilled human teachers to study the usability of active and interactive learning in daily life tasks, filling the current gap between research and daily life applications.

References

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[2] D. Silver, J. A. Bagnell, and A. Stentz, "High performance outdoor navigation from overhead data using imitation learning," in *Robotics: Science and Systems*, 2009, pp. 262–269.

[3] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer, "HG-DAgger: Interactive imitation learning with human experts," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8077–8083.

[4] C. Celemin and J. Ruiz-del Solar, "An interactive framework for learning continuous actions policies based on corrective feedback," *Journal of Intelligent & Robotic Systems*, vol. 95, pp. 77–97, 2019.

[5] W. Elmenreich, "An introduction to sensor fusion," Austria: Vienna University Of Technology, Tech. Rep., 2002.

[6] G. Franzese, C. E. Celemin, and J. Kober, "Learning interactively to resolve ambiguity in reference frame selection," in *Conference on Robot Learning (CoRL)*, 2020.

[7] A. Giusti, J. Guzzi, D. C. Ciresan, F. L. He, J. P. Rodriguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. D. Caro, D. Scaramuzza, and L. M. Gambardella, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661–667, 2016.

[8] S. Ross, N. Melik-Barkhudarov, K. Shaurya Shankar, A. Wendel, D. Dey, J. A. Bagnell, and M. Hebert, "Learning monocular reactive uav control in cluttered natural environments," in *IEEE International Conference on Robotics and Automation*, 2013, pp. 1765–1772.

[9] Z. Huang, C. Lv, Y. Xing, and J. Wu, "Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding," *IEEE Sensors Journal*, vol. 21, no. 10, pp. 11 781–11 790, 2021.

[10] I. Kang, R. Cimurs, J. H. Lee, and I. Hong Suh, "Fusion drive: End-to-end multi modal sensor fusion for guided low-cost autonomous vehicle," in *17th International Conference on Ubiquitous Robots (UR)*, 2020, pp. 421–428.

[11] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1527–1533.

[12] F. Codevilla, M. Miiller, A. Lopez, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4693–4700.

[13] G.-H. Liu, A. Siravuru, S. Prabhakar, M. Veloso, and G. Kantor, "Learning end-to-end multimodal sensor policies for autonomous navigation," in *Conference on Robot Learning (CoRL)*, 2017.

[14] I. L. Davis and A. Stentz, "Sensor fusion for autonomous outdoor navigation using neural networks," *IEEE International Conference on Intelligent Robots and Systems*, vol. 3, pp. 338–343, 1995.

[15] X. Zhou, Y. Gao, and L. Guan, "Towards goal-directed navigation through combining learning based global and local planners," *Sensors (Switzerland)*, vol. 19, no. 1, 2019.

[16] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[17] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep kernel learning," in *Artificial Intelligence and Statistics*, 2016, pp. 370–378.

[18] Q. V. Le, A. J. Smola, and S. Canu, "Heteroscedastic gaussian process regression," in *22nd International Conference on Machine Learning (ICML)*, 2005, pp. 489–496.

[19] J. Lin, H. Zhu, and J. Alonso-Mora, "Robust vision-based obstacle avoidance for micro aerial vehicles in dynamic environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2682–2688.

[20] G. Franzese, A. Mészáros, L. Peternel, and J. Kober, "ILoSA: Interactive learning of stiffness and attractors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[21] R. Perez-Dattari, C. Celemin, G. Franzese, J. Ruiz-del Solar, and J. Kober, "Interactive learning of temporal features for control: Shaping policies and state representations from human feedback," *IEEE Robotics & Automation Magazine*, vol. 27, no. 2, pp. 46–54, 2020.