Incremental Learning through Probabilistic Behavior Prediction

Sheida Nozari[1,2], Ali Krayani[1], Lucio Marcenaro[1], David Martin[2] and Carlo Regazzoni[1]
Department of Engineering and Naval architecture (DITEN), University of Genoa, Italy[1]
Intelligent systems lab, University Carlos III de Madrid, Spain[2]
emails: {sheida.nozari, ali.krayani}@edu.unige.it, {lucio.marcenaro, carlo.regazzoni}@unige.it, dmgomez@ing.uc3m.es

***Abstract*—Learning from expert demonstrations effectively reduces the number of interactions required to train a policy between the learning agent and its environment. Where agent-environment interactions can be costly, reinforcement learning is critical, and imitation learning suffers significantly from learning hierarchical policies when the imitative agent encounters an unobserved state by the expert agent. We propose a probabilistic incremental imitation learning method that employs a Dynamic Bayesian Network to encode observed teaching-agent's behaviors. The presented model grows and matures based on imitation loss formulation at a discrete level in the online learning procedure. The learning agent is trained by using long-term predictions from the generative learning model to replicate the teacher's motion while learning how to choose an appropriate action through new experiences. Our results affirm that a Dynamic Bayesian optimal approach provides a principled framework and outperforms conventional reinforcement learning methods.**

***Index Terms*—Imitation learning, Dynamic Bayesian network, performance analysis, autonomous tracking, action prediction**

I. Introduction

Imitation learning (IL) is a general method for rapidly acquiring new skills from an expert agent in order to accelerate policy learning when solving reinforcement learning (RL) problems [1]. While RL methods rely on uninformed random exploration to locally improve a policy, IL leverages prior knowledge about a problem in terms of learning a task by providing demonstrations performed by an expert. The goal of IL is to learn a policy that performs expert policy quickly. Since the expert policy may be sub-optimal for RL, performing IL is frequently used to provide an efficient warm start to the RL problem and thus reduce the number of interactions with the environment [2].

However, since IL is highly reliant on expert demonstrations, the learning agent might fail to reach the goal in an unseen environment. On the other hand, an agent needs to perform much exploration to learn proper behavior in RL. Additionally, RL can suffer from poor scalability, and it can be challenging to design a reward function that leads RL to the desired behavior [3]. These problems can be relieved by performing a policy sequence updated by mimicking the expert demonstrations to converge on average to the performance of the expert policy [4], [5]. Integration of both modalities was shown that this rate is sufficient to make IL more efficient than solving an RL problem from scratch [2].

Motivated by the above discussion, we propose a framework for autonomous tracking in a continuous environment that combines IL with RL. IL is used as a pre-training step to encode an expert demonstration in a Dynamic Bayesian Network (DBN) [6] that describes desired behaviors. The DBN is a probabilistic graphical model (PGM) that employs graph-based representation to encode various multidimensional random variables and represent causal relationships among them [7]. Recent studies have shown the utility of PGMs at factorizing causal relationships between latent states of multisensory data and encoding semantic relationships between random variables [8]. Moreover, the work in [9] have demonstrated how to make inferences on PGMs in a data-driven way. Usually, those works use stochastic models that follow a Markovian assumption where an event's probability depends on the state previously attained. Due to its hierarchical nature, DBN can express temporal relationships among high-level variables capturing abstract semantic information about the environment and low-level distributions capturing rough sensory information with their respective evolution through time. This work employs the discrete information of an expert probabilistic model enabling the learning-agent to improve its actions by minimizing the imitation cost that allows avoiding abnormal states in the future.

II. Proposed framework

This section includes two main phases, offline learning and online learning. In the former, we learn an Expert model ($\mathring{E}$) encoding dynamical behaviors of a Teacher (T) moving to a fixed goal (G). In the latter, an incremental IL model is learned where a Learner (L) attempts to learn sub-optimal behavior by observing T demonstrations and updates its knowledge while transiting in a continuous environment to reach G.

A. Offline learning phase

The offline learning process aims to learn $\mathring{E}$ based on T behavior which can be used as a reference model by L. $\mathring{E}$ consists of a DBN representing the T dynamics in the environment and modelling hierarchical relationships between different variables with their respective evolution through time. At each time instant t (DBN's *slice*), causal relationships between variables are encoded through *inter-slice links*. While causal relationships between variables in subsequent time instances are encoded through *temporal links* (see Fig. 1). The DBN model consists of three levels, the bottom level depicts the T's observations represented by Z_t^T. The middle

level illustrates the *generalized states* (GSs) expressed as $\tilde{\boldsymbol{X}} = \{\tilde{X}_t\}_{t=1,\dots,t}$, where $\tilde{X}_t = [X_t,\ \dot{X}_t]^\intercal$, $\dot{X}_t \sim \frac{X_t - X_{t-1}}{\Delta t}$ and Δt is the sampling time. The top level represents the T's discrete states explaining the dynamical transition behaviors reflected into a semantic discrete space. The observation model

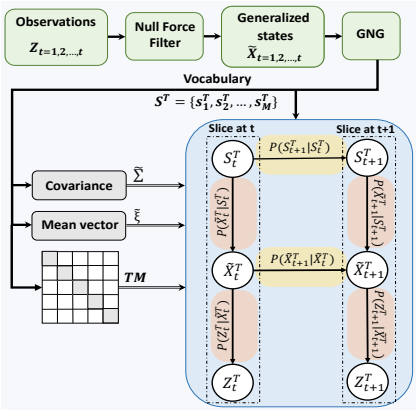


Fig. 1: Proposed DBN structure. Inter-slice links are depicted in orange and temporal-links are colored in yellow.

that maps $\tilde{\mathrm{X}}_t^T$ to Z_t^T is defined as:

$$\mathrm{Z}_t^T = \mathrm{H}\tilde{\mathrm{X}}_t^T + v_t, \tag{1}$$

where $H{=}[\mathrm{I}_d\ \ 0_{d,d}]$ is the observation matrix that parametrize the observation model and v_t is the Gaussian measurement noise, such that, $v_t \sim \mathcal{N}(0, \mathrm{R})$. We assume that the dynamics of GSs evolve according to a static equilibrium assumption described as:

$$\tilde{\mathrm{X}}_t^T = \mathrm{A}\tilde{\mathrm{X}}_{t-1}^T + w_t, \tag{2}$$

where $\mathrm{A} \in \mathbb{R}^{d\times d}$ is the dynamic matrix and w_t is the process noise, such that $w_t \sim \mathcal{N}(0, \mathrm{Q})$. This implies a Null Force Filter (NFF) that can be interpreted as an unmotivated Kalman Filter, it uses the innovations obtained by observing a sequence Z_t^T to estimate the next state that describes the agent's motion in the GS space. The innovations can be seen as mismatches between observations and predictions as:

$$\dot{\boldsymbol{\upsilon}} = \mathrm{H}^{-1}\big(\mathrm{Z}_t^T - \mathrm{H}\tilde{\mathrm{X}}_t^T\big), \tag{3}$$

The couples $(\tilde{\mathrm{X}}^T, \dot{\upsilon})$ are defined as generalized errors (GEs) that can be clustered in an unsupervised manner using the Growing Neural Gas with utility measurement (GNG-U) [10]. GNG-U outputs a set $\mathbf{S}^T$ of discrete variables (i.e., clusters) representing the discrete level of the DBN structure and forming the so called Vocabulary. Each $\mathrm{s}_m \in \mathbf{S}^T$ is assumed to follow a multivariate Gaussian distribution, such that $\mathrm{s}_m \sim \mathcal{N}(\tilde{\xi}_m,\ \tilde{\Sigma}_m)$, where $\tilde{\xi}_m = [\xi_m,\ \dot{\xi}_m]^\intercal$ is the GS centroid of the m-th cluster and Σ_m is its covariance matrix. The DBN discrete-level represents the activated cluster ($s_t \in \boldsymbol{S}^T$) at each time instant. Our work assumes that the learner uses the discrete information in $\boldsymbol{S}^T$ as a $\mathrm{flashback-memory}$ ($\mathring{\mathcal{M}}$) that guides the RL procedure. The probabilistic law that regulates transition among different local forces captured by different clusters can be estimated in different ways (e.g. frequentist or geometrical) and encoded in a Transition Matrix (TM) that estimates the transition probabilities $\mathrm{P}(\mathrm{s}_{t+1}|\mathrm{s}_t)$.

B. Online learning phase

In this phase, we propose an online incremental IL model (OIL) that allows L to learn how to improve the actions and reach G by minimizing imitation loss. $\mathring{\mathcal{M}}$ can be used during the online learning phase providing suitable predictions and learning policies to teach the best set of actions ($\mathcal{A}$) that L needs to accomplish its task. Therefore, $\mathring{\mathcal{M}}$ leads the active states during new experiences that describes how the agent can act in the environment to change sensory signals in order to match internal predictions of $\mathring{E}$ and thus to imitate efficiently by decreasing the abnormalities. This work suggests training L through the inclusion of $\mathring{\mathcal{M}}$ representing T's behaviors, which postulate that L optimizes its motions based on $\mathring{\mathcal{M}}$'s predictions over time. RL can formalize the underlying decision-making as an MDP, a model of a discrete-time process wherein an agent's actions may stochastically influence its environment. The proposed approach endows L with the capability of estimating the imitation cost, whereby minimizing the imitation cost (i.e., maximizing rewards in RL) ensures an equilibrium between L and its surrounding. Accordingly, lets define L's state and action at a given time instant t as s_t^L, a_t^L, respectively. We hypothesize that the L uses a probabilistic discrete representation $\boldsymbol{S}^T$ that encodes relevant information about the observed T's behaviors (i.e., $\mathring{\mathcal{M}}$) instead of exploiting explicitly from T (which rejects the idea of a buffer that replays previously observed T's states $\tilde{\boldsymbol{X}}^T$). We aim at, *i)* modeling a dynamic multiple reward function by considering the abnormalities between L and $\mathring{\mathcal{M}}$ at each t to *ii)* regulate the L's actions in the online IL stage.

1) Imitation cost: Two policies are considered, the action-based and the state-based to evaluate L's loss using the activated cluster ($\mathring{\mathrm{s}}$) from $\mathring{\mathcal{M}}$, where $\mathring{\mathrm{s}}$ is the closest $\mathring{E}$'s cluster to the current L's state (s_t^L) calculated by Euclidean distance. ***Action-based policy.*** Minimizing the divergence between the current L's action and the mean action of the activated cluster ($\mathring{s}$) such that:

$$\dot{P}_t = d_{\mathcal{M}}\big(g_a(\mathring{s}_t),\ a_t^L\big), \tag{4}$$

where $d_{\mathcal{M}}(\mathtt{X}, \mathtt{x})$ is the Mahalanobis distance between a distribution $\mathtt{X}$ and a point $\mathtt{x}$, $g_a(\cdot)$ is a function that extracts the action-distribution from a GS-distribution, such that $g_a(\mathring{s}_t) \sim \mathcal{N}(\tilde{\xi}_t,\ \tilde{\Sigma}_t^a)$ and Σ_t^a is the action's covariance information. $\mathring{s}_t \sim \mathcal{N}(\tilde{\xi}_t,\ \tilde{\Sigma}_t^a)$, which can be obtained according to:

$$\mathring{s}_t = \underset{s_m}{\operatorname{argmin}} \|s_t - \tilde{\xi}_m\|_2. \tag{5}$$

State-based policy. Minimizing the distinction between the current L's state (s_t^L) (produced by the last action a_{t-1}^L) and the predicted state from the activated cluster ($\mathring{s}_{t|t-1}$). The discrete probability $p(s_t|s_{t-1})$ from $\mathring{\mathcal{M}}$ is employed to estimate $\mathring{s}_{t|t-1}$. The term $\mathring{s}_{t-1}$, required in $p(\mathring{s}_t|\mathring{s}_{t-1})$, is calculated based on (5). The state-based policy can be written as:

$$\ddot{P}_t = d_{\mathcal{M}}\big(g_s(\mathring{s}_{t|t-1}),\ s_t^L\big), \tag{6}$$

where $g_s(\cdot)$ is a function that extracts the state-distribution from a GS-distribution, such that $g_s(\mathring{s}) \sim \mathcal{N}(\tilde{\xi}_t,\ \tilde{\Sigma}^s_t)$. Σ^s_t is the state's covariance information. The policies indicate the imitation loss regarding the a^L and s^L at each t in a continuous range $[0, 1]$ that describes the abnormality value. Hence, by minimizing the global imitation loss ($\hat{\mathcal{R}}_t$), the learning model can maximize the learning rate and gain a high reward. $\hat{\mathcal{R}}_t$ takes into account the mean value of both policies as:

$$\hat{\mathcal{R}}_t = \mathbb{E}(\dot{P}_t, \ddot{P}_t), \tag{7}$$

2) Action's magnitude: At each t, the a^L_t consists of a unit vector displacement (among a discrete set of unit vectors $\boldsymbol{\mathcal{A}}$) multiplied by a constant Ψ, which is calculated as the norm of ($\mathring{s}_t$) such as $a^L_t = \Psi\mathcal{A}$, where $\mathcal{A} = \{a_1, a_2, \ldots, a_8\}$ is a set of eight cardinal and ordinal directions used by L and $\Psi = \|\mathring{s}_t\|$. The selection of $a^L_t \in \boldsymbol{\mathcal{A}}$ is based on ϵ-greedy $\in [0, 1]$, which decays over the episodes. In case of exploration, a^L_t tends to be selected at random to explore more new positions that can be exploited in the future. In exploiting, actions are selected as follow:

$$a^L_t = \underset{a}{\operatorname{argmax}}(Q(s_t, a)). \tag{8}$$

L records the experienced states (s^+_t) along with the performed actions ($a_t \in \mathcal{A}$) in a incremental function $Q(s, a)$ and the new states are saved in set S_Q that grows incrementally as experiences are observed over time:

$$\mathrm{Q} = \begin{bmatrix} \mathrm{P}(a^{\mathrm{L}}_1|\mathrm{s}^{\mathrm{L}}_1) \ldots \mathrm{P}(a^{\mathrm{L}}_1|\mathrm{s}^+) & \ldots \\ \mathrm{P}(a^{\mathrm{L}}_2|\mathrm{s}^{\mathrm{L}}_1) \ldots \mathrm{P}(a^{\mathrm{L}}_2|\mathrm{s}^+) & \ldots \\ \vdots \quad \ddots \quad \vdots & \vdots \\ \mathrm{P}(a^{\mathrm{L}}_{\mathrm{N}}|\mathrm{s}^{\mathrm{L}}_1) \ldots \mathrm{P}(a^{\mathrm{E}}_{\mathrm{N}}|\mathrm{s}^+) & \ldots \end{bmatrix}, \tag{9}$$

where $\sum_{n=1}^{\mathrm{N}=8} \mathrm{P}(a^{\mathrm{L}}_n|\mathrm{s}^+_m) = 1$ such that s^+_m are the new explored states. In order to weigh up the trained model than $\mathring{E}$, L clusters all the recorded pairs $[\mathrm{s}^+_t, a_t]$ by employing GNG. The latter outputs a set of clusters representing the new states ($\hat{\mathrm{S}}$) and the corresponded mean actions ($\dot{\mathrm{a}}$) which are added to the updated Q-table (Q^*) defined as:

$$\mathrm{Q}^* = \begin{bmatrix} \mathrm{P}(\dot{\mathrm{a}}_1|\hat{\mathrm{S}}_1) & (\dot{\mathrm{a}}_1|\hat{\mathrm{S}}_2) \ldots \mathrm{P}(\dot{\mathrm{a}}_1|\hat{\mathrm{S}}_M) \\ \mathrm{P}(\dot{\mathrm{a}}_2|\hat{\mathrm{S}}_1) & (\dot{\mathrm{a}}_2|\hat{\mathrm{S}}_2) \ldots \mathrm{P}(\dot{\mathrm{a}}_2|\hat{\mathrm{S}}_M) \\ \vdots & \vdots \quad \ddots \quad \vdots \\ \mathrm{P}(\dot{\mathrm{a}}_N|\hat{\mathrm{S}}_1) & (\dot{\mathrm{a}}_N|\hat{\mathrm{S}}_2) \ldots \mathrm{P}(\dot{\mathrm{a}}_N|\hat{\mathrm{S}}_M) \end{bmatrix}. \tag{10}$$

L adapts the action selection procedure by updating the Q-table defined in (9) based on the imitation cost policies at each t. Since the provided Q is a probabilistic table, updating Q value can be rewritten in a probabilistic form as follows:

$$\mathrm{Q} = (1-\eta)\mathrm{P}(a_{t-1}|\mathrm{s}_{t-1}) + \eta\left[(1-\hat{\mathcal{R}}_t) + \gamma \max_{a_t} \mathrm{P}(a_t|s_t)\right], \tag{11}$$

where η is the learning rate which controls how quickly the learning agent adopts to the explorations imposed by the environment, $(1-\hat{\mathcal{R}}_t)$ is the normalized reward measurement with a range in $[0, 1]$, and γ is a discount factor.

III. EXPERIMENTAL EVALUATION

A. Experimental setup

The proposed framework is validated using a simulated dataset consisting of sensorial information collected by T where it attempts to reach G from different starting points. T moves based on the velocity field model proposed in [11], that $\vec{G}(r) = \left(\beta - \lambda e^{\frac{-r^2}{\psi}}\right)\hat{r}$, where r is the distance to G, $\lambda \leq \beta$ and $\hat{r}$ is a unit vector pointing at G. The T positional information and the corresponding velocities are obtained from the odometry module. Sensory data representing positional information from these experiments are used to learn the expert trajectories encoded in $\mathring{E}$ that L uses to imitate T.

B. Offline learning phase

This section shows the process of learning $\mathring{E}$ from T data. The NFF is used as an initial filter employed on the collocated data during tracking G. NFF outputs the GEs defined in (3) which can be clustered using GNG that outputs a set of discrete clusters representing the discrete regions of the trajectories generated by T. Fig. 2-(a)-(b)-(c)-(d) illustrate the clusters and the corresponding TM.

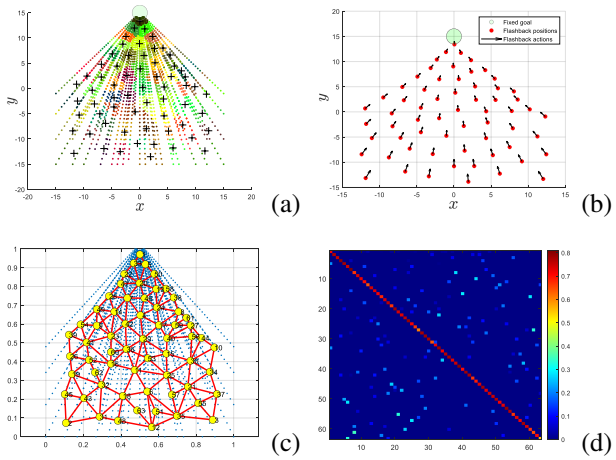


Fig. 2: Learning $\mathring{E}$. a) clustering of GEs, b) mean velocity of each cluster, c) clusters' relationship, and (d) TM.

C. Online learning phase

During the online phase, L modifies its actions based on the learned clusters during offline phase. Q-table records the L's observations and the corresponding actions as defined in (9). The experiments are done in a simulated environment. For having a fair comparative evaluation, all the experiments are considered with fixed steps. We run each algorithm over $4k$ episodes by different start positions to learn how to track G through learning the imitation policies. All experiments used the same *Stop condition*, which is met when: i) a minimum distance to target is accomplished (success) ii) a maximum time in the environment is reached (lost) or, iii) the agent goes out of boundary (outside). We evaluate the performance

of the proposed method and compare it with three learning algorithms, namely, the general cumulative reward-based Q-learning, inverse reinforcement learning (IRL), self learning (SL) in RL context (distance-based evaluation).

1) Performance evaluation: Action selection procedure has a big impact on the L's effort to reach the targeted G. A good policy requires fewer actions and in parallel less time to finish the mission. Fig. 3-(a) shows the number of taken actions by L for each episode using different methods. It describes the presented approach (OIL) makes less actions compared to other methods. This can be explained that evaluating L's movement using $\mathring{E}$ can improve the actions that lead the agent to the desired next state. L adopting the proposed method has higher successful trajectories than SL, Q-learning and IRL as depicted in Fig. 3-(b). Our approach uses a threshold ρ to initialize learning of new explored s^L in Q-table (see (9)). Since ρ has a great impact on the Q-function's complexity,

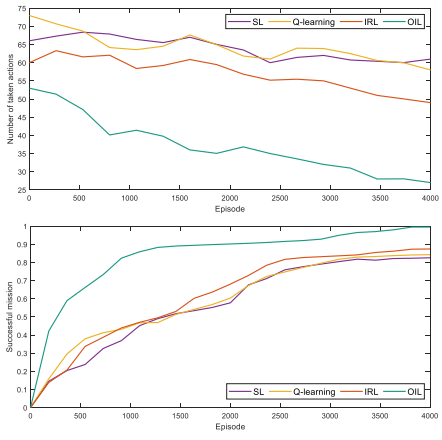


Fig. 3: Learner performance. a) The number of taken action by L, and b) The success rate to reach G.

we train L with different ρ values, depending on the distance between the current state s_t^L and the set of recorded states ($\mathbf{S}_Q$) in Q-table. By considering the success rate and the required execution time obtained by each ρ value, we select the suitable value as shown in Table I. $\rho = 1$ and $\rho = 3$ have almost the same success rate but the required time by $\rho = 3$ is more optimal than $\rho = 1$. Fig. 4 demonstrates how modifying the actions can reduce the exploration and minimize the imitation cost resulting in a high learning rate during the training phase.

TABLE I: Training the learning model with different ρ values. The selected threshold is $\rho = 3$.

ρ	1	1.5	2	2.5	3	3.5	4	4.5	5
success(%)	96.74	95.99	96.11	96.04	96.52	94.87	93.01	91.56	88.93
time(s)	110.49	93.71	102.02	99.89	90.24	97.83	100.04	109.33	114.26

2) Learning cost evaluation: Two main factors affect $\hat{\mathrm{R}}$ (see (7)), the action difference at time t (see (4)) and the state divergence after performing a_t by L (see (6)). Fig. 5 illustrates the imitation loss in both policy $\dot{P}$ where L is under control of action selection at each t, and policy $\ddot{P}$ which by

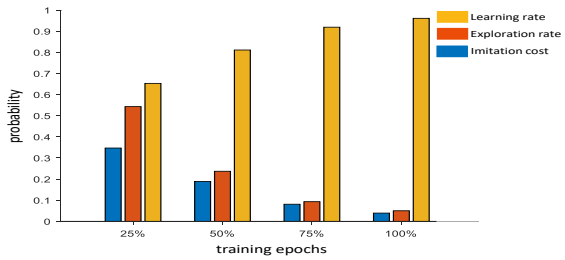


Fig. 4: Presenting the exploration and learning rates after each training quarter and their impact on the imitation cost.

improving a_t^L leads to minimizing the divergence between prediction and evidence. Further, Fig. 5 shows that $\hat{\mathrm{R}}$ drops down capably in less than 2k training episodes, and after 3k episodes, its value tends to stable below 0.1, reaching about 0.039. Therefore, L learns to maximize the likelihood with $\mathring{E}$. Fig. 6-(a)-(b) presents the performance of the proposed method

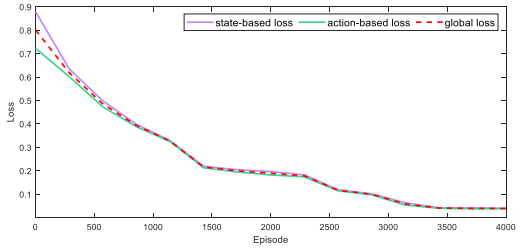


Fig. 5: Imitation loss measurement by three imitation policies.

during training and testing, respectively, in terms of success, lost, and going outside (explained in III-C). Also, Fig. 6-(a)-(b) provide a comparison with other learning methods. It is demonstrated that the proposed method (OIL) outperforms others in both training and testing stage, which is attributed to the effectiveness of motion prediction while dealing with abnormalities that improve the success rate. Additionally, during testing, results showed that by $4k$ training episodes, L could move in a continuous environment to effectively reach G whereas other methods still have a high failure rate. As discussed in II-B.2, L clusters all observed states

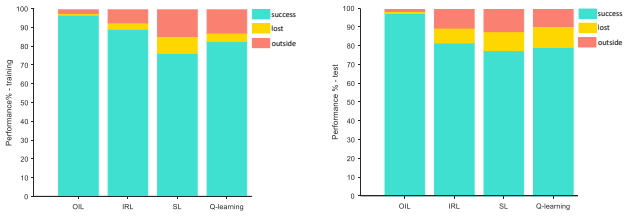
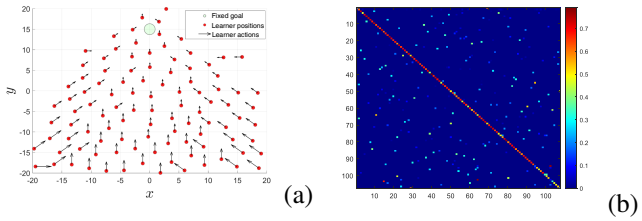


Fig. 6: Results after $4k$ training episodes. a) Training results, and b) Testing results. In both stages OIL has higher success than other methods.

and the corresponding performed actions. The recorded pairs are clustered for two reasons: to calculate the mean action

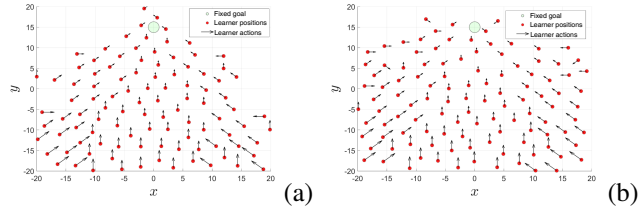
value of the corresponding clusters to have comparable data with $\mathring{E}$ and avoid looking in too many states in the Q-table. Fig. 7-(a)-(b) depict the Q* clusters (see (10)) and the corresponding TM. Comparing sub-figures (Fig. 2-(d) and



(a) (b)

Fig. 7: Discrete state-action representation from global imitation policy $\hat{\mathcal{R}}$ (a), and the corresponding TM (b).

Fig. 7-(b)) show that L has an expanded TM than $\mathring{E}$ after exploring and learning new states, allowing L to predict better and select desired actions. Under probabilistic inference, such an incremental learning process endows L with the capability of avoiding abnormal states. Meanwhile, to evaluate the effect of each imitation strategies ($\dot{P}$ and $\ddot{P}$), L is trained with each policy individually. Fig. 8-(a)-(b) demonstrate the learned clusters through each policy. Comparing Fig. 7-(a) with Fig. 8-(a)-(b) shows how applying both policies in parallel ($\hat{\mathcal{R}}$) generates the most efficient training. In testing stage, three



(a) (b)

Fig. 8: Discrete state-action representation from the action-based policy $\dot{P}$ (a), and the state-based policy $\ddot{P}$ (b).

Q*-tables obtained with $\hat{\mathcal{R}}$, $\dot{P}$ and $\ddot{P}$ are employed with ϵ-greedy $= 0$ to generate 300 trajectories (from new start positions) that are compared with the T's behavior. We use two distance measurements to compare trajectories from testing the provided Q*-tables with the T's behaviors: Spatio-Temporal Euclidean Distance (STED) [12] and Symmetrized Segment-Path Distance (SSPD) [13]. STED uses temporal information by comparing trajectories point to point. SSPD is a shape-based distance that compares trajectories as a whole. Table II shows the mean value of distance measurements between test trajectories (over 300 starting points) of different imitation policies and T's behaviors. Furthermore, Table II presents the quantitative results from testing the trained models by $\hat{\mathcal{R}}$, $\dot{P}$ and $\ddot{P}$ in terms of success, lost, and going outside (explained in III-C).

IV. Conclusion

We have proposed an incremental IL method where a learning-agent does not require to repeat the teaching-agent's behaviors explicitly. It learns by watching the teaching-agent and developing a probabilistic model of the critical aspects of its observations. Therefore, the learner is not limited to recalling exact observations of the behaviors but employs a probabilistic model as a $\mathring{\mathcal{M}}$ for guiding an RL method that allows the learning-agent to learn a previously observed task on its own. Future work concentrates on applying the GNG on the learning model during the online phase to update the transition matrix in real-time and improve predictive abilities at both discrete and continuous levels that enrich the learning-agent with the capability to explain abnormal situations and how they can be avoided in the future.

TABLE II: Testing results after $4k$ training episodes.

imitation policy	success (%)	lost (%)	outside (%)	STED	SSPD
trained by $\hat{\mathcal{R}}$	97.22	1.05	1.73	0.551	0.164
trained by $\dot{P}$	90.76	4.23	5.01	1.105	0.362
trained by $\ddot{P}$	89.01	4.96	6.03	1.359	0.411

References

[1] H. Le, N. Jiang, A. Agarwal, M. Dudik, Y. Yue, and H. Daumé III, "Hierarchical imitation and reinforcement learning," in *International conference on machine learning*. PMLR, 2018, pp. 2917–2926.

[2] C.-A. Cheng, X. Yan, N. Wagener, and B. Boots, "Fast policy learning through imitation and reinforcement," *arXiv preprint arXiv:1805.10413*, 2018.

[3] K. Judah, A. Fern, P. Tadepalli, and R. Goetschalckx, "Imitation learning with demonstrations and shaping rewards," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, 2014.

[4] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[5] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *ICML*, vol. 97. Citeseer, 1997, pp. 12–20.

[6] Z. Ghahramani, "Learning dynamic bayesian networks," in *International School on Neural Networks, Initiated by IIASS and EMFCSC*. Springer, 1997, pp. 168–197.

[7] L. E. Sucar, "Probabilistic graphical models," *Advances in Computer Vision and Pattern Recognition. London: Springer London. doi*, vol. 10, pp. 978–1, 2015.

[8] S. Benferhat, P. Leray, and K. Tabia, "Belief graphical models for uncertainty representation and reasoning," *A Guided Tour of Artificial Intelligence Research: Volume II: AI Algorithms*, pp. 209–246, 2020.

[9] M. Baydoun, D. Campo, V. Sanguineti, L. Marcenaro, A. Cavallaro, and C. Regazzoni, "Learning switching models for abnormality detection for autonomous driving," in *2018 21st International Conference on Information Fusion (FUSION)*, July 2018, pp. 2606–2613.

[10] H. Iqbal, D. Campo, M. Baydoun, L. Marcenaro, D. M. Gomez, and C. Regazzoni, "Clustering optimization for abnormality detection in semi-autonomous systems," in *1st International Workshop on Multi-modal Understanding and Learning for Embodied Applications*, 2019, pp. 33–41.

[11] D. Campo, A. Betancourt, L. Marcenaro, and C. Regazzoni, "Static force field representation of environments based on agents' nonlinear motions," *EURASIP Journal on Advances in Signal Processing*, vol. 2017, no. 1, p. 13, 2017.

[12] M. Nanni and D. Pedreschi, "Time-focused clustering of trajectories of moving objects," *Journal of Intelligent Information Systems*, vol. 27, no. 3, pp. 267–289, 2006.

[13] P. C. Besse, B. Guillouet, J.-M. Loubes, and F. Royer, "Review and perspective for distance-based clustering of vehicle trajectories," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3306–3317, 2016.