Variational Denoising Autoencoders and Least-Squares Policy Iteration for Statistical Dialogue Managers

Vassilios Diakoloukas*, Fotios Lygerakis*, Michail G. Lagoudakis* and Margarita Kotti†

***Abstract*—The use of Reinforcement Learning (RL) approaches for dialogue policy optimization has been the new trend for dialogue management systems. Several methods have been proposed, which are trained on dialogue data to provide optimal system response. However, most of these approaches exhibit performance degradation in the presence of noise, poor scalability to other domains, as well as performance instabilities. To overcome these problems, we propose a novel approach based on the incremental, sample-efficient Least-Squares Policy Iteration (LSPI) algorithm, which is trained on compact, fixed-size dialogue state encodings, obtained from deep Variational Denoising Autoencoders (VDAE). The proposed scheme exhibits stable and noise-robust performance, which significantly outperforms the current state-of-the-art, even in mismatched noise environments.**

***Index Terms*—Variational Autoencoders, Denoising, Dialogue Systems, sample-efficient Statistical Dialogue Managers, Least-Squares Policy Iteration**

I. Introduction

The progress made in speech and natural language processing has established the area of Spoken Dialogue Systems (SDS). An essential component of any SDS is the Dialogue Manager (DM) [1]. Typically, dialogues are temporal processes, which exhibit an exponentially increasing uncertainty as the environmental noise, the user ambiguous response level and the dialogue complexity rise. To overcome this problem, data-driven, Statistical DMs (SDMs) have been considered.

A recent trend is to use large amounts of dialogue data at sentence level to train sequence-to-sequence neural networks in an end-to-end approach [2], [3]. On the other hand, the task-oriented systems, which can be used in domain-specific services, mostly model the dialogue as either a Markov Decision Process (MDP) [4] or a Partially Observable Markov Decision Process (POMDP) [5]. The dialogue data are then used in a Reinforcement Learning (RL) framework [1], in which the system optimizes robust and flexible dialogue policies by receiving rewards for each decision taken in a given dialogue state. The proposed method belongs to the second category.

The dialogue state at each time-step is represented as a belief state (BS) vector, which encodes information from the dialogue history, the user intention and his input. For large domains with a huge ontology, such BS vectors result in an exponential increase of the possible trajectories, making the policy optimization intractable. Simplified BS alternatives, such as the summary BS (sumBS) [6], are still relatively large, consisting of a domain-dependent and often redundant set of state-features. Similarly, the state representation in [7] includes not only the user intent and slot, but also confidence scores from the recognizer and the previous system action.

However, the large dimensionality of the state-space can introduce performance degradation and instability, which is the issue we aim to solve with the use of autoencoders to transform the state-space. The issue of degradation and instability is studied in [8], where a comparison of the most common neural-network-based RL algorithms, such as Deep Q-learning Networks (DQN) [9], Advantage Actor Critic (A2C) [10] and episodic Natural Actor Critic (eNAC) [11], is made on different simulated environments and domains using the sumBS dialogue state representation. The comparison also included GP-SARSA [12]. In all cases, poor generalization and performance instabilities were observed, particularly in large domains, and significant performance degradation, when the noise increases.

To overcome these problems, alternative BS representations have been proposed, with this paper proposing a representation providing significantly better results than the state-of-the-art. Such an alternative are the domain-independent features (DIP) [13], which are fixed in size across the domains, however they don't include possibly useful domain-specific information and they consist of correlated features. In [14] an extremely compact state-space representation is introduced, which has shown large performance instabilities. In [15], a feed forward network (FNN) and a recurrent neural network (RNN) are used to automatically train feature extractors in the form of feature functions for each slot to obtain dialogue BS abstractions. The training can be performed jointly with a DQN-based policy [16]. However, the resulting system is prone to non-optimal decisions and slow convergence.

In [17] a feature selection mechanism is utilized, based on the weight values of the learning algorithm. The weights are initially approximated with the Temporal Difference (TD) algorithm and the selected features are then used in the framework of the sample-efficient Least-Squares Policy Iteration (LSPI) [18]. Recently, in [19], state-of-the-art performance was obtained using deep denoising autoencoders to produce robust BS encodings for the GP-SARSA learning algorithm.

A. Contribution

In this work, we propose a novel use of the incremental LSPI, originally proposed in [18], as a dialogue manager.

*School of Electrical and Computer Eng, Technical Uni. of Crete, Greece.
† Artificial Intelligence, Deloitte, London, UK.
F. Lygerakis and M. Kotti were with Toshiba Research Europe Limited, Cambridge, UK, when part of this research was conducted.

LSPI is a parametric, model-free RL algorithm that combines policy iteration with linear value function approximation and has many key advantages from which a dialogue manager can benefit. These include the elimination of the slow stochastic approximation procedure, LSPI's excellent sample efficiency, its ability to converge in just a few iterations and the small policy footprint. To further enhance the algorithm's flexibility and convergence, we propose using the incremental form of LSPI for on-line policy optimization. A thorough research in the literature has led to only one other related work in [17], however in this case the input to the policy manager is not the output of a formal spoken language understanding analysis, but a set of heuristic, hand-crafted, domain-dependent features. Furthermore, the number of actions is limited and the form of LSPI used is not incremental.

A second contribution is that our approach capitilizes on a new state-space representation based on noise-robust and compact encodings obtained through unsupervised training of deep Variational Denoising Auto-Encoders (VDAE). In contrast to our recent work in [19], here we combine these encodings with LSPI instead of GP-SARSA. Our scheme proves its efficiency under a variety of noisy conditions, both matched and mismatched, outperforming the current state-of-the-art.

II. Background and Methodology

A. LSPI for Dialogue Management

In the framework of a statistical DM, the belief of the dialogue state at each turn t, is represented as a state vector $\boldsymbol{s}_t$. Ideally, this state representation encodes explicit and implicit dialogue information, including the user spoken input hypothesis, the inferred user intention and the dialogue history. At each step of interaction, the manager chooses an action $a_t \in \mathcal{A}$ from a finite set of possible actions $\mathcal{A}$. The decision is made through a policy $\pi(\boldsymbol{s}_t) = a_t$, based on the current state $\boldsymbol{s}_t$. Then, the manager observes the resulting next state $\boldsymbol{s}_{t+1} = \boldsymbol{s}'_t$ and the reward r_t received based on a reward function mapping $R(\boldsymbol{s}_t, a_t) = r_t$. The policy is optimized based on the tuples $(\boldsymbol{s}_t, a_t, r_t, \boldsymbol{s}'_t)$ of samples, so as to maximize the cumulative reward of a dialogue, estimated by a state-action value function $Q^\pi(\boldsymbol{s}, a)$, i.e. $\pi(\boldsymbol{s}) = \text{argmax}_a\, Q^\pi(\boldsymbol{s}, a)$. Since the state-space of a dialogue is often high-dimensional, the value function within LSPI is approximated by a linear architecture:

$$\widehat{Q}^\pi(\boldsymbol{s}, a) = \sum_{j=1}^{k} \phi_j(\boldsymbol{s}, a) \cdot w_j^\pi = \boldsymbol{\phi}(\boldsymbol{s}, a)^\top \boldsymbol{w}^\pi \quad (1)$$

where $\phi_j(\boldsymbol{s}, a), j = 1, \cdots, k$ are fixed, but arbitrary basis functions or features of state $\boldsymbol{s}$ and action a and are assumed to be linearly independent. The w_j^π's are then adjustable parameters or weights, which denote the contribution of the j-th basis function and suffice to represent and recover the entire system's policy for any given state.

1) Incremental LSTDQ: In the framework of LSPI, the learning process of the weights is typically performed using the LSTDQ [18] algorithm. The exact values of weights are obtained by solving the $k \times k$ linear system: $\mathbf{A}\boldsymbol{w}^\pi = \mathbf{b} \Leftrightarrow \boldsymbol{w}^\pi = \mathbf{A}^{-1}\mathbf{b}$, where $\mathbf{A}$ and $\mathbf{b}$ are updated for every given sample in the dialogue. For large action and state spaces, the required inversion of the matrix $\mathbf{A}$ is computationally very expensive. Also, very often $\mathbf{A}$ is not full rank, since in the statistical dialogue systems framework, the belief state vector can be high dimensional and sparse and results in large zero blocks within the $\mathbf{A}$ matrix.

To avoid the matrix singularities and the high computational cost of multiple matrix inversions, we adopt a more efficient form of the LSTDQ algorithm [18], [20] in which the updates based on the training samples are made directly to the inverse $\mathbf{B} = \mathbf{A}^{-1}$ of the rotation matrix and the update equations take the following form for any sample $(\boldsymbol{s}_t, a_t, r_t, \boldsymbol{s}'_t)$:

$$\mathbf{B}^{t+1} \leftarrow \mathbf{B}^t - \frac{\mathbf{B}^t \phi(\boldsymbol{s}_t, a_t)\big(\phi(\boldsymbol{s}_t, a_t) - \gamma\phi(\boldsymbol{s}'_t, \pi(\boldsymbol{s}'_t))\big)^\top \mathbf{B}^t}{1 + (\phi(\boldsymbol{s}_t, a_t) - \gamma\phi(\boldsymbol{s}'_t, \pi(\boldsymbol{s}'_t)))^\top \mathbf{B}^t \phi(\boldsymbol{s}_t, a_t)} \quad (2)$$

$$\mathbf{b}^{t+1} \leftarrow \mathbf{b}^t + \phi(\boldsymbol{s}_t, a_t) r_t, \quad (3)$$

where γ is a discount factor. The weights are recovered at any time by simple matrix-vector multiplication: $\boldsymbol{w}^\pi = \mathbf{B}\mathbf{b}$. In this way we can obtain incremental policy updates for small training batches or even a single training episode and transform LSPI into an online RL algorithm, while keeping the benefits of the data efficiency and stability.

2) Block Basis Functions: The basis functions $\phi_j(\boldsymbol{s}, a)$ in Equation (1) can be defined in various forms, which are intended to capture the underlying structure of the actual state-action pairs. However, given the complexity, dimensionality and ambiguity of the state-action space, the definition of a meaningful, natural set of basis functions is not straightforward and there is always the risk to define basis functions that are not even linearly independent, leading to poor approximations of the Q-value function.

In this paper, we choose block basis state-action feature encodings due to their inherent simplicity. Specifically, for a domain with n discrete dialogue actions $\mathcal{A} = (a_1, a_2, \cdots, a_n)$, the block-basis feature vector is constructed as a concatenation of n block vectors, each having the size d of the state-vector $\boldsymbol{s}$. Then, the block-basis feature vector for the state $\boldsymbol{s}$ and a corresponding active action a is created as: $\phi(\boldsymbol{s}, a) = [c_1\boldsymbol{s}, c_2\boldsymbol{s}, \cdots, c_n\boldsymbol{s}]^\top$, where $c_i, i = 1, \cdots, n$ is a binary indicator variable, defined as:

$$c_i = \begin{cases} 1, & a_i = a \\ 0, & otherwise \end{cases}$$

Although this can be an elegant representation for the dialogue domain, since it nicely discriminates the action contribution, it has a main disadvantage. Even a small-to-medium action space can result to sparse and high dimensional block-basis features, which adds large computational burdens during the policy learning process and makes the use of large action spaces impractical. It is therefore necessary to obtain compact state representations.

B. VDAE State-Space Encodings

To obtain noise-robust and compact state-vectors, we utilize deep, feed-forward autoencoder (AE) architectures consisting of N hidden layers. AEs are the concatenation of two symmetrical neural-networks, the encoder and the decoder. The input to the encoder, $\boldsymbol{x} \in \mathbb{R}^{d_x \times 1}$, is considered to be in this work the summary Belief State representation (sumBS) [6]

with a domain-specific length d_x. The encoder then projects this input to a smaller latent space through a successive set of hidden layers $h_j, j = 1, \cdots, \lceil \frac{N}{2} \rceil$ with gradually reduced dimensions. The projection process is based on the functions:

$$h_1(\boldsymbol{x}) = f\left(\mathcal{W}_{h_1 x}\boldsymbol{x}\right) \text{ and } h_j(\boldsymbol{x}) = f\left(\mathcal{W}_{h_j h_{j-1}} h_{j-1}(\boldsymbol{x})\right),$$

where $f = \tanh()$ denotes the activation function, $\mathcal{W}_{h_1 x} \in \mathbb{R}^{d_1 \times d_x}$ defines the matrix of weights connecting the input and the first hidden layer and $\mathcal{W}_{h_j h_{j-1}} \in \mathbb{R}^{d_j \times d_{j-1}}$ are the weight matrices between the j and $j-1$ hidden layers.

The latent representation from the bottleneck layer $h_{\lceil \frac{N}{2} \rceil}$, is then mapped back to the input $\boldsymbol{x}$ through a reverse mapping function defined in the decoder, with transposed weight matrices $\mathcal{W}^{dec} = \mathcal{W}^{\top}$. The same latent vector $\boldsymbol{s} = h_{\lceil \frac{N}{2} \rceil}(\boldsymbol{x})$ is used as the compact state input for the LSPI optimization.

The simple AE networks may suffer from poor generalization to unseen data, particularly considering the sparse and continuous nature of the sumBS vector in the input. This is mainly attributed to the form of the latent space, which may contain gaps corresponding to forms of training samples that were never used as input.

To overcome this problem, we used Variational AEs (VAEs) [21], [22]. VAEs create continuous latent spaces, since the latent vectors $\boldsymbol{s}$ are generated randomly from a parametric inference model defined as a multivariate Gaussian distribution $q(\boldsymbol{s}|\boldsymbol{x} : \boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{s} : \boldsymbol{\mu}, \boldsymbol{\sigma})$, where $\boldsymbol{\lambda} = \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$, $\boldsymbol{\mu}$ is the mean vector of the Gaussian and $\boldsymbol{\sigma}$ is the corresponding vector of standard deviations. The $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ vectors are approximated through a corresponding pair of weight matrices $\mathcal{W}_{\boldsymbol{\mu}}$ and $\mathcal{W}_{\boldsymbol{\sigma}}$ in the bottleneck layer $h_{\lceil \frac{N}{2} \rceil}$, which are optimized as parameters of the neural network. Optimization of the VAE is made on the basis of the following loss function:

$$J_{VAE}(\boldsymbol{x}, \boldsymbol{y}) = \mathbb{E}\left[||\boldsymbol{x} - \boldsymbol{y}||^2\right] + D_{KL}\left(q(\boldsymbol{s}|\boldsymbol{x} : \boldsymbol{\lambda})||p(\boldsymbol{s})\right) \quad (4)$$

where $\boldsymbol{y}$ denotes the network's prediction at the output, D_{KL} denotes the KL-Divergence among the true latent variable distribution $p(\boldsymbol{s})$ which is typically chosen to follow the standard Gaussian $\mathcal{N}(0, 1)$ and the approximation $q(\boldsymbol{s}|\boldsymbol{x} : \boldsymbol{\lambda})$ learned by the encoder.

In a typical SDS, the presence of noise, introduced from speech recognition errors, semantic interpretation errors (e.g. acoustic confusability, ambiguity of natural language, incomplete utterances, etc.), as well as the uncertainty of user's goal, can result in poor state representations. Adopting the methodology introduced in [23], we therefore choose to overcome this problem by forcing the VAE to compensate with artificially corrupted sumBS vectors $\widetilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{n}$ denotes the noise vector produced by an unknown distribution and is applied to the clean sumBS $\boldsymbol{x}$. In this scheme, which we call Variational Denoising AEs (VDAE), the inference model $q(\boldsymbol{s}|\widetilde{\boldsymbol{x}} : \boldsymbol{\lambda})$ learns to generate latent vectors from corrupted sumBS state vectors $\widetilde{\boldsymbol{x}}$ that have been artificially injected with noise at different Semantic Error Rates (SER). Specifically, the corrupted vector $\widetilde{\boldsymbol{x}}$ is obtained considering the true semantic information for a slot with probability $1 - P_{SER}$, and making random slot value selections from all the available values in the ontology, with probability P_{SER}. It is worth noting that this artificial corruption process can be applied to data from both real and simulated dialogues. In this way, the VDAE will be trained to be tolerant in noise.

III. Experiments

To evaluate the proposed methodology, we implemented both the VDAE and the incremental LSPI algorithm within the PyDial toolkit [24]. In the proposed approach, both the LSPI policy and the VDAE are optimized in parallel using batches of dialogue episodes. Since the optimization of the VDAE is an unsupervised learning process, it doesn't require an optimized policy to converge. We therefore, choose to pre-train the VDAE on randomly generated sumBS vectors. Further optimization is then performed in parallel to the LSPI, to dynamically adapt to the environment's uncertainty.

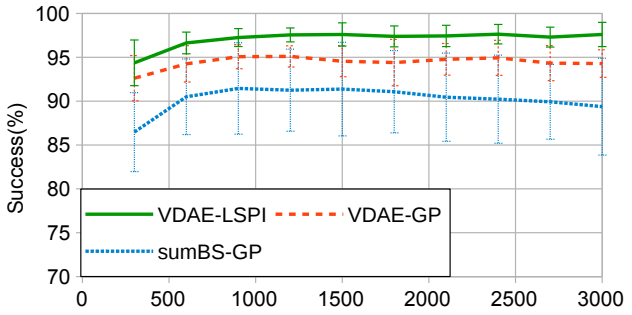
We performed a series of experiments on simulated dialogues in the following domains: *Cambridge Restaurants (CR)*, *Laptops11 (LAP11)* and *San Francisco Restaurants (SFR)*, with 17, 38 and 23 discrete system actions respectively [15]. For each domain, we used a different VDAE although they all shared the same hyper-parameters and an architecture of 7 hidden layers with gradually reduced size. The initial sumBS size, which was 265, 257 and 636 for the CR, LAP11 and SFR domains respectively, was encoded in a latent vector of length 30 in all cases. ADAM optimizer, an exponentially decaying learning rate and a dropout rate of 0.6 were used. LSPI weights were also randomly initialized and the optimization was made incrementally per dialogue sample. The total reward at the end of each dialogue is computed as: $R = I_S - T$, where I_S is an indicator function, with $I_S = 20$ for successful dialogues and $I_S = 0$ otherwise, and T is the number of dialogue turns which are limited to 20. Our experiments were configured for 0%, 15%, 30% and 45% SER noise levels. The evaluation was performed after every 300 training samples, using 300 test dialogues. To be fair in our evaluation, we performed 5 independent runs for each experiment using the same randomly varying initializations of the random number generator.

Dialogues: 3000		**Domains**		
SER	BS	CR	LAP11	SFR
0%	sumBS-GP	98.4%(±1.1)	86.8%(±3.8)	95.2%(±1.3)
	VAE-GP	96.5%(±2.7)	94.7%(±1.7)	93.7%(±2.1)
	VAE-LSPI	**99.7%**(±0.4)	**98.5%**(±1.1)	**98.4%**(±1.1)
15%	sumBS-GP	96.4%(±2.2)	66.5% (±2.3)	81.6% (±1.6)
	VDAE-GP	95.5%(±0.8)	91.7%(±0.5)	93.6%(±1.0)
	VDAE-LSPI	**99.0%**(±0.8)	**97.9%**(±0.6)	**97.0%**(±0.7)
30%	sumBS-GP	88.5%(±4.2)	51.4%(±9.3)	66.3%(±5.3)
	VDAE-GP	92.9%(±1.3)	90.2%(±1.9)	89.9%(±2.7)
	VDAE-LSPI	**98.1%**(±1.2)	**97.7%**(±0.9)	**95.3%**(±1.6)
45%	sumBS-GP	78.0%(±3.4)	24.1%(±5.5)	53.9%(±6.8)
	VDAE-GP	92.3%(±3.3)	87.9%(±2.7)	88.6%(±2.9)
	VDAE-LSPI	**93.6%**(±2.7)	**96.7%**(±0.9)	**94.6%**(±1.5)

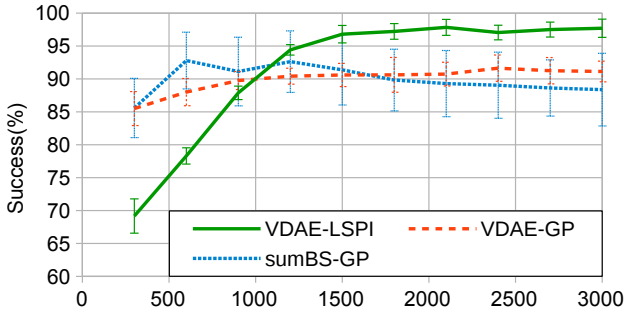
TABLE I: Average dialogue success after 3000 dialogues. Standard deviation in parenthesis. Best score in bold.

Table I summarizes the results of our experiments. We show average dialogue success after 3000 dialogues for the baseline sumBS with GP-SARSA (sumBS-GP), the VDAE with GP-SARSA (VDAE-GP) and the VDAE with LSPI (VDAE-LSPI). The performance superiority of the VDAE-based representations is even more prominent when the noise level increases.

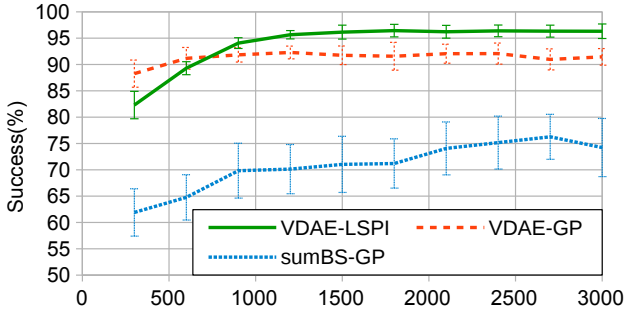
However, it is the VDAE-LSPI that clearly wins in all the environments and domains with the absolute performance gain reaching 8.8% over the VDAE-GP and 72.6% over the sumBS-GP. The average dialogue success rate of the proposed VDAE-LSPI system is retained remarkably high for all the domains and all the noise levels. Similar were the findings in terms of the average total reward, since in all experiments the VDAE-LSPI produced significantly higher rewards.



(a) CR domain



(b) LAP11 domain



(c) SFR domain

Fig. 1: Average performance comparison on multiple noise levels with SER chosen uniformly from 0%, 15%, 30%, 45%.

Figure 1 also shows the average performance trend over the different SER environments as a function of the training samples, for all the domains considered. It can be seen that in all domains the proposed VDAE with LSPI scheme exhibits a smooth performance curve and after approximately 1200 episodes it outperforms the approaches based on the GP-SARSA algorithm. It is worth noting that the deviation of the VDAE-LSPI combination is particularly low, indicating a very robust performance for all noise levels. Furthermore, the VDAE-LSPI retains its top performance after 1200 episodes without fluctuations and instabilities.

Even in mismatched noise conditions, the VDAE-LSPI approach maintains its remarkable, noise-robust performance. For instance, a VDAE-LSPI system trained on 45% SER achieves an average performance of 97.8% ($\pm$0.8) over all domains when tested on 15% SER, while the corresponding average performance for the sumBS-GP and the VDAE-GP is 84.7% ($\pm$10.7) and 91.2% ($\pm$2.3) respectively. This is a good indication that the proposed system will perform equally well in a real life scenario, where the exact noise level may be unknown and varying.

SER/ ENV	**Model**	CR		LAP11		SFR	
		Succ	Rew	Succ	Rew	Succ	Rew
0%	VDAE-LSPI	**99.7%**	13.3	**98.5%**	**11.4**	**98.4%**	11.3
	FM-DGNN [25]	99.0%	**13.8**	79.8%	8.5	98.1%	**12.6**
Env.1	Feudal-DQN [15]	89.3%	11.7	65.5%	5.7	71.1%	7.1
15%	VDAE-LSPI	**99.0%**	**13.1**	**97.9%**	**11.2**	**97.0%**	**10.9**
	FM-DGNN [25]	97.7%	12.9	89.1%	9.1	91.9%	10.2
Env.3	Feudal-DQN [15]	92.6%	**11.7**	89.6%	9.4	90.0%	9.7
30%	VDAE-LSPI	**98.1%**	**12.7**	**97.7%**	**10.8**	**95.3%**	**9.8**
	FM-DGNN [25]	90.9%	10.5	77.8%	5.7	80.4%	6.5
Env.6	Feudal-DQN [15]	90.6%	10.4	78.5%	6.0	83.0%	7.1

TABLE II: VDAE-LSPI average success and reward (3000 episodes) compared to selected state-of-the-art performance on 4000 episodes

A direct comparison with all state-of-the-art approaches is not feasible, due to the different experimental protocols and policies used. However, a fair comparison can be made with selected recent works utilizing PyDial in the same environments as shown in Table II. The reported accuracies for both the Feudal-DQN combination in [15] and the AgentGraph methods such as the FM-DGNN in [25] with 4000 episodes, for the environments 1, 3 and 6, which correspond to our 0%, 15% and 30% SER, are consistently lower than the ones reported here for 3000 episodes. The performance gains of our VDAE-LSPI increase for higher noise levels and for more complex domains such as the LAP11. Similar are the findings when comparing with [8], where the sumBS was used with different RL algorithms on the same domains and environments. A comparison is also possible with [14], where the max accuracy for 45% SER on the CR domain is 52.9% compared to 93.6% in this work, and with [19], which shows that VDAE-LSPI is on average more than 5% better compared to GP-SARSA based approaches.

IV. Conclusion

We propose a novel approach for noise-robust statistical dialogue managers. It uses compact, fixed-size, state-space encodings automatically obtained from VDAEs to perform policy optimization based on an incremental form of the LSPI. We provide evidence that the proposed scheme is sample-efficient and highly noise-robust. Specifically, it exhibits smooth performance curves and superior performance, in both matched and mismatched noise conditions steadily for a multitude of domains. Our proposed scheme consistently outperforms the GP-SARSA based approaches with accuracy gains ranging from 8.8% to 72.6%, and significantly surpasses current top-performing state-of-the-art approaches.

REFERENCES

[1] Steve J. Young, Milica Gasic, Blaise Thomson, and Jason D. Williams, "Pomdp-based statistical spoken dialog systems: A review," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1160–1179, 2013.

[2] Oriol Vinyals and Quoc V. Le, "A neural conversational model," *ArXiv*, vol. abs/1506.05869, 2015.

[3] Iulian V. Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. 2016, AAAI'16, p. 3776–3783, AAAI Press.

[4] Esther Levin, Roberto Pieraccini, and Wieland Eckert, "Using markov decision process for learning dialogue strategies," in *Proc. ICASSP*, 1998, pp. 201–204.

[5] Jason D. Williams and Steve Young, "Partially observable markov decision processes for spoken dialog systems," *Computer Speech and Language*, vol. 21, no. 2, pp. 393–422, Apr. 2007.

[6] Jason D. Williams and Steve Young, "Scaling pomdps for spoken dialog management," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 2116–2129, Sept. 2007.

[7] Maryam Fazel-Zarandi, Shang-Wen Li, Jin Cao, Jared Casale, Peter Henderson, David Whitney, and Alborz Geramifard, "Learning robust dialog policies in noisy environments," in *Conversational AI Workshop (NIPS)*, 2017.

[8] Iñigo Casanueva, Pawel Budzianowski, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Stefan Ultes, Lina Maria Rojas-Barahona, Steve J. Young, and Milica Gašić, "A benchmarking environment for reinforcement learning based task oriented dialogue management," *CoRR*, vol. abs/1711.11023, 2017.

[9] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013.

[10] Mehdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman, "Policy networks with two-stage training for dialogue systems," in *SIGDIAL*, 2016.

[11] Pei-Hao Su, Pawel Budzianowski, Stefan Ultes, Milica Gašić, and Steve J. Young, "Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management," in *SIGDIAL*, 2017.

[12] Yaakov Engel, Shie Mannor, and Ron Meir, "Reinforcement learning with Gaussian processes," in *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, 2005, pp. 201–208.

[13] Zhuoran Wang, Tsung-Hsien Wen, Pei-Hao Su, and Yannis Stylianou, "Learning domain-independent dialogue policies via ontology parameterisation," in *SIGDIAL*, 2015.

[14] Margarita Kotti, Vassilios Diakoloukas, Alexandros Papangelis, Michail Lagoudakis, and Yannis Stylianou, "A case study on the importance of belief state representation for dialogue policy management," in *INTERSPEECH*, 2018, pp. 986–990.

[15] Iñigo Casanueva, Paweł Budzianowski, Pei-Hao Su, Stefan Ultes, Lina M. Rojas Barahona, Bo-Hsiang Tseng, and Milica Gašić, "Feudal reinforcement learning for dialogue management in large domains," in *NAACL-HLT*, 2018, pp. 714–719.

[16] Iñigo Casanueva, Paweł Budzianowski, Stefan Ultes, Florian Kreyssig, Bo-Hsiang Tseng, Yen-Chen Wu, and Milica Gašić, "Feudal dialogue management with jointly learned feature extractors," in *SIGDIAL*, 2018.

[17] Lihong Li, Jason D. Williams, and Suhrid Balakrishnan, "Reinforcement learning for dialog management using least-squares policy iteration and fast feature selection," in *INTERSPEECH*, 2009.

[18] Michail G. Lagoudakis and Ronald Parr, "Least-squares policy iteration," *Journal of Machine Learning Research*, vol. 4, pp. 1107–1149, 2003.

[19] Fotios Lygerakis, Vassilios Diakoloulas, Michail Lagoudakis, and Kotti Margarita, "Robust belief state space representation for statistical dialogue managers using deep autoencoders," in *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU*, 2019.

[20] Alborz Geramifard, Michael Bowling, and Richard S Sutton, "Incremental least-squares temporal difference learning," in *Proceedings of the National Conference on Artificial Intelligence*, 2006, vol. 21, p. 356.

[21] Diederik P. Kingma and Max Welling, "Auto-encoding variational bayes," *The 2nd International Conference on Learning Representations (ICLR)*, 2013.

[22] Kuan Han, Haiguang Wen, Junxing Shi, Kun-Han Lu, Yizhen Zhang, Di Fu, and Zhongming Liu, "Variational autoencoder: An unsupervised model for encoding and decoding fmri activity in visual cortex," *NeuroImage*, vol. 198, pp. 125 – 136, 2019.

[23] Pascal Vincent, Isabelle Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, pp. 3371–3408, 2010.

[24] Stefan Ultes, Lina Maria Rojas-Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Pawel Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gašić, and Steve J. Young, "PyDial: A multi-domain statistical dialogue system toolkit," in *ACL*, 2017.

[25] Lu Chen, Zhi Chen, Bowen Tan, Sishan Long, Milica Gašić, and Kai Yu, "Agentgraph: Toward universal dialogue management with structured deep reinforcement learning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 9, pp. 1378–1391, 2019.