Actor-Double-Critic: Incorporating Model-Based Critic for Task-Oriented Dialogue Systems

Yen-Chen Wu[1], Bo-Hsiang Tseng[1], and Milica Gašić[2]

[1]University of Cambridge, UK
{ycw30, bht26}@cam.ac.uk
[2]Heinrich Heine University Düsseldorf, Germany
gasic@uni-duesseldorf.de

Abstract

In order to improve the sample-efficiency of deep reinforcement learning (DRL), we implemented imagination augmented agent (I2A) in spoken dialogue systems (SDS). Although I2A achieves a higher success rate than baselines by augmenting predicted future into a policy network, its complicated architecture introduces unwanted instability. In this work, we propose actor-double-critic (ADC) to improve the stability and overall performance of I2A. ADC simplifies the architecture of I2A to reduce excessive parameters and hyper-parameters. More importantly, a separate model-based critic shares parameters between actions and makes back-propagation explicit. In our experiments on Cambridge Restaurant Booking task, ADC enhances success rates considerably and shows robustness to imperfect environment models. In addition, ADC exhibits the stability and sample-efficiency as significantly reducing the baseline standard deviation of success rates and reaching the 80% success rate with half training data.

1 Introduction

Spoken Dialogue Systems (SDS) enable human-computer interaction via natural language. The core of SDS, dialogue management, can be formulated as an RL problem (Levin et al., 1997; Young et al., 2013; Williams, 2008). Great advancements can be achieved with deep RL algorithms (Dhingra et al., 2016; Chang et al., 2017; Budzianowski et al., 2017; Casanueva et al., 2017; Liu et al., 2018; Gao et al., 2018; Takanobu et al., 2019; Wu et al., 2020). Yet, deep RL methods are notoriously expensive in terms of the number of interactions they require. Even relatively simple tasks can require thousands of labelled dialogues and modelling complex behaviour such as a multi-domain application might need substantially more (Gašić et al., 2011; Li et al., 2016; Su et al., 2016).

Model-based reinforcement learning (MBRL) is one way of improving sample-efficiency in RL (Tamar et al., 2016; Silver et al., 2016; Gu et al., 2016; Nagabandi et al., 2018; Oh et al., 2017). By learning the environment model, we can predict the future states after taking a certain action. In a dialogue system, that means the system can predict the user's behaviour. In contrast, the model-free RL algorithms only learn the mapping of belief states and Q-values and do not make use of the user behaviour patterns in the training data. In other words, model-free RL is wasting actions by going through similar transitions multiple times to get accurate return estimations.

Dyna-Q (Sutton, 1990; Sutton et al., 2012) has achieved some success in SDS (Peng et al., 2018; Su et al., 2018; Wu et al., 2019; Zhang et al., 2019) by generating training data for agents and keeping improving its environment model from real interactions between agents and users. Nevertheless, the noisy data generated by inaccurate environment models could adversely affect the experience replay buffer and result in convergence toward suboptimal performance. This problem is even more critical in real-world tasks such as real-world dialogue systems where training a good environment model is challenging.

I2A (Weber et al., 2017) addresses this problem by augmenting model-based information into the input of policy networks in order to filter out the noise generated by poor environment models. However, I2A introduces unwanted instability when we applied it to a dialogue system due to its complex architecture and excessive hyper-parameters. The unstable performance makes it even harder to tune the parameters.

In this paper, we propose Actor-Double-Critic (ADC), a new architecture to augment model-based information into the policy network. By training two critics from model-free and model-based data

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 854–863
November 16 - 20, 2020. ©2020 Association for Computational Linguistics

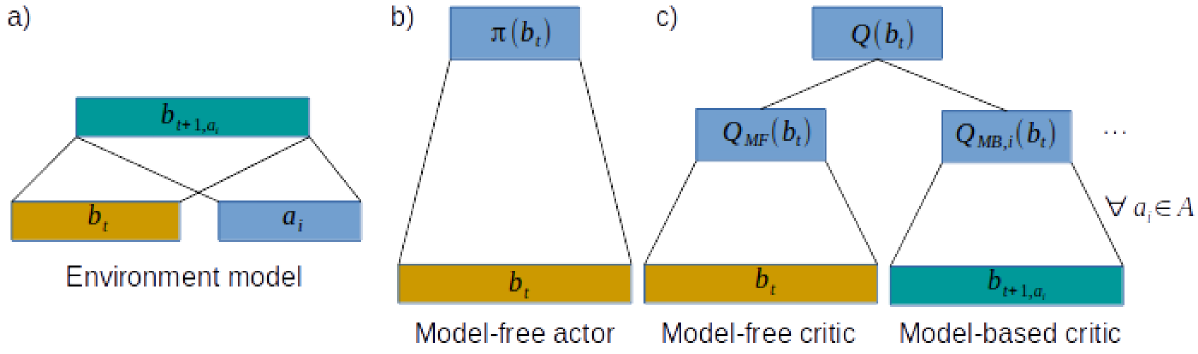

Figure 1: ADC architecture: Green blocks indicate predicted belief states. a) the environment model predicts the next time step b_{t+1,a_i} conditioned on an action a_i. b) the actor outputs the policy pi as in a standard actor-critic architecture. c) the two critics estimate Q-values based on the current belief state and predicted next belief states respectively. Final Q-values are the weighted sum of the outputs of two critics. Note that model-based critic predicts i-th Q-value based on b_{t+1,a_i}, so this process is repeated for all actions $a_i \in A$ to obtain all of the Q-values.

separately and combining them in an ensemble, we reduce the number of redundant parameters and make back-propagation more efficient. In the Cambridge Restaurant dialogue system task, experimental results show a substantial improvement in success rates. Regarding sample efficiency, ADC takes only half of baseline training data to achieve the 80% success rate. In addition, ADC is the most stable approach among all considered baselines. Compared to a model-free actor-critic algorithm, ACER (Wang et al., 2016), it reduces the standard deviation of success rates from 7.7 to 1.2. It also proves more stable than a Bayesian model-free algorithm GP-SARSA (Gašić et al., 2010).

2 Dialogue management through reinforcement learning

Dialogue management can be cast as a continuous MDP (Young et al., 2013) composed of a continuous multivariate belief state space B, a finite set of actions A and a reward function $R(b_t, a_t)$. The belief state b is a probability distribution over all possible (discrete) states. At a given time t, the agent (policy) observes the belief state $b_t \in B$ and executes an action $a_t \in A$. The agent then receives a reward $r_t \in R$ drawn from $R(b_t, a_t)$. The policy π is defined as a function $\pi : B \times A \rightarrow [0, 1]$ that with probability $\pi(b, a)$ takes an action a in a state b. For any policy π and $b \in B$, the value function V_π corresponding to π is defined as:

$$V^\pi(b) = \mathbb{E}\{r_t + \gamma r_{t+1} + ...|b_t = b, \pi\} \qquad (1)$$

where $0 \leq \gamma \leq 1$, is a discount factor and r_t is a one-step reward. The objective of reinforcement learning is to find an optimal policy π^*, i.e. a policy that maximizes the value function in each belief state. Equivalently, the goal is to find an optimal policy π^* that maximises the discounted total return

$$R = \sum_{t=0}^{T-1} \gamma^t r_t(b_t, a_t) \qquad (2)$$

over a dialogue with T turns, where $r_t(b_t, a_t)$ is the reward when taking action a_t in dialogue state b_t at turn t and γ is the discount factor.

3 Imagination Augmented Agent (I2A)

I2A (Weber et al., 2017) manages to implicitly incorporate all the possible future information into the policy network. Basically, it can be divided into three hierarchies:

- **Imagination core**. An environment model is trained on future states and rewards prediction conditioned on an action. By interacting with a baseline actor, the environment model is used to simulate potential trajectories.
- **Single imagination roll-out**. To efficiently use these simulated trajectories, the agent learns an encoder that extracts information from these imaginations including both states and rewards. The encoder is designed to select useful information and ignore the noisy one generated by imperfect models.
- **Augmentative architecture**. For each possible action, the simulated trajectories are generated. All the information extracted from

trajectories are concatenated together and provided as additional context to a policy network.

However, we found that I2A's hierarchical architecture is not stable enough when experimented on SDS tasks. This architecture contains several fragile components which have a strong impact on the performance, such as the environment model and the roll-out policy network. Excessive hyperparameters, like rollout-depth and embedded feature sizes for the encoder, also make it hard to conduct parameter tuning and apply I2A to real-world applications.

4 Actor-Double-Critic (ADC)

To increase the stability of the augmenting-style approaches, we simplify the previous architecture and propose a key component – model-based critic. As illustrated in Figure 1, we train two critics based on model-free and model-based information respectively and combine their outputs by the weighted sum in an ensemble.

In this section, we explain why we simplify the architecture in these ways and the benefits of using a model-based critic.

4.1 Simplified architecture

To reduce the model complexity, we simplify the architecture in the following three ways,

- Our environment model predicts only the next belief state b_{t+1,a_i} conditioned on an action a_i: the model does not predict **rewards**. That is because the reward signals in SDS domain are sparse and hard to predict.

- In I2A, the pre-trained environment model will not be updated while learning policy since the policy network is robust to imperfect model. Besides, obtaining pre-training data is not challenging in a simulated game. However, in the real world, pre-training data for SDS is hard to collect. In our approach, in order to improve the sample efficiency, the environment model is **updated during policy learning**.

- We discard the **roll-out policy network**. Since the policy always changes, the predicted action sequences change as well. Since we aim at reducing the uncertainties in our framework, roll-out length is set to 1 without using the roll-out policy network.

4.2 Model-based critic

By definition, a Q-value can be decomposed as:

$$Q_i^\pi(b_t, a_i) = r_t + \gamma V(b_{t+1,a_i}) \qquad (3)$$

In dialogue system tasks, r_t is typically set to -1 for each turn to penalize lengthy dialogue in our experimental setting. At the end of a dialogue, r_t varies depending on the result yet we do not need to predict Q-values at that time. Hence, r_t is a constant in Equation 3 for dialogue system tasks. Given that r_t and γ are constants, we can train an estimator for $Q_i^\pi(b_t)$ based on the next belief state b_{t+1,a_i}, which is predicted by the environment model. [1]

We call this estimator *model-based critic* in the actor-critic framework, while the original one is a *model-free critic*. Compared to previous approaches, adopting the model-based critic has the following three benefits:

4.2.1 Parameter sharing

Note that given b_{t+1,a_i}, the model-based critic of ADC predicts only one value Q_i. To obtain all of the Q-values, we firstly predict the next belief states $b_{t+1,a_i} \forall a_i \in A$ using the environment model, and then map each of them to Q_i by the model-based critic. Parameters of the model-based critic are shared between actions and the model-complexity is reduced.

In I2A, $b_{t+1,a_i} \forall a_i \in A$ are concatenated as a large input vector. This means the the number of parameters of the model-based path of I2A is increasing with the number of actions, which is not the case in ADC. In practice, the number of parameters in I2A (1.4 millions) is around five times more than ADC (240 thousands).

4.2.2 No redundant connections

As shown in Equation 3, Q_i is not relevant to other predicted belief state b_{t+1,a_j} where $i \neq j$. Q_i results from the predicted belief state b_{t+1,a_i}. But I2A concatenates all of the predicted belief states and the current belief state together to make the prediction of Q-value. That is, most of the connections in I2A should be updated to zero weights after training. Using model-based critic eliminates these redundant connections and predicts one Q_i at one time to improve the stability of the algorithm.

[1] In other applications where r_t is not a constant, the environment model should also predict the value of r_t.

Algorithm 1: Actor-Double-Critic for Dialogue Policy Learning

```
Input: Total training epochs $N$, the environment model $E$ with
       parameters $\theta_E$ , the model-based critic $MB$, the
       model-free critic $MF$, the actor (Policy network) $P$
       with parameters $\theta_P$ , the experience replay $D$
pre-trained $E$ with precollected conversational data
for $n$=$1$:$N$ do
    // Reinforcement Learning
    while $s$ is not a terminal state do
        predict $b_{t+1,a_i} \forall a_i \in A$ using $E$
        predict $Q_{MB}$ using $MB$
        compute $Q(b_t, a_i)$ by Eq. 4
        with probability $\epsilon$ select a random action $a$ otherwise select
          $a = argmax_{a'} P(b, a')$
        execute $a$, and observe the next belief state $b'$ and reward $r$ update
          dialogue state to $b'$
        store $(b, a, r, b')$ in D
    end
    sample random minibatches of $(b, a, r, b')$ from $D$
    update $\theta_{MF}$, $\theta_{MB}$ via minibatch $Q$-learning according to Equation 4, 5
    update $\theta_P$ according to ACER or another actor-critic algorithms
    // Environment model Learning
    sample random minibatches of training samples $(b, a, r, b')$ from $D$
    update $\theta_E$ via minibatch SGD of multi-task learning
end
```

Agent	#Parameters
ACER	110 K
I2A (Model-free path)	80 K
I2A (Model-based path)	1.2 M
I2A (Total)	1.4 M
ADC (Model-based critic)	110 K
ADC (Total)	240 K
Environment Model	16 K

Table 1: Comparison of the number of parameters.

4.2.3 Explicit update signals

We can also predict $Q^{\pi}(b_t)$ through the model-free critic. The final Q-values are the weighted sum of both two critics in an ensemble way to lower the variance.

$$\begin{aligned} Q^{\pi}(b_t, a_i) =& Q^{\pi}_{MF}(b_t, a_i) \cdot w + \\ & Q^{\pi}_{MB}(b_{t+1,a_i}) \cdot (1 - w), \end{aligned} \tag{4}$$

where $Q^{\pi}_{MF}(b_t, a_i)$ is the output of the model-free critic and $Q^{\pi}_{MB}(b_{t+1,a_i})$ is the output of the model-based one, and w is a weight parameter. We replace their notation with Q^{π}_{MF} and Q^{π}_{MB} to keep the expressions succinct. The model selects information either from the model-free path (when $w = 1$) when the model is noisy or from the model-based path (when $w = 0$) when it provides more accurate information. During the training process, we compute the loss for each critic and w is a hyper-parameter.

$$\begin{aligned} loss_{critics} =& (Q^{\pi}_{MF} - Q^{ret})^2 + \\ & (Q^{\pi}_{MB} - Q^{ret})^2 \end{aligned} \tag{5}$$

where Q^{ret} is the target of Q_{π} using the Retrace algorithm (Munos et al., 2016).

Note that for each training iteration, we update two critics at the same time. In I2A, we cannot identify whether errors are coming from model-based path or model-free path. In our approach, the information flows from two sources clearly instead of an ambiguous one. We have tried to back-propagate loss from Q_{π} through the whole network, but the result is better if we back-propagate the loss defined in equation 5. This result again proves the necessity of using two-critics architecture.

5 Experiments

5.1 Setup

Experiments are conducted on the Cambridge restaurant domain from the PyDial toolkit (Ultes et al., 2017) with a goal-driven user simulator operating on the semantic level (Schatzmann et al., 2007; Schatzmann and Young, 2009), a LSTM-based NLU model (Mrkšić et al., 2016), and a NLG model (Wen et al., 2015). During training, an agent is updated when a dialogue terminates, which is an iteration. Every 200 training dialogues, the agent is tested on 500 dialogues. 10 random seeds were run

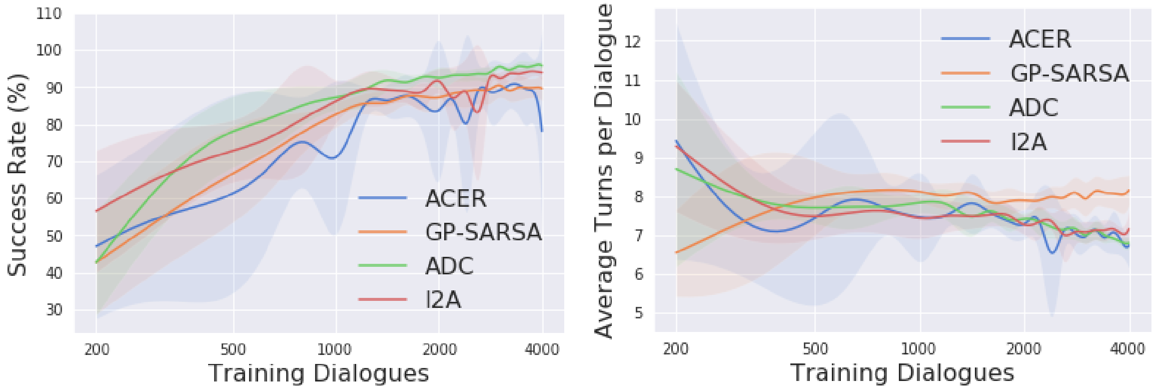

Figure 2: Comparison with baselines. *Left:* Learning curves of success rate. *Right:* average turns per dialogue.

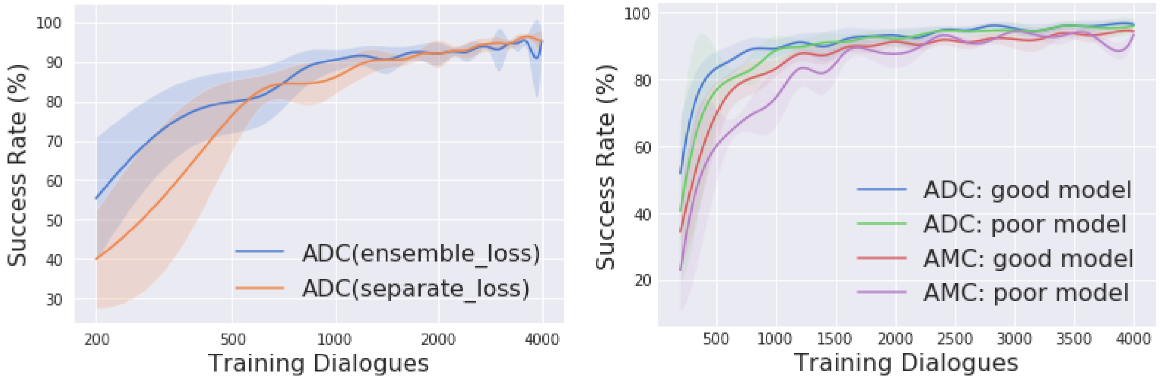

Figure 3: *Left:* Comparison between different update algorithms. *Right:* Experiment on robustness to imperfect model over different architectures.

for each approach to analyze the variance arising from different initialization. The mean ± standard deviation is depicted as the shaded area in Figure 2, 3. The x-axes of Figure 2, 3 are in log scale to put emphasis on both the early stage and the final performance of the training process.

User simulator. To accommodate for ASR error, 15% semantic error rate (SER) is included in the user simulator. The maximum dialogue length is set to 25 turns and γ was 0.99. The reward is defined as 20 for a successful dialogue minus the number of turns it took to complete the dialogue.

Implementation details. The input for all models is the full dialogue belief state b of size 268 and the output action space consists of 16 possible actions. For NN-based algorithms, the size of a mini-batch is 64. ϵ-greedy exploration is used, with ϵ linearly reducing from 0.3 down to 0 over the training process. Two hidden layers are of size 300 and 100 for actor and critic. The Adam optimiser was used with an initial learning rate of 0.001 (Kingma and Ba, 2014). For algorithms employing experience replay, the replay memory has a capacity of 2000 interactions.

5.2 Dialogue agents for comparison

- **GP-SARSA** is a Bayesian baseline, which provides a stable performance by utilising uncertainty estimates.

- **ACER** is the model-free actor-critic baseline and can be perceived as a model-free counterpart of the proposed method. According to the benchmark results (Casanueva et al., 2017), it performs better than other actor-critic methods such as A2C (Fatemi et al., 2016) and eNAC (Su et al., 2017). Since ADC can be applied to any model-free actor-critic method, not all the performance of RL algorithms are reported here. In this paper, we focus on the gap between ACER and ADC rather than the absolute performance. To have a fair comparison, the pre-training data used by model-based

Agent	Suc.	Std.	Required data
ACER	78.1	±7.7	1200
GP-SARSA	89.6	±3.3	800
I2A	93.9	±2.3	750
ADC	95.8	±1.2	600

Table 2: Final performance of each agent after training with 4000 dialogues. Tested in 10 runs, each algorithm reports 1) the average success rate 2) the standard deviation of success rates and 3)the average amount of data required to reach the 80% success rate. The latter two matrices are used to evaluate the stability and sample-efficiency respectively.

approaches were put into the experience buffer of ACER at the beginning of the training.

- **I2A** is the model-based baseline. The environment model is pre-trained with 400 dialogues generated by interactions between a simulated user and an agent.
- **ADC** is the proposed method. The ensemble weight w is 0.5 for each critic. The environment model setting is the same as I2A.

5.3 Comparison with baselines

Success rate. As shown in the left part of Figure 2 and Table 2, ADC outperforms other methods considerably in terms of sample-efficiency, stability, and success rate. I2A performs better than ACER but is still fragile to the initialization, shown as the shaded areas. Compared to I2A, ADC reduces half of the standard deviation of final success rates, from 2.3 to 1.2

In contrast, GP-SARSA is quite stable due to its Bayesian nature. While the standard deviation of the final success rate of I2A is smaller than GP-SARSA, I2A is more unstable in the early stage of the training process. It is worth noticing that ADC is even more stable than GP-SARSA, and reach higher performance in the end. In terms of sample efficiency, ADC uses only half of the data (600 dialogues) to reach the 80% success rate, compared to ACER (1200 dialogues).

Average turns per dialogue As shown in the right part of Figure 2, GP-SARSA takes more turns than other algorithms, and only decrease slightly during training. We found that GP-SARSA tends to take more turns to confirm user intention to stabilize its performance, while some of these confirmations are not necessary. Other approaches steadily reduce the number of turns during the process of training.

5.4 Different back-propagation styles

In the left part of Figure 3, the red line is the learning curve of the agent that back-propagates only one loss from the ensemble output Q, while the brown line is the agent that update each critic separately and the loss back-propagate from ensemble output only pass through ensemble weight w.

We can note that the agent with the separate loss function (as in equation 5) is more stable than the other method. This is because when the ensemble Q closes to Q^{ret}, Q_{MF} and Q_{MB} are not necessarily close to the target Q^{ret}. In contrast, the separate update can make sure each of output value is accurate.

5.5 Robustness to imperfect models

In order to examine the impact of the environment model on ADC, we propose another baseline, actor-model-based-critic (AMC). AMC only use model-based critic to predict Q-value without the model-free critic, so the quality of environment model is critical to AMC. In the experiment, a good environment model is pre-trained with 400 dialogues, and a poor environment model is pre-trained with only 200 dialogues.

In the right part of Figure 3, we can observe that ADC maintains its performance with poor model, while AMC's performance drops a lot. This might be because a poor environment model cannot lead to accurate value-prediction. The aid from a model-free critic is also substantial.

5.6 Comparison in different environment settings

To further investigate the properties of ADC, we test it on 6 different environments (simulated user) settings. For each setting, we report the final performance of each agent after training it with 4000 dialogues. Semantic error rate (SER) models the noise from the ASR and NLU channel (Thomson et al., 2012). In addition to the standard user, an unfriendly one is defined, where the user barely provides any extra information to the system. The action masking mechanism is used in environment 1 & 3 to reduce the action space. The setting of each simulated user is listed in Table 3.

The results are shown in Table 4. In clean environments (1 & 3), ACER learns well after 4000

	Env. 1	Env. 2	Env. 3	Env. 4	Env. 5	Env. 6
SER	0%	0%	15%	15%	15%	30%
Masks	On	Off	On	Off	On	On
User	Standard	Standard	Standard	Standard	Unfriendly	Standard

Table 3: The settings of different environments.

	GP-SARSA		ACER		I2A		ADC	
Task	Suc.	Turns	Suc.	Turns	Suc.	Turns	Suc.	Turns
Env. 1	**99.2**%	6.4	98.6%	6.0	97.9%	6.0	99.1%	6.0
Env. 2	95.7%	7.2	87.3%	6.5	79.8%	5.8	**98.7**%	6.0
Env. 3	95.8%	7.7	95.3%	7.1	**96.3**%	7.0	96.1%	7.0
Env. 4	89.6%	8.2	78.1%	6.7	93.9%	7.2	**95.8**%	6.8
Env. 5	92.5%	9.6	94.0%	8.2	94.2%	8.0	**95.6**%	8.0
Env. 6	90.0%	9.0	81.0%	8.1	87.9%	8.1	**92.0**%	7.9

Table 4: Success rates and average turns after 4000 training dialogues. The highest success rate is highlighted.

dialogues. Yet, in noisy environments (2 & 4), ADC outperforms ACER significantly. In environment 5, an unfriendly user was used. But this defect does not affect the algorithms a lot as action mask is used, so the number of available actions are reduced and therefore the task is less difficult. It is worthy to note that in environment 6, ADC outperforms hand-crafted policy (89.6% (Casanueva et al., 2017)) and demonstrates the flexibility of reinforcement learning that can learn from environments. Overall, ADC demonstrates its robustness in all environments especially for the environments without action masks.

5.7 Case study

To further investigate the behaviour of different agents during the training process, we sampled a dialogue session in environment 4 (the setting of the environment is mentioned in section 4.6) after 500 epochs. The agent of ACER and ADC have 57% and 88% success rate respectively.

As shown in Table 5, ACER informs the restaurant in the early stage while ADC is more conservative and takes more turns. ADC asks more questions before giving the recommended restaurant and sometimes confirms the booking to make sure the one it provides fulfills all the requirements. Besides, ACER keeps asking the same question and sometimes gives a completely wrong reply. That is because, without the aid of environment model, ACER cannot predict that the next belief state will be the same and thus cannot foresee the unwanted repetitive conversation which leads to the failure of dialogues.

6 Conclusions

The policy optimisation algorithm presented in this paper provides a model-based augmentation and improves their performance with spoken dialogue systems (SDS). Our contributions are two-fold:

1) We adopted I2A, a model-based reinforcement learning approach, on SDS and demonstrated that it can elaborate rich information generated by environment models. 2) Our proposed algorithm further reduces instability by introducing a simple architecture to augment model-based information into policy network. We used ACER as an actor-critic model-free baseline, but this method can augment any deep actor-critic algorithm.

One interesting topic for future research is model-based actors. In our experiments, incorporating a model-based actor did not work as effectively as ADC. We plan to solve the problems inhibiting in model-based actor and make this algorithm applicable to policy learning approaches (Schulman et al., 2017; Takanobu et al., 2019).

Simulation Sample

Cambridge Restaurant Booking User Goal:
Constraints: (Area: North), (Price range: Dont Care), (Food type: Chinese)
Requests: (Address: None), (Name: None), (Description: None)
Patience: 4

	ACER agent (57% Success rate)		ADC agent (88% Success rate)
1	usr: Hello. I want to find a restaurant in Cambridge.	1	usr: Hello. I want to find a restaurant in Cambridge.
2	agt: Do you want Asian oriental food?	2	agt: Which price range do you want? expensive or moderate?
3	No, I want some Chinese food.	3	usr: I don't care.
4	agt: Ok, I will book Midsummer House Restaurant for you.	4	agt: Asian oriental or Mexican food?
5	usr: Any other choices?	5	usr: Chinese food, please.
6	agt: **The phone number of Midsummer House Restaurant is 01223 369299.**	6	agt: What else do you want?
7	usr: Ok.	7	usr: Just a Chinese restaurant in the north.
8	agt: **The postcode of Midsummer House Restaurant is CB4 1HA.**	8	agt: Do you want some expensive food?
9	usr: I want a restaurant.	9	usr: Yes, please.
10	agt: **The phone number of Midsummer House Restaurant is 01223 369299.**	10	agt: Hakka is a Chinese restaurant in the north area.
11	usr: In the north area.	11	usr: Any other choices?
12	agt: **The address of Midsummer House Restaurant is Midsummer Common.**	12	agt: The Hotpot is also a Chinese restaurant in the north area.
13	usr: I don't care.	13	usr: What is the address?
14	agt: The postcode of Midsummer House Restaurant is CB4 1HA.	14	agt: **Which price range do you want?**
15	usr: **What's the food type of Midsummer House?**	15	usr: Give me the address of The Hotpot.
16	agt: **North or centre?**	16	agt: The address of The Hotpot is 66 Chesterton Road Chesterton.
17	usr: North.	17	usr: What is the description?
18	agt: The food type of Midsummer House is British.	18	agt: **Which price range do you want?**
19	usr: I want some light bites.	19	usr: What is the description of The Hotpot?
20	agt: **North or south?**	20	agt: The Hotpot serves a variety of Chinese dishes at their restaurant and can cater for private parties of up to five five guests upon request.
21	usr: North.	21	usr: Thank you! Bye.
	.. .. Repeat turns 20, 21 till the user is out of patience. Failure		Success

Table 5: Sample dialogue sessions by ACER and ADC agents trained at epoch 500 in the environment 4 with 15% SER. The **bold sentences** are improper or repetitive responses from the agent. (agt: agent, usr: user)

Acknowledgement

Yen-Chen Wu and Bo-Hsiang Tseng are supported by Cambridge Overseas Trust and the Ministry of Education, Taiwan. Milica Gašić's contribution is part of a project that has received funding from the European Research Council (ERC) under the Grant agreement No. 804636.

References

Paweł Budzianowski, Stefan Ultes, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Inigo Casanueva, Lina Rojas-Barahona, and Milica Gašić. 2017. Subdomain modelling for dialogue management with hierarchical reinforcement learning. *arXiv preprint arXiv:1706.06210*.

Iñigo Casanueva, Paweł Budzianowski, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Stefan Ultes, Lina Rojas-Barahona, Steve Young, and Milica Gašić. 2017. A benchmarking environment for reinforcement learning based task oriented dialogue management. *arXiv preprint arXiv:1711.11023*.

Cheng Chang, Runzhe Yang, Lu Chen, Xiang Zhou, and Kai Yu. 2017. Affordable on-line dialogue policy learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2200–2209.

Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2016. Towards end-to-end reinforcement learning of dialogue agents for information access. *arXiv preprint arXiv:1609.00777*.

Mehdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman. 2016. Policy networks with two-stage training for dialogue systems. *arXiv preprint arXiv:1606.03152*.

Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1371–1374.

Milica Gašić, Filip Jurčíček, Simon Keizer, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. 2010. Gaussian processes for fast policy optimisation of pomdp-based dialogue managers. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 201–204. Association for Computational Linguistics.

Milica Gašić, Filip Jurčíček, Blaise Thomson, Kai Yu, and Steve Young. 2011. On-line policy optimisation of spoken dialogue systems via live interaction with human subjects. In *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pages 312–317. IEEE.

Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. 2016. Continuous deep q-learning with model-based acceleration. In *International Conference on Machine Learning*, pages 2829–2838.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Esther Levin, Roberto Pieraccini, and Wieland Eckert. 1997. Learning dialogue strategies within the markov decision process framework. In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 72–79. IEEE.

Jiwei Li, Alexander H Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2016. Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*.

Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. 2018. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. *arXiv preprint arXiv:1804.06512*.

Nikola Mrkšić, Diarmuid O Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2016. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777*.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. 2016. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1054–1062.

Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. 2018. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7559–7566. IEEE.

Junhyuk Oh, Satinder Singh, and Honglak Lee. 2017. Value prediction network. In *Advances in Neural Information Processing Systems*, pages 6118–6128.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Kam-Fai Wong. 2018. Integrating planning for task-completion dialogue policy learning. *arXiv preprint arXiv:1801.06176*.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152. Association for Computational Linguistics.

Jost Schatzmann and Steve Young. 2009. The hidden agenda user simulation model. *IEEE transactions on audio, speech, and language processing*, 17(4):733–747.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

David Silver, Hado van Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David Reichert, Neil Rabinowitz, Andre Barreto, et al. 2016. The predictron: End-to-end learning and planning. *arXiv preprint arXiv:1612.08810*.

Pei-Hao Su, Pawel Budzianowski, Stefan Ultes, Milica Gasic, and Steve Young. 2017. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. *arXiv preprint arXiv:1707.00130*.

Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. On-line active reward learning for policy optimisation in spoken dialogue systems. *arXiv preprint arXiv:1605.07669*.

Shang-Yu Su, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Yun-Nung Chen. 2018. Discriminative deep dyna-q: Robust planning for dialogue policy learning. *arXiv preprint arXiv:1808.09442*.

Richard S Sutton. 1990. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine Learning Proceedings 1990*, pages 216–224. Elsevier.

Richard S Sutton, Csaba Szepesvári, Alborz Geramifard, and Michael P Bowling. 2012. Dyna-style planning with linear function approximation and prioritized sweeping. *arXiv preprint arXiv:1206.3285*.

Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided dialog policy learning: Reward estimation for multi-domain task-oriented dialog. *arXiv preprint arXiv:1908.10719*.

Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel. 2016. Value iteration networks. In *Advances in Neural Information Processing Systems*, pages 2154–2162.

Blaise Thomson, Milica Gasic, Matthew Henderson, Pirros Tsiakoulis, and Steve Young. 2012. N-best error simulation for training spoken dialogue systems. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*, pages 37–42. IEEE.

Stefan Ultes, Lina M Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Inigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gasic, et al. 2017. Pydial: A multi-domain statistical dialogue system toolkit. *Proceedings of ACL 2017, System Demonstrations*, pages 73–78.

Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. 2016. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*.

Théophane Weber, Sébastien Racanière, David P Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adria Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, et al. 2017. Imagination-augmented agents for deep reinforcement learning. *arXiv preprint arXiv:1707.06203*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Jason D Williams. 2008. The best of both worlds: Unifying conventional dialog systems and pomdps. In *Ninth Annual Conference of the International Speech Communication Association*.

Yen-Chen Wu, Bo-Hsiang Tseng, and Carl Edward Rasmussen. 2020. Improving sample-efficiency in reinforcement learning for dialogue systems by using trainable-action-mask. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8024–8028. IEEE.

Yuexin Wu, Xiujun Li, Jingjing Liu, Jianfeng Gao, and Yiming Yang. 2019. Switch-based active deep dyna-q: Efficient adaptive planning for task-completion dialogue policy learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7289–7296.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Zhirui Zhang, Xiujun Li, Jianfeng Gao, and Enhong Chen. 2019. Budgeted policy learning for task-oriented dialogue systems. *arXiv preprint arXiv:1906.00499*.