Towards Causality-Aware Inferring: A Sequential Discriminative Approach for Medical Diagnosis

Junfan Lin, Keze Wang, Ziliang Chen, Xiaodan Liang, and Liang Lin

Abstract—Medical diagnosis assistant (MDA) aims to build an interactive diagnostic agent to sequentially inquire about symptoms for discriminating diseases. However, since the dialogue records for building a patient simulator are collected passively, the collected records might be deteriorated by some task-unrelated biases, such as the preference of the collectors. These biases might hinder the diagnostic agent to capture transportable knowledge from the simulator. This work identifies and resolves two representative non-causal biases, i.e., (i) default-answer bias and (ii) distributional inquiry bias. Specifically, Bias (i) originates from the patient simulator which tries to answer the unrecorded inquiries with some biased default answers. To eliminate this bias and improve upon a well-known causal inference technique, i.e., propensity score matching, we propose a novel propensity latent matching in building a patient simulator to effectively answer unrecorded inquiries; Bias (ii) inherently comes along with the passively collected data that the agent might learn by remembering what to inquire within the training data while not able to generalize to the out-of-distribution cases. To this end, we propose a progressive assurance agent, which includes the dual processes accounting for symptom inquiry and disease diagnosis respectively. The diagnosis process pictures the patient mentally and probabilistically by intervention to eliminate the effect of the inquiry behavior. And the inquiry process is driven by the diagnosis process to inquire about symptoms to enhance the diagnostic confidence which alters as the patient distribution changes. In this cooperative manner, our proposed agent can improve upon the out-of-distribution generalization significantly. Extensive experiments demonstrate that our framework achieves new state-of-the-art performance and possesses the advantage of transportability.

Index Terms—Causal inference, reinforcement learning, decision making, medical diagnosis assistant.

I. INTRODUCTION

A MEDICAL diagnosis assistant (MDA) aims to learn an active agent from passively collected data, which can be used to collect symptom information and make preliminary diagnoses. Specifically, the MDA agent sequentially inquires about symptom information from the user/patient, and proactively terminates the interaction by informing the discriminant diagnosis. Such sequential discrimination has been formulated as a Markov decision process and is resolved by reinforcement learning (RL) [1], [2], [3], [4]. However, we found that the previous RL-based MDA methods [3], [4] have omitted several causal biases in building the patient simulator and designing the diagnostic agent. Learning MDA agents without mitigating these biases would hamper the agent to discover the causal and transportable skills of interest behind the data. According to the sources of these biases, we denote them as default-answer bias and distributional inquiry bias, respectively.

Default-Answer Bias: In prior methods [3], [4], the patient simulator will choose a dialogue diagnosis record from the collected dataset as the anchor record (as exemplified in the top of Fig. 1), and then answer the inquiry from the MDA agent by looking the answer up in the anchor record. However, as the passive record only reflects the factual side of the world, the simulator might fail to answer the unrecorded inquiries (i.e., the counterfactual aspects) from the agent, illustrated in Fig. 1. To deal with the counterfactual inquiries, prior works [3], [4] make the simulator render ‘not sure’ responses as the default answers. Unfortunately, the default-answer strategy will bring about the collider/selection bias [5] among the symptom, inquiry, and answer, named default-answer bias in our paper. As depicted in Fig. 2, by convention, the answer/observation is the result/collider of the inquiry from the agent and symptom of the patient. As shown in the dashed box of Fig. 2, controlling the observation to be ‘not sure’ will result in a biased association that the status of the symptom is dependent on the status of the inquiry. This bias hampers the agent from learning the causal relationship between symptoms and diseases. Worse, due to the sparsity of the symptom information in MDA, the simulator will be frequently inquired about unrecorded symptoms during the training phase, amplifying the collider bias’s negative impact. Training and evaluating under the simulator with severe collider bias cannot fully reflect the advantages of the diagnostic agent.
is
Propensity Latent Matching
simulated patient), to
Propensity Score Matching
(PLM), which
in the figure) by
from the potential outcome framework (PO). In our
answer the
symptom inquiries
its self-report to start a diagnosis process. Then, the simulator answers the
counterfactual symptom inquiries about the unobserved symptoms.

Fig. 1. The patient simulator chooses a record as the anchor record and gives
its self-report to start a diagnosis process. Then, the simulator answers the factual
symptom inquiries which are already observed in the anchor record but would fail
to answer the counterfactual symptom inquiries about the unobserved symptoms.

To reduce the default-answer bias, the simulator should infer the symptom ignoring the inquiry and render the answer accordingly rather than providing the default answer to the counterfactual inquiry. In the field of causal inference, one of the most popular counterfactual inference methods to provide such ‘ignorability’ [6], [7] is Propensity Score Matching (PSM) [8], [9], [10] from the potential outcome framework (PO). In our context, PSM is also suitable to address the sparsity problem by abstracting the sparse covariates to low-dimensional propensity scores via a propensity score estimator. However, during the interactive learning process of the MAD agent, counterfactual inquiries are frequently raised. The counterfactual inquiry might be out-of-distribution to the propensity score estimator, resulting in unreliable propensity score estimation. To this end, our paper proposes a novel Propensity Latent Matching (PLM), which conducts matching in the latent space. Unlike PSM which predicts the propensity score after seeing the counterfactual inquiry,

Fig. 2. The causal diagrams among the symptom, observation, and inquiry. Causally (boxed by solid line), the status of the symptom is independent of the status of the inquiry. When the observation is controlled as “not sure” (boxed by dashed line), the status of the symptom is biased and becomes dependent on the status of the inquiry.

Fig. 3. The causal diagrams for the diagnostic agent. 1) The distributional inquiry bias misleads the agent to stop inquiry with insufficient symptom observation; 2) Furthermore, insufficient observations might cause the diagnoser to overfit the training distribution.

our PLM models the latent features before the counterfactual inquiry gets involved in estimating the propensity, which makes the matching more reliable.

Distributional Inquiry Bias: Another representative bias that comes along with passive data is the distributional bias, which is also one of the key obstacles for training the agent towards “learning how” rather than “remembering what”. Due to the difficulty and high cost of collecting MDA data, the diagnostic agent suffers from a limited amount of training data and might be fragile in out-of-distributional cases. For example, if inquiring about the symptom ‘coughing’ is enough for distinguishing between two diseases among the training samples, the agent tends to learn a ‘shortcut’ that only inquires about ‘coughing’ which also hampers the learning of discrimination. We name such kind of bias as the distributional inquiry bias. As shown in Fig. 3, the distributional inquiry bias not only affects the inquiry process (i.e., the effect on ‘stop inquiring’) but also affects the learning of diagnosis mediated by the ‘non-inquired observations’. In this manner, at the deployment phase where more symptom inquiries are needed, the agent might be unable to collect enough information and make reliable diagnoses.

To mitigate these issues, we propose a novel MDA agent, progressive assurance agent (P2A), which leverages the confidence of diagnosis to modulate the inquiry process and conducts the intervention to improve the robustness of the diagnosis. In specific, P2A is a dual-process agent, consisting of two separate yet cooperative branches, i.e., the diagnosis branch and the inquiry branch. The diagnosis branch takes advantage of the intervention/do-calculus [11], i.e., do(Causes of non-inquired observations = simulated patient), to cut off the effect path from distributional inquiry bias to the diagnosis. And the inquiry branch is modulated by the diagnosis branch to collect evidence to enhance the diagnostic confidence until the decision threshold is met. The diagnostic confidence can sensitively reflect the distribution changes and thus adapt the inquiry branch to out-of-distributional scenarios. Taking a binary-alternatives case as an example (Fig. 4), the diagnosis branch first reasons and plans according to historical observations (namely, \(s_t\) in the figure) by intervening in the unknown aspects, i.e., the non-inquired observations, of the patient. By intervention, the diagnostic decision in our P2A is made according to the more comprehensive and imaginary patient information instead of the observed information limited to the inquiry behavior. Then, the imagined patients are diagnosed into different disease clusters (in either orange or green). Once the population of one cluster is significantly
overwhelming the other by a decision threshold [12], the agent will stop inquiring and the corresponding disease of that cluster will be informed. Particularly, the competitive behavior between different clusters and the concept of the decision threshold of our P2A are all derived from the diffusion model in neuroscience [12]. Not limited to the binary case in neuroscience, our P2A has extended the diffusion model to a multi-alternatives setting from a computational perspective.

Overall, the main contributions of our paper are tri-fold: i) We identify the default-answer bias that existed in the previous patient simulator and propose a propensity-based patient simulator (PBPS) using the novel propensity latent matching; ii) we also identify the distributional inquiry bias and propose a novel MDA agent, progressive assurance agent, to eliminate the distributional bias via intervention; iii) we develop a neuro-inspired ‘decision threshold’ mechanism for addressing the sequential discrimination problems, which seamlessly connects the inquiry and discrimination to make reliable and interpretable decisions. Considering the addressed causality issues, we name our framework, consisting of PBPS and P2A, as Causality-Aware MDA (CA-MDA). Experimental results demonstrate that: i) our PBPS is superior in answering counterfactual symptom inquiry and generating more informative answers; ii) our P2A advances in capturing the symptom-disease relationship and generalizing it to out-of-distribution cases, and also possesses the advantage of sample-efficient and robustness.

The remainder of this paper is organized as follows. Section II comprehensively reviews the related works. Section III presents the background of CA-MDA and the limitation of RL-based formulation. Section IV elaborates on our CA-MAD and Section V shows the experimental results and human evaluation of two public MDA benchmarks with comprehensive evaluation protocols. Section VI concludes this paper.

II. RELATED WORK

In the field of causal inference, there are many different methods proposed in recent years, which can be approximately categorized into two branches according to whether the causal relationship is modeled explicitly or implicitly, namely, the structural causal model (SCM) framework [13] and the potential outcome framework [6]. The methods based on the SCM usually incorporate a causal diagram (CD) and structural causal equations [14]. Then, the collected data are injected into the model to infer the causal effect. Recently, many approaches based on the CD have been proposed in computer vision [15], [16], [17], [18], [19], reinforcement learning [20], [21], [22], [23], etc. Most of these works are based on the general CD for the population. In MDA, patients with the same diseases may have different symptoms due to individual differences. As for modeling the individual-level relation, the model-free causal inference framework (i.e., the potential outcome framework [6] (PO)) is widely adopted. PO infers the causal effect from passive observational data without an explicit causal model and it’s also capable of handling individual causal effects.

In MDA, the collected dialogue records are passively observed, that’s, the existence of the unobserved symptom is missing [24]. To fill in the missing answer, current works [3], [4] proposed to build patient simulators on the collected data. In specificity, they adopt patient-doctor conversation records [25], [26] to generate responses, and use the default answer to respond with the counterfactual inquiry, which will introduce unexpected collider/selection biases. The collider/selection biases are introduced when there are controlled colliders [5]. These biases usually cannot be solved by boosting with more data [27]. The most straightforward solution is to develop a new simulator that does not respond with the counterfactual inquiry using the default answer but sampling the answer from the authentic distribution. To model the answer distribution from the collected data, [28] incorporated a parametric model of the environment into the dialogue agent to generate a simulated user experience, which however is prone to overfit the characteristics of the training data [29]. Instead of direct estimation, matching techniques have been shown to be better at producing less biased and more causality-consistent estimation [30]. In MDA, the collected data are usually very sparse therefore ordinary matching methods are usually infeasible due to the problem of lack of overlap. To resolve this problem, propensity score matching (PSM) [8], [10] is introduced to estimate the low-dimension propensity of the instance for matching. However, PSM requires a reliable propensity score estimator to perform well. In the sequential interaction process, the covariates will be affected by the previous unfamiliar action which might degrade the reliability of the estimated propensity score. Our PLM is not the first method to introduce latent features for improving propensity score matching [31], which limits the measurement error of propensity score. Different from prior works, the latent features in PLM are not given at the beginning and are learned through self-supervised...
learning. Moreover, PLM directly matches the latent features since the propensity score is affected by the inquiry. Similar to our PLM, a recent study in self-supervised learning, masked auto-encoder (MAE) [32] also observe that latent learning generally has better generalization ability. Differently, our PLM is optimized to model the action distribution while MAE is optimized for general purposes and requires extra task-specific tuning for downstream applications.

As for the task-oriented dialogue agent, most of the current task-oriented dialogue systems adopt the framework of reinforcement learning (RL) [1], [33], [34], and some works [35], [36], [37] adopt the sequence-to-sequence style for dialogue generation. For medical dialogue systems, due to a large number of symptoms, reinforcement learning is a better choice for topic selection [38], [39], [40]. In the context of MDA, the actions of symptom inquiry and disease diagnosis are discrete. Therefore, most of the current MDA methods exploit the classical discrete control methods, e.g., Deep Q-Network (DQN) [1] to select the actions [38] applied DQN to diagnose using synthetic data. While [4] first did experiments on real-world data using DQN. To include explicit medical inductive bias for improving the diagnostic performance, KR-DQN [3] proposed an end-to-end model guided by symptom-disease knowledge priors. KR-DQN applies the predefined conditional probability of symptom and disease to transform the estimated Q-values.

Differently, our work follows the logic of MDA in real life and formulates it as a sequential discriminative decision-making problem. To model the decision-making process, neuroeconomics proposes a diffusion model [41]. Inspired by these discoveries, our proposed dual-process P2A also incorporates the concept of decision threshold for the diagnoser to determine when to inform disease. Like ours, PG-MI-GAN [42] adopts a separate diagnoser from the inquiry policy. Specifically, it trains an inquiry generator to generate a sequence of inquiries undissected by the discriminator and further uses a pretrained diagnoser to finetune the generator. As we discovered, the design of the diagnoser needs to take the indirect bias [43], distributional bias [44], [45], [46], [47] as well as training bias [48] into account to better capture the symptom-disease relationship and improve the transportability [49] of the diagnosis agent.

Moreover, as a medical application, an MDA agent is also required to consider the uncertainty of its decision to provide a robust and trustful diagnosis result for the ethics concern. Most of the current MDA agent is allowed to jump into informing a disease without any regulation. Instead, our MDA agent takes the uncertainty to augment the decision-making process. There are plenty of works studying how to combine uncertainty and exploration [48], [50], [51], [52], [53], [54]. However, different from ours, these works do not employ uncertainty to provide a stop mechanism for sequential decision-making.

III. PRELIMINARIES

Medical Diagnosis Assistant (MDA): As shown in Fig. 1, a diagnosis record consists of a self-report, symptom statuses \( y = [y_0, y_1, \ldots, y_{N_y}] \) (1 for ‘no’, 0 for ‘not sure’, 1 for ‘yes’), and a ground-truth disease label \( d \in [1, \ldots, N_d] \), where \( N_y \) is the number of symptoms and \( N_d \) is the number of diseases. An MDA diagnosis process starts with a patient’s self-report, which forms the initial dialogue state, \( s_{t=0} \in S \), where \( S \subseteq \mathbb{R}^{N_y} \) denotes the state space and \( s_t \) maintains the values of all mentioned symptoms (i.e., -1 for ‘no’, 0 for ‘not sure’ and 1 for ‘yes’) up to timestep \( t \). And \( A = [1, \ldots, N_y, N_y + 1, \ldots, N_y + N_d] \) represents the action space of the agent. The MDA agent selects an action \( a_t \in A \), to either inquire about symptoms \( (a_t \leq N_y) \) or inform the diagnostic result \( (a_t > N_y) \). And since the interaction ends when the discriminative action is made, we use \( T \) to denote the timestep when the diagnosis ends and \( \Delta_T \) stands for the diagnosis action/detection and \( \Delta_T \) is the final state of the diagnosing process. The patient simulator updates the dialogue state by \( \mathcal{P} : S \times A \rightarrow S \). Practically, the patient simulator will return a response/observation \( o_t \) to the inquiry \( a_t \), and then update the dialogue state according to \( o_t \). Although the observation \( o_t \) is assumed to have the same value as the symptom status \( y_{a_t} \) in MDA, we will use them differently according to the contexts. To indicate whether a symptom has been inquired about, we use a binary vector \( m_t \) to record the visited inquiries up to time \( t \), where \( m_{t,a_{t+1}} = 1 \). MDA is a sequential discriminative problem that sequentially inquires about symptoms and makes a discriminative diagnosis at the end. Formally, MDA is to search for an agent \( \pi : S \rightarrow A \) that:

\[
\max_{\pi} \mathbb{E}_{P,\pi} \left[ -\max(T, T^*) \cdot T + (\mathbb{I}(o_T = N_y + d) + 1) \right],
\]

where \( T^* \) stands for the optimal number of inquiries for patient \( P \). For different patients, \( T^* \) varies as the inquiries required for the diagnosis varies. \( \mathbb{I}(\cdot) \) is an indicator function that returns 1 if the propositional logic formula in the bracket is satisfied, otherwise returns 0. The optimality of (1) is obtained when the inquiry process collects just enough symptom information for an accurate diagnosis. To this end, there exists a subordinate relationship among the actions, i.e., inquiry serves for discrimination.

Reinforcement Learning (RL) for MDA: To apply RL for resolving MDA, previous methods [3], [4] design complex reward functions, i.e., \( R : S \times A \rightarrow \mathbb{R} \) for different inquiry/diagnosis actions. The target of reinforcement learning is to solve via a policy form \( \pi : S \rightarrow A \), which maximizes the expected sum of rewards:

\[
\max_{\pi} \mathbb{E}_{\pi} \left[ \sum_{t=0}^{T} \gamma^t R(s_t, a_t) \right],
\]

where \( \gamma \in [0, 1) \) is a discount factor. However, since all actions are abstracted into numerical rewards and connected by accumulation in (2), the subordinate relationship between inquiry and diagnosis cannot be modeled explicitly. To mitigate this issue, previous methods assign significantly larger rewards/punishments for correct/incorrect diagnoses than inquiries. This strategy is based on the numerical association between inquiry and diagnosis, instead of the causal relationship. Inference based on association usually requires a dedicated design for reward functions to balance the importance of different actions. And it would also cause some unexpected behavior when handling
IV. CAUSALITY-AWARE MDA

To mitigate the two identified biases that exist in previous RL-based methods, i.e., the default-answer bias and the distributional inquiry bias, we propose our CA-MDA, which is comprised of the propensity-based patient simulator (PBPS) and the progressive assurance agent (P2A). Moreover, P2A leverages a neuro-inspired decision threshold to establish the subordinate relationship between inquiry and diagnosis explicitly. The overview of the interaction between PBPS and P2A is shown in Fig. 5.

A. Propensity-Based Patient Simulator

As shown in Fig. 6, a simulator is required to generate a counterfactual response $o_t$ to a counterfactual inquiry $a_t$ according to the simulator state $z_t = (y_t, d)$. Different from the dialogue state $s_t$, simulator state $z_t$ includes the symptom statuses $y_t$ and the disease label $d$. And $y_{t=0}$ is the original symptom existences $y$ of the anchor record. $y_t$ is updated after inferring an unrecorded symptom existence. To causally predict the answer $o_t$, one of the most popular causal inference algorithms is propensity score matching (PSM). Specifically, PSM conducts matching based on the sample’s propensity score $e_t = P(O_t | z_t, a_t)$. After matching, PSM refers to the observation from the matched samples as a potential answer. By drawing the answers from matched samples, the generated answers of PSM are more authentic and less biased. However, the propensity score is conditioned on the inquiry as shown in Fig. 6. When $a_t$ is a counterfactual inquiry, the combination of $(z_t, o_t)$ might be unfamiliar to the propensity estimators and cause the estimation unreliable. As the interaction goes on, the error will be accumulated, forming a vicious spiral.

Propensity Latent Matching: To this end, we propose a novel matching strategy, Propensity Latent Matching (PLM). Different from the propensity score, our method takes a step back and matches with the propensity of latent features $f_t = f(z_t)$ before the inquiry takes effect, where

$$e_t = P(O_t | z_t, a_t) = P(O_t | f(z_t), a_t).$$  \hfill (3)

Equation (3) shows that PLM is mathematically equivalent to PSM. The relationship between $f_t$ and $e_t$ is shown in Fig. 6. From the illustration, we can see that the benefit of PLM is that $f_t$ is not affected by the counterfactual inquiry $a_t$, and thus the counterfactual inference is more reliable. And if $a_t$ is still unobserved among the matched records, $o_t$ will be 0, and $z_{t+1}$ remains the same as $z_t$. This prevents the simulator state from becoming out-of-distribution during the interaction process.

In building PBPS, we use multilayer perceptron (MLP) $f_{\phi_L} (\cdot)$ to model propensity latent features $f_t$, where $\phi_L$ denotes the parameter of the network. And we use another MLP $e_{\phi_0} (f_{\phi_L} (\cdot), a)$ parameterized by $\phi_S$ to model the propensity score $e_t$, satisfying (3). To optimize $\phi_L$ and $\phi_S$, we deploy the self-supervised strategy. Specifically, we treat the recorded symptom existence $y$ as the $y_T$. To obtain the immediate simulator state (e.g., $z_t$ in Fig. 6), we mask off some of $y$ according to the visitation mask $m_t$. The mask is used to mask off the symptoms that are mentioned after timestep $t$ in the diagnosis record, resulting in the immediate state $z_t = (y \odot m_t, d)$. To this end, $e_{\phi_0} (f_{\phi_L} (\cdot), a)$ can be learned via minimizing the cross entropy loss CE:

$$\min_{\phi_L, \phi_0} E_{y, d, m, a} [\text{CE} (e_{\phi_0} (f_{\phi_L} (y \odot m_t, d), a), y_a)],$$  \hfill (4)
Algorithm 1: Propensity-Based Patient Simulator (PBPS): \( P_{PBPS}(s_t, a_t; p) \).

**Input:** \( s_t, a_t \), simulator state \( z_t \) and the self-report

**Output:** \( s_{t+1} \)

1: if \( t = -1 \) then
2: Initialize the dialogue state \( s_0 \rightarrow [0, \ldots, 0] \) and the visitation indicator \( m_0 \rightarrow [0, \ldots, 0] \) with the size of \( N_s \), and initialize the simulator state \( z_0 \rightarrow (y, d) \)
3: Let \( s_{0,a} \leftarrow y_a \) and \( m_{0,a} \leftarrow 1 \) for each \( a \) parsed from the self-report
4: Return \( s_0 \)
5: else
6: Initialize\(a \leftarrow a_t, s_{t+1} \leftarrow s_t, m_{t+1} \leftarrow m_t, z_{t+1} \leftarrow z_t, q' \leftarrow P \)
7: if \( (\{y_a = 0\} \land \{m_{t+1,a} = 0\}) \), sample \( q' \sim P(q|f_t, a) \) according to (5).
8: Generate response \( o_t \leftarrow (q') \)
9: Update dialogue state \( s_{t+1} \) by \( s_{t+1,a} \leftarrow o_t \)
10: Update visitation history \( m_{t+1} \) by \( m_{t+1,a} \leftarrow 1 \)
11: Update simulator state \( z_{t+1} \) by \( y_{t+1,a} \leftarrow o_t \)
12: Return \( s_{t+1} \)

where \( a \) is the index of one of the masked symptoms and \( y_a \neq 0 \) is the symptom existence of the \( a \)-th symptom. After training, the output of \( f_{a_t}(\cdot) \) is the propensity latent features for matching.

During inference, for a counterfactual inquiry \( a_t \), the simulator explores all records \( \{q_i\}_{i=1}^N \) that have the same disease (i.e., \( d(q) = d \)) and also have \( a_t \) in their observed symptoms (i.e., \( y_{a_t} \neq 0 \)). The similarity weight of a record \( q \) is formulated as

\[
P(q|f_t, a_t) \propto I \left((d(q) = d) \land (y_{a_t} = 0)\right) \times e^{-\|f_t - f(q)\|^2/\sigma^2},
\]

where \( f(q) = f_{a_t}(y_a \circ m_t, d(q)) \) and \( e^{-\|f_t - f(q)\|^2/\sigma^2} \) is a bell-shaped function normalizing the similarity into probability (\( \sigma \) indicates a standard deviation). Here, \( m_t \) indicates the visited symptoms of \( y_t \). The similarity of propensity latent features of the patient records implies that the existences of symptoms of these records are more probably similar. Then the patient simulator can sample a record according to the similarity weight, i.e., \( q' \sim P(q|f_t, a_t) \). And the symptom existence \( y_{a_t}^{(q')} \) of the sample record is used as the answer to the inquiry \( a_t \). The state transition of PBPS with anchor record \( p \), i.e., \( s_{t+1} = P_{PBPS}(s_t, a_t; p) \), is presented in Algorithm 1, where line 7 (highlighted in bold and italics) is the extra operation of our simulator compared to the previous simulator \( P_{PS} \).

**B. Progressive Assurance Agent**

Our Progressive Assurance Agent (P2A) consists of two separate yet cooperative branches for symptom inquiry and disease diagnosis, as shown in Fig. 5. The inquiry branch inquires about symptoms to get \( s_t \) from PBPS to increase diagnosis confidence, while the diagnosis branch imagines and reasons the future scenarios to robustly estimate the disease and its confidence per step until the confidence is high enough (satisfying Decision Threshold (DT) [12]) to inform a disease. In this manner, if the patient is unfamiliar/out-of-distribution to the agent, the decision threshold will modulate the inquiry branch to adapt to the unfamiliar case. Formally, the overall optimization target of P2A is:

\[
\max_{\pi_s, \phi} \mathbb{E}_{\rho, \pi_s} \left[ \sum_{t=0}^{\text{max}(T, T_{DT})-1} \gamma^t r + I(f_{\phi}(s_{T_{DT}}) = d) \right],
\]

where \( T_{DT} \) stands for the minimal number of inquiries for reaching the decision threshold (DT), \( r \) is a negative constant, and \( s_{T_{DT}} \) stands for the state when the decision threshold is met. \( \pi_s \) is the inquiry policy parameterized by \( \theta \) and \( f_{\phi} \) denotes the diagnoser parameterized by \( \phi \). Different from the RL target in (2), our P2A explicitly models the subordinate relationship between inquiry and diagnosis via the decision threshold as formulated in (6). Moreover, after introducing the decision threshold, the inquiry policy and the diagnostic agent can be optimized separately. The left-hand side is the optimization target for the inquiry branch, i.e., to reach the decision threshold as soon as possible. The right-hand side is the target for the diagnosis branch, i.e., to accurately discriminate disease when the decision threshold is met. In the following, we will elaborate on the formulation of the diagnosis branch and inquiry branch. The algorithm pipeline of P2A is illustrated in Algorithm 2.

**Diagnosis Branch:** To eliminate the distributional inquiry bias on diagnosis, our P2A intervenes in the “non-inquired observation” to be independent of the inquiry behavior. Specifically, our P2A first infers the possible final state \( \hat{s}_T \) (i.e., the state at the end of the diagnosing process) according to the inferred observation \( s_t^v \), i.e., \( P(\hat{s}_T|s_t^v) \). Then, our P2A intervenes in the observations of the non-inquired symptoms \( s_t^v \) with the estimated simulated patient, i.e., \( d(s_t^u) = \hat{s}_T - s_t^v \). One can intuitively treat such intervention processes as “imagining and reasoning the future interactions”. After that, our P2A feeds the inferred and intervened symptoms as input to the discriminative diagnoser \( P(d|\hat{s}_T) \). In total, the diagnoser of our P2A diagnoses by using the observed symptoms via \( P(d|s_t^u) = \sum_{s_t^v} P(d|\hat{s}_T)P(s_t^u|s_t^v) \). We train the intervenor \( f_{\phi_G}(\cdot) \) and the discriminative diagnoser \( f_{\phi_c}(\cdot) \) for \( P(\hat{s}_T|s_t^v) \) and \( P(d|\hat{s}_T) \), respectively.

**Intervener aims at predicting the final symptom state \( \hat{s}_T \) from the current inferred state \( s_t^u \). Therefore, we model it as a generative problem. \( \phi_G \) aims at predicting the symptom mask indicating which symptoms are observed up to time \( t \), \( a \) is the index of a masked symptom, i.e., \( m_{t,a} = 0 \) \( m_{t,a} = 1 \). The target for \( f_{\phi_G} \) is to recover the masked information. And the final state \( s_T \) is sampled from the data replay buffer \( D_{\tau} \) which stores the final state during the training process (line 16 in Algorithm 2). As shown in the overview (Fig. 5), the Monte Carlo sampling is applied by obeying the generative model \( f_{\phi_G}(s_t|s_t^u, m_t) \) to sample \( K \) possible final states \( s_T^{(k)} \) \( k=1 \). Note that, the inferred symptoms of \( s_t^v \) remain the same in these final states. After the intervention, the
Algorithm 2: Progressive Assurance Agent (P2A).

Input: Initial inquiry policy parameters \( \theta \), intervenor parameters \( \phi_C \), B bootstrapping diagnosers parameters \( \phi_B \), empty replay buffer \( D_Q \) and final state buffer \( D_C \).

for each episode do
  2: Get one anchor record \( p \) and initialize the state \( s_0 \)
  for \( t = 0 : T_{max} \) do
    4: Use the intervenor to generate \( K \) final states:
       \( \{ s^{(k)}_{t} \} \sim f_{\phi_C}(s_{t}) \), \( k \in [1, K] \)
    Feed the final states to \( B \) bootstrapping diagnosers:
       \( \mathbb{P}_t = \{ p^{(k,b)}_{t} \}, \forall k \in [1, K], b \in [1, B] \}
    6: Calculate the statistics \( \mu_t, \sigma_t \) of \( \mathbb{P}_t \)
    if \( \text{DT}(\mu_t, \sigma_t) \) then \( \triangleright \) Meet DT or time out
    8: Inform disease: \( a_t = \max_i \mu^{(i)}_{t} + N_s \) if 10% chance then \( \triangleright \) \varepsilon\-greedy explore
    else \( a_t \) inquire about a random unobserved symptom
    \( \triangleright \) Use inquiry policy to select inquiry
    12: \( a_t = \max_i Q_{\phi}(s_t, \mu_t, a) \)
    Interact with \( \mathcal{P}_{\mathcal{PBS}}: s_{t+1} = \mathcal{P}_{\mathcal{PBS}}(s_t, a_t; p_t) \)
    14: Store transition \( (s_{t-1}, \mu_{t-1}, a_{t-1}, s_t, \mu_t) \) in \( \mathbb{P}_Q \)
    if \( a_t > N_s \) then \( \triangleright \) Action of informing disease
    Store records \( (s_{t-1}, d(p_t)) \) in \( \mathbb{P}_C \)
    break
  16: if time to update then
    Sample mini-batch from \( \mathbb{P}_Q \) and update \( \theta \) (12) and (13).
  20: Sample \( B \) mini-batches from \( \mathbb{P}_C \) with replacement to update \( \phi_C \) and \( \phi_B \) (7) and (8).

\( K \) imaginary final states are fed to the diagnoser \( f_{\phi_B}(\cdot) \) to check whether the decision threshold is met or not.

Decision Threshold: Intuitively, doctors stop inquiring to inform diseases when they are confident that inquiring about more symptoms would not overturn his diagnosis. Therefore, we propose the decision threshold (DT) to mimic such an introspective process, that is, the agent would stop inquiring to inform the preferred disease if the agent believes that the probability of the preferred disease is high enough so that inquiring more symptoms would not overturn the preferred disease probabilistically. To estimate the probability of each disease and its confidence, bootstrapping technique [48], [50], [51] is adopted to train ensembles of diagnosers.

Bootstrapping diagnosers are trained to diagnose using the final state \( s_T \) stored in the data replay buffer. The target of diagnoser \( i \) with parameter \( \phi_{B,i} \) is

\[
\min_{\phi_{B,i}} \mathbb{E}_{(s_T,d) \sim D_C} \left[ \text{CE}(f_{\phi_{B,i}}(s_T), d) \right].
\]  

Note that, since P2A terminates the interaction when the decision threshold is met, \( s_T \) can be regarded as the same as \( s_{t_{\text{DT}}} \). In this sense, (8) is optimized for the right-hand side of (6).

During inference, the final states sampled from the intervenor are fed into \( B \) bootstrapping diagnosers, resulting in a final disease probability set \( \{ p^{(k,b)}_{t} \}_{k=1, b=1} \). The final disease probability set is then used to calculate the expectation \( \mu_t = [\mu^{(1)}_{t}, \ldots, \mu^{(N_a)}_{t}] \) and standard deviation \( \sigma_t = [\sigma^{(1)}_{t}, \ldots, \sigma^{(N_a)}_{t}] \) of diseases:

\[
\mu_t = \frac{1}{KB} \sum_{b=1}^{B} \sum_{k=1}^{K} p^{(k,b)}_{t}, \quad \sigma^2_t = \frac{1}{KB} \sum_{b=1}^{B} \sum_{k=1}^{K} (p^{(k,b)}_{t} - \mu_t)^2,
\]

which are further used to calculate the confidence intervals of diseases. Moreover, bootstrapping diagnosers are also popular in reducing the unexpected noisy effect introduced by the data sampling process and parameter initialization. This kind of noisy effect would hamper the model’s in-distributional performance [48].

With the mean and standard deviation, DT would be met if the probability of the preferred disease is beyond the upper bound of the 6\( \sigma \) confidence interval [55], [56] of the other diseases’ probabilities. Denote the preferred disease as \( i \), i.e., \( i = \text{arg}\max_j \mu^{(j)}_{t}, \forall j \in [1, N_a] \). DT is formulated as

\[
\text{DT}(\mu_t, \sigma_t) = \begin{cases} \text{True}, & \forall j \neq i, \mu^{(i)}_{t} > \mu^{(j)}_{t} + 3\sigma^{(i)}_{t}, \\ \text{False}, & \text{otherwise} \end{cases}
\]

Inquiry Branch: The left-hand side of (6) is the problem of finding the shortest path to reach the decision threshold, which can be well-addressed by reinforcement learning. Therefore, we apply deep Q-learning [1], [57] to optimize the inquiry policy. Specifically, the inquiry branch is modeled by a Q network [57] parameterized by \( \theta \), which takes the concatenation of the state \( s_t \) and the current disease probabilities \( u_t \) to predict the inquiry action \( a_t \in [1, N_a] \).

\[
a_t = \max_i Q_{\phi}(s_t, u_t). \]

The optimization target of inquiry policy is:

\[
\min_{\theta} \mathbb{E}_{\theta} \left[ \left[ (r + \gamma \max_i Q_{\theta_{\text{arg}}}(s_{t+1}, u_{t+1}, a) - Q_{\theta}(s_t, u_t, a_t))^2 \right] \right],
\]

where \( r = -0.1 \) is a negative constant for punishing redundant inquiries. And the training data are sampled from the replay buffer \( D_Q \) (line 14 in Algorithm 2). The parameter \( \theta_{\text{targ}} \) is used for stabilizing the training, updated with a momentum factor \( \alpha \):

\[
\theta_{\text{targ}} = \alpha \theta_{\text{targ}} + (1 - \alpha) \theta.
\]

V. Experiments

To justify the effectiveness of our CA-MDA, we mainly focus on answering the following questions: 1) Is the propensity-based patient simulator better at answering counterfactual symptom inquiries? (Reducing default-answer bias) 2) Does the progressive assurance agent achieve better in-distribution and out-of-distribution diagnostic performance? (Reducing distributional inquiry bias) 3) Does the decision threshold mechanism bring about a more robust diagnosing process? More converging curves, hyperparameter analysis, and extended experiments are presented in Appendix C, available online. For better understanding, we also provide numerical running examples in Appendix D, available online.
**Dataset:** We perform extensive and comprehensive evaluations on two MDA benchmarks, i.e., MuZhi (MZ) [4] composed of 586 training and 142 test records with 66 symptoms and 4 diseases; DingXiang (DX) [3] composed of 423 training and 104 test records with 41 symptoms and 5 diseases. More details about the benchmarks are placed in Appendix C, available online. \(P_{\text{train}}\) and \(P_{\text{all}}\) denote the patient simulators organized by the training records and all records (for training and testing) respectively. For instance, \(P_{\text{train}}\) represents the PBPS using training records in a benchmark to interact with the agent.

**Baselines:** To answer Question 1), we have compared our PBPS with the previous patient simulator (PS) [3], [4]. To our knowledge, all prior MDA works did not propose new patient simulators but used PS with the default-answer strategy. As for the direct estimator, we adopt a generative world model (GEN) [28]. PBPS-PSM is our propensity-based simulator using propensity score matching and PBPS uses our propensity latent matching. As for Question 2), we have compared our P2A against three baselines using our PBPS, i.e., DQN [4], KR-DQN [3], and PG-MI-GAN [42]. DQN combines symptom inquiry and disease diagnosis into a single policy network and trains it by deep Q-learning [1], [58]. Improved from DQN, KR-DQN [3] adds a knowledge-routing module at the head of the policy using predefined disease-symptom knowledge, i.e., matrices of conditional/joint probability. Distinguished from DQN and KR-DQN, our P2A disentangles the disease diagnosis from the policy. Similar to ours, PG-MI-GAN [42] adopts a separate diagnoser from the inquiry policy. Specifically, it trains an inquiry generator to generate a sequence of inquiries that are hard for the discriminator to distinguish, and further uses a pretrained diagnoser to finetune the generator. Its inquiry generator is pretrained through imitation, which lacks reasoning between inquiries and diseases. Moreover, its diagnoser is fixed during the finetuning process and does not evolve with the inquiry generator. To testify to the robustness and safeness of the decision threshold mechanism (i.e., Question 3), we examine whether the accuracy of P2A can be improved significantly when the decision threshold is met during diagnosing under both In. and Out. settings.

**A. Evaluation on PBPS**

To evaluate how accurately a simulator can answer the counterfactual symptom inquiries, we propose a novel causal metric, named Inferential Ability (IA).

**Inferential Ability:** Factually, there is no ground truth for the counterfactual symptom inquiries. In MDA, fortunately, we know what should have been observed if the queried symptom was not inquired about, that’s, ‘not sure’, as shown in Fig. 2. This allows us to convert some recorded symptoms to unrecorded and relabel their observation as ‘not sure’ to synthesize new records. Building simulators with the relabeled records, we can examine how the simulators answer the counterfactual symptom inquiries by inquiring about the relabeled symptom inquiries. After that, we can use the original observations to compare with the answers from the simulators.

| Benchmarks | PS [3], [4] | GEN [28] | PBPS-PSM | PBPS* | PBPS |
|------------|-------------|----------|----------|-------|-------|
| MZ         | 0.0±0.0     | 0.12±0.01| 0.23±0.03| 0.57±0.02| 0.62±0.02|
| DX         | 0.0±0.0     | 0.14±0.03| 0.27±0.03| 0.58±0.03| 0.66±0.01|

Practically, to measure the IA of the patient simulators, we relabel all implicit symptoms from the records and only provide the record \(p_{\text{ex}}\) with explicit symptoms and disease to the patient simulators. After that, we inquire about the relabeled implicit symptom of different patient simulators and calculate the accuracy of their responses. Formally,

\[
IA = \frac{1}{D} \sum_{p \in \text{testset}} \frac{1}{|y_{\text{im}}|} \sum_{y_a \in y_{\text{im}}} \mathbb{I}(P_{\text{train}}(s_0, a; p_{\text{ex}}) = y_a),
\]

where \(s_0\) represents the initialized state with the explicit symptoms, and \(y_{\text{im}}\) represents the implicit symptoms of the test record \(p\), and \(D\) is the size of the test set, and \(|y_{\text{im}}|\) is the number of the implicit symptoms. The better a patient simulator can infer the non-inquired symptoms, the higher the average accuracy will be.

In our experiment, we split the complete dataset (including training samples and test samples) into five folds and calculate the cross-validation IA for each patient simulator. The results and the standard deviation are presented in Table I. According to the results, the vanilla patient simulator is completely non-causal as it fails to infer any symptoms. The generative model learns the correlation without eliminating the collider biases and also demonstrates a quite small IA. In contrast, our PBPS performs much better at inferring correct symptoms. As shown in Table I, PBPS-PSM performs better than PS and direct estimator, which implies that PSM is less biased than PS and direct estimator. Moreover, we also compare propensity score-vector matching, which uses a vector of propensity scores of all symptom inquiries for matching. Since the score vector is also independent of the inquiry as PBPS, we denote the simulator using propensity score-vector matching as PBPS*. From Table I, we can see that without conditioning on the inquiry, PBPS* also performs significantly better than PBPS-PSM, and has a close performance to PBPS. Since the PBPS has a more compact dimension, we use PBPS in the later experiments.

**Symptom Density:** The symptoms in the collected patient records are sparse and therefore there is a great need for a patient simulator that can infer the unrecorded symptom status during training the interactive MDA agents. Besides the inferring accuracy as measured by IA, we are also interested in the proportion of the answers which are not default answers ‘not sure’, termed as Symptom Density (SD). Formally,

\[
SD = \frac{1}{D \times N_s} \sum_{p \in \text{trainset}} \sum_{a \in [1, \ldots, N_s]} \mathbb{I}(P_{\text{train}}(s_T, a; p) \neq 0),
\]

where \(s_T\) has all explicit and implicit symptoms of the record been observed. The higher SD means the simulator is more likely to answer an inquiry informatively.
TABLE II  
THE QUANTITATIVE EVALUATION OF THE PATIENT SIMULATORS BY USING THE SD METRIC

| Benchmarks | PS [3, 4] | GEN [28] | PBPS-PSM | PBPS* | PBPS |
|------------|-----------|-----------|-----------|-------|-------|
| MZ         | 0.085     | 0.081     | 0.174     | 0.391 | 0.425 |
| DX         | 0.116     | 0.110     | 0.185     | 0.414 | 0.438 |

SD is calculated among \( P_{\text{train}} \), \( P_{\text{train}} \), \( P_{\text{train}} \), \( P_{\text{train}} \), and \( P_{\text{test}} \). As shown in Table II, our PBPS has obtained the highest score in SD, meaning that our PBPS can generate more informative answers.

**Human Evaluation:** Besides, we conduct human evaluation between PS and our PBPS to distinguish which simulator is more capable of generating disease-related answers from the angle of the human doctor. We invited six human doctors to interact repeatedly with PS and PBPS, and score the *Naturalness* (NT, whose answers are informative like the human patient) and the *Symptom-Disease Consistency* (SD, whose answers are more disease-related) for each simulator per evaluation episode. More details about the human evaluation are provided in Appendix C, available online. Since we asked the human participant to choose from two simulators, the scores in Fig. 7 represent the proportions of the preference. As observed in Fig. 7, the averaging NT and SD of our PBPS have exceeded the PS sharply, which means in the view of human experts, our PBPS can generate more informative and disease-related answers.

B. Evaluation on P2A

As we have answered the first question, we propose empirical studies to demonstrate the superiority of our P2A to answer the other two questions.

To understand the benefits of the bootstrapping technique and the intervention adopted in our diagnosis agent, we design two test settings, i.e., in-distribution diagnosis (In.) and out-of-distribution diagnosis (Out.), respectively. In the In. setting, the dialogue episodes used for training and testing are all generated by interacting with \( P_{\text{train}} \). This setting aims to evaluate the basic diagnostic performances without being affected by the distribution bias since the training and testing episodes are from the same distribution. As for the Out. setting, the training dialogue episodes are generated from \( P_{\text{train}} \), then the trained diagnostic agents are tested by interacting with \( P_{\text{test}} \). All the test-simulated patients are invisible during training. Note that all evaluation results are calculated by averaging the results from five runs with different random seeds. The standard deviation is provided in each table and plot (shadow areas). This setting is to verify the cross-distribution generalization abilities of the diagnostic agent. In this setting, we are interested how the intervention process can help improve transportability. In these experiments, we define a maximum number of inquiries to avoid degenerating the active symptom inquiry into a form-filling manner. And P2A-DT is the diagnostic performance for those cases that can meet DT within the maximum number of inquiries.

**Accurate and Robust Diagnosis Across In. / Out. Settings:**

In Table III, we compare different baselines by evaluating the mean success rate over the last 20,000 training episodes. Our P2A outperforms the other RL baselines with a clear margin in either In. or Out. setting. Ought to be regarded that, unlike P2A, all other baselines perform very sensitively when the patient simulators are different for training and testing (the Out. setting). In the following, we would like to interpret the results from various aspects and ablate each component of our P2A.

**Intervention Module Facilitates the Learning of Diagnosers:**

In this part, we would like to understand whether the intervention module can help the diagnosers learn to diagnose better. To investigate the diagnosers solely, we first train P2A and then take out the bootstrapping diagnosers from P2A for evaluation. To prevent introducing the hint of inquiry behavior in the input states, we fill the non-observed symptoms of the test records using a similar formulation as Equ. (15) by replacing “trainset” and “testset” with “trainset” and “testset”, respectively. In other words, all unrecorded symptoms in the test set are inquired about at the same time and therefore there is no difference in both the category and order of the inquiries. We then feed these filled states to the diagnosers to measure the diagnostic accuracy, as summarized in Table IV. From the results, we can observe that the intervention module facilitates the learning of the diagnosers.

**Intervention Module Help Improve MDA Transportability:**

To testify whether the intervention can help improve MDA performance in terms of out-of-distribution generalization, we evaluate the diagnostic performance under In. and Out. settings. The results are presented in the columns of “No intervention” and “P2A” (with intervention) of Table V, which illustrates the averaged effects of under In. and Out. settings for intervention are 0.024 (0.888 – 0.864 + 0.944 – 0.920)/2) and 0.088 (0.784 – 0.696 + 0.928 – 0.840)/2). From these results, we observe that the intervention improves significantly under out-of-distribution settings. And we also notice that with intervention, the overall performances in both In. and Out. settings are improved, which might be due to the better learning of the diagnosers with intervention, as we discussed above.

**Bootstrapping Diagnosers Help Handle the in-Distribution Diagnosis:**

To understand whether the bootstrapping helps improve under In. setting better than Out. setting, we compare our methods with or without either bootstrapping under both settings. The results are presented in columns “No bootstrapping” and “P2A” of Table V, which illustrates that the averaged...
TABLE III

| Data | Setting | DQN [4] | KR-DQN [3] | PG-MI-GAN [42] | P2A |
|------|---------|---------|------------|----------------|-----|
| MZ   | Out.    | 0.697±0.031 | 0.653±0.016 | 0.721±0.025 | 0.784±0.035 |
|      | In.     | 0.845±0.026 | 0.755±0.016 | 0.741±0.017 | 0.888±0.032 |
| DX   | Out.    | 0.880±0.024 | 0.709±0.011 | 0.676±0.020 | 0.928±0.021 |
|      | In.     | 0.932±0.018 | 0.777±0.014 | 0.733±0.024 | 0.944±0.015 |

Fig. 8. The comparison of DQN [4], KR-DQN [3], and P2A on MZ and DX across In./Out. evaluation settings. The curves denote the mean of the success rate over iterations with deviations.

TABLE IV

| No intv. | MZ | Intv. | DX |
|----------|----|-------|----|
|          | 0.935±0.02 | 0.972±0.01 | 0.971±0.02 |
|          | 0.981±0.02 | | |

Fig. 9. Performance of P2A with PS and P2A with our PBPS under In. setting with DX and MZ datasets.

effects of bootstrapping are 0.037 (=\((0.888 - 0.834 + 0.944 - 0.924)/2\)) and 0.009 (=\((0.784 - 0.784 + 0.928 - 0.911)/2\)) under In. and Out. settings, respectively. From these results, we found that the bootstrapping technique improves the agent more under in-distribution settings. Besides, we noticed under Out. settings of MZ, the performances with and without bootstrapping are nearly the same, which indicates that improvement across distribution is mostly due to the intervention. However, we also notice a case that, under In. on DX, P2A without intervention degrades slightly more than P2A without bootstrapping. The explanation is that the intervention module not only functions in out-of-distribution cases but also helps the diagnosers learn to capture better symptom-disease relationships in general as we analyze above.

Decision Threshold is the Key Ingredient of P2A: In the above discussions, we study the effect of different components on the confidence modeling of the decision threshold. Here, we wonder what if both in-distribution and out-of-distribution confidence are not taken into account. This means the decision threshold is paralyzing since \(K \times B = 1\) and the standard deviation is 0 in Equ. (9), which makes the decision threshold always satisfied at the very beginning of the interaction. As shown in column “No DT (No boot. & No intv.)” in Table V, P2A degrades significantly in different settings. From these results, we can see that our decision threshold provides a platform where different modules, i.e., bootstrapping and intervention, can demonstrate their ability in problem-solving.

Decision Threshold Indicates Robustness Diagnosis: In Table V, within the cases that the decision threshold is satisfied (“P2A-DT”), the diagnostic accuracies maintain a very high level, i.e., exceeding 0.9 under different settings. This implies the rationale of using the decision threshold to indicate a robust diagnosis which is quite crucial for the real-world scenario. As also plotted in Fig. 8, P2A-DT appears to possess the abrupt availability of the diagnosis when the decision threshold is satisfied. Fig. 8 illustrates the success rate over the iteration of the training episodes required to train the RL agents. P2As (P2A and P2A-DT) achieve faster convergence and higher success rates in upper bounds than all other baselines. Remarkably, those episodes that met the DT achieved a very high success rate even at the very beginning of the training phase (red curve), meaning that DT only needs a small amount of training data to work reliably (i.e., the abrupt availability of diagnostic knowledge). Such reliable diagnosing performance is significant specifically in the MDA task as the data are expensive to collect.

PBPS is Beneficial to P2A With Diagnosis Accuracy and Uncertainty Modeling: To testify whether our PBPS is beneficial to P2A from both diagnosis performance and the accuracy of uncertainty modeling, we have also evaluated P2A with PS and our PBPS under the in-distribution setting, as shown in Fig. 9. ‘PS’ denotes P2A trained with PS. ‘PS-DT’, ‘PBPS’ as well as ‘PBPS-DT’ follow the same denotation rules. From the results, we observe that without our PBPS, the performance of P2A dropped sharply. Especially, under In. setting, ‘PBPS-DT’ is
able to achieve near-perfect results, meaning that with responses from our PBPS, the uncertainty modeling of P2A can almost capture the actual symptom-disease relation. Besides, the improvement of accuracy with ‘DT’ is much more obvious for ‘PBPS’, which implies that the agent trained with our PBPS gains a sharp improvement performance with better uncertainty estimation.

Different Scales of $\sigma$ for the Decision Threshold: In this part, we study the selection of different scales of $\sigma$ for the decision threshold. The experimental results are listed in Table VI. From the results, we observe that, besides some extreme cases (i.e., $1\sigma$ or $12\sigma$), P2A doesn’t appear to be sensitive to the scales of $\sigma$. The potential reason is that the diseases are distinguishable as long as the necessary symptom information has been observed. In this sense, if the scale of $\sigma$ is large enough for driving P2A to collect the necessary information for diagnosing, P2A would perform about the same. In this sense, if the scale is too small, e.g., $1\sigma$, P2A will be unable to collect enough information for diagnosing. However, we also observe that when the scale is too large, e.g., $12\sigma$, the decision threshold becomes difficult to meet which might hamper the learning of inquiry policy.

Cross-Dataset Generalization: Although different datasets are collected from different researchers [3, 4], we found that there are two shared diseases of the two datasets, i.e., upper respiratory tract infection, and infantile diarrhea. To this end, some might be interested in how different methods perform when training in DX and testing in MZ in these two shared diseases, and vice versa. To align different types of symptoms in different datasets, we make some modifications that match symptoms in the two datasets according to the semantic similarity (the L2-distance of embeddings extracted by BERT-tiny [59]). After matching, we simply rename the symptoms in one dataset with their most similar symptoms in the other. We use MZ $\rightarrow$ DX to denote the dataset which is MZ originally and is revised to DX by renaming, and so is DX $\rightarrow$ MZ. The different MDA agents are trained in the in-distribution setting using the original datasets. As for cross-dataset evaluation, all data from MZ $\rightarrow$ DX and DX $\rightarrow$ MZ are used. The results are shown in Table VII. From

the results, It can be seen that our P2A can still obtain the best performance in cross-dataset generalization.

VI. CONCLUSION

This paper presents a complete framework for the MDA task, CA-MDA, including a simulator PBPS that tackles the problem of counterfactual symptom inquiry by PLM, and an MDA agent P2A that additionally eliminates the distributional bias via intervention and models the confidence to drive the symptom inquiry. Through mining the passive observational data in such a collaborative representation manner, to the best of our knowledge, this paper is the first study to propose a simple yet concise paradigm to address sequential discrimination tasks with only passive observational data. Experimental results demonstrate that PBPS can generate more informative and disease-related answers. Moreover, P2A is more accurate and robust across distributions. Our introduced Decision Threshold provides a reliable stop mechanism for MDA agents. In the future, we will study how to extend our framework to broader ranges of sequential discriminative decision-making problems with passive observational data.

REFERENCES

[1] V. Mnih et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, 2015, Art. no. 529.
[2] S. R. Sutton, A. D. Mcallester, P. S. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” in Proc. Int. Conf. Neural Inf. Process. Syst., 1999, pp. 1057–1063.
[3] L. Xu, Q. Zhou, K. Gong, X. Liang, J. Tang, and L. Lin, “End-to-end knowledge-routed relational dialogue system for automatic diagnosis,” in Proc. Conf. Assoc. Advance. Artif. Intell., 2019, pp. 7346–7353.
[4] Z. Wei et al., “Task-oriented dialogue system for automatic diagnosis,” in Proc. Assoc. Comput. Linguistics, 2018, pp. 201–207.
[5] S. R. Cole et al., “Illustrating bias due to conditioning on a collider,” Int. J. Epidemiol., vol. 39, no. 2, pp. 417–420, 2010.
[6] D. B. Rubin, “Estimating causal effects of treatments in randomized and nonrandomized studies,” J. Educ. Psychol., vol. 66, no. 5, 1974, Art. no. 688.
[7] J. Neyman, “On the application of probability theory to agricultural experiments: Essay on principles,” Roczniki Nauk Rolniczych, vol. 10, pp. 1–51, 1923.
[8] R. H. Dehejia and S. Wahba, “Propensity score-matching methods for nonexperimental causal studies,” Rev. Econ. Statist., vol. 84, pp. 151–161, 2002.
[9] P. C. Austin, “An introduction to propensity score methods for reducing the effects of confounding in observational studies,” Multivariate Behav. Res., vol. 46, no. 3, pp. 399–424, 2011.

[10] P. R. Rosenbaum and D. B. Rubin, “The central role of the propensity score in observational studies for causal effects,” in Biometrika. London, U.K.: Oxford Univ. Press, 1983.

[11] J. Pearl, “The Do-Calculus revisited,” in Proc. Conf. Uncertainty Artif. Intell., 2012, pp. 3–11.

[12] R. Ratcliff, P. L. Smith, S. D. Brown, and G. McKoon, “Diffusion decision model: Current issues and history,” Trends Cogn. Sci., vol. 20, no. 4, pp. 260–281, 2016.

[13] J. Pearl et al., “Causal inference in statistics: An overview,” Statist. Surv., vol. 3, pp. 96–146, 2009.

[14] K. A. Bollen and J. Pearl, “Eight myths about causality and structural equation models,” in Handbook of Causal Analysis for Social Research, Berlin, Germany: Springer, 2013, pp. 301–328.

[15] L. Chen, H. Zhang, J. Xiao, X. He, S. Pu, and S.-F. Chang, “Counterfactual critic multi-agent training for scene graph generation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 4613–4623.

[16] T. Kaihua, N. Yulei, H. Jianqiang, S. Jiaxin, and Z. Hanwang, “Unbiased scene graph generation from biased training,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 3716–3725.

[17] J. Qi, Y. Niu, J. Huang, and H. Zhang, “Two causal principles for improving visual dialog,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 10 860–10 869.

[18] T. Wang, J. Huang, H. Zhang, and Q. Sun, “Visual commonsense R-CNN,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 10 760–10 770.

[19] E. Abbassnejad, D. Teney, A. Parvaneh, J. Shi, and A. V. D. Hengel, “Counterfactual vision and language learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 10 044–10 054.

[20] C. Yu, J. Liu, S. Nemati, and G. Yin, “Reinforcement learning in healthcare: A survey,” ACM Comput. Surv., vol. 55, no. 1, pp. 1–36, 2021.

[21] I. Dasgupta et al., “Causal reasoning from meta-reinforcement learning,” arXiv:1901.08162.

[22] M. Oberst and D. Sontag, “Counterfactual off-policy evaluation with Gumbel-Max structural causal models,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 4881–4890.

[23] N. J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, “Counterfactual multi-agent policy gradients,” in Proc. AAAI Conf. Artif. Intell., 2018, pp. 2974–2982.

[24] E. Bareinboim and J. Pearl, “Causal inference and the data-fusion problem,” in Proc. Nat. Acad. Sci. USA, vol. 113, 2016, pp. 7345–7352.

[25] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young, “External validity: From Do-Calculus to transportability across populations,” Stat. Sci., vol. 29, pp. 579–595, 2014.

[26] M. Jakubowski, “Latent variables and propensity score matching: A model,” in Modern Probability Theory and Its Applications. Hoboken, NJ, USA: Wiley, 2006.

[27] J. VanderWeele, “A three-way decomposition of a total effect into direct, indirect, and interactive effects,” Epidemiology, vol. 24, no. 2, 2013, Art. no. 224.

[28] J. Pearl, “Causal diagrams for empirical research,” Biometrika, vol. 82, no. 4, pp. 669–688, Dec. 1995.

[29] D. Koller and N. Friedman, Probabilistic Graphical Models: Principles and Techniques, Cambridge, MA, USA: MIT Press, 2009.

[30] X. Huang and M. Valtorta, “Pearl’s Calculus of intervention is complete,” in Proc. Conf. Uncertainty Artif. Intell., 2012, pp. 217–224.

[31] J. Pearl and E. Bareinboim, “External validity: From Do-Calculus to transportability across populations,” Stat. Sci., vol. 29, pp. 579–595, 2014.

[32] P. Whaite and F. P. Ferrie, “Autonomous exploration: Driven by uncertainty,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 3, pp. 203–205, Mar. 1997.

[33] P. Whaite and F. P. Ferrie, “From uncertainty to visual exploration,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 13, no. 10, pp. 1038–1049, Oct. 1991.

[34] I. Osband, J. Aslanides, and A. Cassirer, “Randomized prior functions for deep reinforcement learning,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2018, pp. 8626–8636.

[35] Y. Bura, H. Edvats, A. Starkey, and O. Klimov, “Exploration by random network distillation,” in Proc. Int. Conf. Learn. Representations, 2019.

[36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, arXiv:1810.04805.
Keze Wang received the PhD degrees from Sun Yat-sen University in 2017 and Hong Kong Polytechnic University in 2019. He is nationally recognized as the distinguished young scholars, currently serving as an associate professor with the School of Computer Science, Sun Yat-sen University, and a doctoral supervisor. In 2018, he worked as a postdoctoral researcher with the University of California, Los Angeles, and returned to Sun Yat-sen University in 2021 as part of the “Hundred Talents Program.” He has focused on reducing deep learning’s dependence on training samples and mining valuable information from massive unlabeled data.

Ziliang Chen received the BS degree in mathematics and applied mathematics from Sun Yat-sen University, Guangzhou, China and the PhD degree in computer science and technology, advised by Prof. L. Lin. After that, he worked as an assistant professor in Jinan University. His research interests include computer vision and machine learning.

Xiaodan Liang received the PhD degree from Sun Yat-sen University, in 2016, advised by Liang Lin. She is an associate professor with Sun Yat-sen University. She served as an area chair of CVPR 2020, ICCV 2019 and WACV 2020. She was a postdoc researcher with Machine Learning Department, Carnegie Mellon University, working with Prof. Eric Xing, from 2016 to 2018. She has published several cutting-edge projects on human-related analysis including human parsing, pedestrian detection and instance segmentation, human pose estimation, and activity recognition.

Liang Lin is a full professor of Sun Yat-sen University. He served as the executive R&D director and distinguished scientist of SenseTime Group from 2016 to 2018, taking charge of transferring cutting-edge technology into products. He has authored or coauthored more than 200 papers in leading academic journals and conferences. He is an associate editor of *IEEE Transactions Human-Machine Systems*, and *IET Computer Vision*. He served as area chairs for numerous conferences such as CVPR, ICCV, and IJCAI. He is the recipient of numerous awards and honors including Wu Wen-Jun Artificial Intelligence Award, ICCV Best Paper Nomination in 2019, Annual Best Paper Award by Pattern Recognition (Elsevier) in 2018, Best Paper Dimond Award in IEEE ICME 2017, and Google Faculty Award in 2012. He is a fellow of IET.