frontiers in Artificial Intelligence

ORIGINAL RESEARCH
published: 15 December 2021
doi: 10.3389/frai.2021.798659

Towards Effective Patient Simulators

Vadim Liventsev[1,2]*, *Aki Härmä*[2] *and Milan Petković*[1,2]

[1]Eindhoven University of Technology, Eindhoven, Netherlands, [2]Philips Research (Netherlands), Eindhoven, Netherlands

In this paper we give an overview of the field of patient simulators and provide qualitative and quantitative comparison of different modeling and simulation approaches. Simulators can be used to train human caregivers but also to develop and optimize algorithms for clinical decision support applications and test and validate interventions. In this paper we introduce three novel patient simulators with different levels of representational accuracy: HeartPole, a simplistic transparent rule-based system, GraphSim, a graph-based model trained on intensive care data, and Auto-ALS—an adjusted version of an educational software package used for training junior healthcare professionals. We provide a qualitative and quantitative comparison of the previously existing as well as proposed simulators.

Keywords: reinforcemenet learning, healthcare, simulators and models, clinical methods, markov decision chain

1 INTRODUCTION

Patient simulators have been applied extensively in nursing education (Wendy et al., 2001; Nehring and Lashley, 2004; McFetrich, 2006), both as physical mannequins and as digital patient simulation software (Lee et al., 2020). However, patient simulators have another important application: they can be used to predict how patients' health would respond to various treatments and thus evaluate proposals for novel treatment strategies. And now, with the advent of artificial intelligence, one can use reinforcement learning (Yu et al., 2019), genetic programming (Koza, 1994; Niazkar and Niazkar, 2020) or both (Liventsev et al., 2021) to generate treatment strategies that lead to the best outcomes for the patient according to a certain simulator. In this paper we review the field of interactive patient simulators and evaluate their potential to be used for automated treatment discovery.

The contributions of this paper are as follows:

- A systematic review of patient simulators in Healthcare Informatics
- *Auto-ALS*—a learning aid for junior healthcare professionals adapted to reinforcement learning
- *HeartPole*—a novel simple and transparent pseudosimulator designed to be a convenient benchmark
- *GraphSim*—a data-driven graph-based simulator based on MIMIC-IV dataset (Johnson et al., 2021) for maximal accuracy

1.1 Scope of This Survey

This survey only includes interactive patient simulators where a (human or software) agent communicates its clinical intervention decisions to the simulator and the simulator, in turn,

1) *Predicts* future patient state given current patient state and clinical interventions.
2) *Rewards* or punishes (negatively rewards) that judges if the result of the treatment is positive or negative. The model can be as simple as just negatively rewarding patient deaths or as complex as using the predictions of the *prediction model* to help the agent recognize during the episode whether they are on the right track to a positive outcome.

OPEN ACCESS

Edited by:
Naimul Khan,
Ryerson University, Canada

Reviewed by:
Batyrkhan Omarov,
Al-Farabi Kazakh National University, Kazakhstan
Arriel Benis,
Holon Institute of Technology, Israel

***Correspondence:**
Vadim Liventsev
v.liventsev@tue.nl

Specialty section:
This article was submitted to Medicine and Public Health, a section of the journal Frontiers in Artificial Intelligence

Received: 20 October 2021
Accepted: 29 November 2021
Published: 15 December 2021

Citation:
Liventsev V, Härmä A and Petković M (2021) Towards Effective Patient Simulators. Front. Artif. Intell. 4:798659. doi: 10.3389/frai.2021.798659

Any interactive simulator with a prediction model and a reward model (and thus, all simulators in this paper) can be represented as *Episodic Partially Observable Markov Decision Process* (EPOMDP) as described in (Liventsev et al., 2021):

$$M = \left(\mathcal{S}_{nt}, \mathcal{S}_t, \mathcal{A}, \mathcal{O}, p_o\left(o|s,a\right), p_s\left(s_{\text{next}}|s_{\text{prev}},a\right), p_r\left(r|s,a\right), p_{\text{init}}\left(s\right)\right) \tag{1}$$

Here, $\mathcal{S}_{nt}$ is the set of *non-terminal patient states*. $\mathcal{S}_t$ is the set of *terminal patient states* (death, discharge from hospital or end of outpatient treatment) such that getting to one of these states signifies the end of an episode. $\mathcal{A}$ is the set of *actions* that the learning agent can perform, and $\mathcal{O}$ is the set of *observations* about the current state that the agent can make. Distributions p_o, p_s, and p_{init} represent the simulator's prediction model, while p_r represents the reward model.

Markov Decision Processes are a standard formalism in the field of Reinforcement Learning (Sutton and Barto, 2018), so standard RL algorithms can be used in conjunction with any simulator from this paper for treatment discovery.

There are 2 notable examples of systems that can be referred to as patient simulators, but do not make it easy to evaluate treatment strategies and fall outside the scope of this survey:

1) prediction models without a reward model
2) synthetic data generators (Chen et al., 2021)

They former are tools like HumMod (Hester et al., 2011) that help predict how change in one variable of patient health can affect another, for instance, how digoxin intake affects blood pressure. A reward model is absolutely necessary for the application area that we focus on in this paper—development and validation of treatment strategies. Developing it is also a non-trivial problem in its own right—which health outcomes are considered *good* depends on severity of the patient's condition (*alive* is a sufficiently successful outcome for the most severe conditions, but not for others) as well as the patient's own preferences (Street et al., 2012; MühlbacherMühlbacher and Juhnke, 2013). Thus, we view prediction models as components that can be used to develop a full simulator in the future.

The latter simulate clinical scenarios by generating samples of clinical histories. They provide a privacy-preserving way to generate a lot of training data for treatment discovery (James et al., 2018; Gong et al., 2020; Rankin et al., 2020; Wang et al., 2021), however, they don't model counterfactuals. That is, they don't answer questions of the sort "How would the patient's health respond to decision X"? For example, Synthea (Cavalcante et al., 2016) is a framework for generation of entire synthetic Electronic Health Records, EHRs, of patients, from the first to the last patient encounter, but the system does not provide ways to test alternative interventions on patients.

1.2 Simulators as Models

What makes an *effective* patient simulator? We argue that there are two use cases for a patient simulator: modeling and benchmarking, with divergent effectiveness criteria.

Firstly, a simulator can be used as a *patient model* aiming to represent the real world evolution of patient health in response to various treatments as closely as possible. The effectiveness criterion for a patient model is simple—*accuracy*. Accuracy can be measured by comparing the simulator's predictions against a dataset of clinical histories. One metric in particular that has been used for simulator validation is *mean predictive error over dynamic transitions* (Summers et al., 2009). If the simulator was obtained with machine learning, the data also has to be different from the training data: a different dataset or a holdout set can be used for that purpose.

The trust graph (see **Figure 1**) for a model is equally simple: if the data used for development of the simulator is a representative sample of the true distribution of the patients and the methodology used to turn build an interactive simulator based on these data is sound, the simulator is an accurate representation of a real world patient. And if the training algorithm used to discover the optimal treatment under this simulator is sound, the resulting treatment strategy can be trusted as well. Unfortunately, these conditions often hold only partially, or do not hold at all.

1.3 Common Biases

One type of bias, present in most data-driven healthcare simulators is *sampling bias*. If patient data is used to develop a simulator, this data is a sample (and, potentially, a biased sample) of the patient population that may or may not represent the population accurately. This problem is exacerbated by a profound shortage of healthcare datasets available to the research community. The biggest currently available dataset is MIMIC-IV (Johnson et al., 2021)—a large database of electronic health records in intensive care. However, MIMIC-IV was collected in one hospital in Boston and thus represents a demographically biased sample of the population. This bias is likely to percolate into any simulator developed based on MIMIC-IV.

But the most crippling type of simulator bias that makes many simulators completely unsuitable to serve as patient models is *confirmation bias*. *Confirmation bias* occurs when the developers of a simulator are aware of state of the art clinical practices and, intentionally or unintentionally, develop a simulator that rewards them and punishes alternatives. In *didactic* simulators used for education of clinical professionals, confirmation bias exists by design: students are, after all, trained to follow established clinical protocols. For a particular example of intentional confirmation bias in simulator design see *Virtu-ALS* (**section 2.3**). In this simulator, a decision that violates the existing emergence care protocol will be registered as a mistake, explained as such to the student, and *not actually implemented*.

1.4 Simulators as Benchmarks

Heavily biased and otherwise inaccurate simulators should not be used as patient models and should not be used as a basis for treatment recommendations. However, they can make a great *benchmarks* for learning algorithms. A benchmark is used not to develop a novel treatment strategy and apply it in the real world, but to compare different machine learning algorithms for healthcare against each other. Simulators that exhibit confirmation bias are a good fit for this, since they are similar

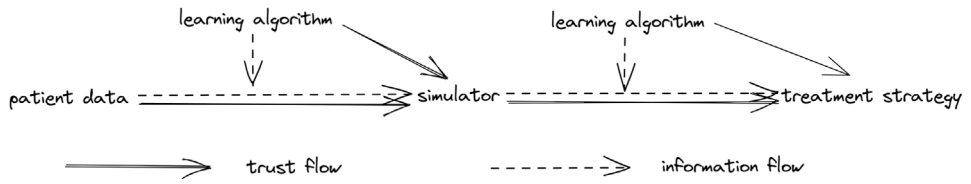

FIGURE 1 | Trust graph of simulator as a model.

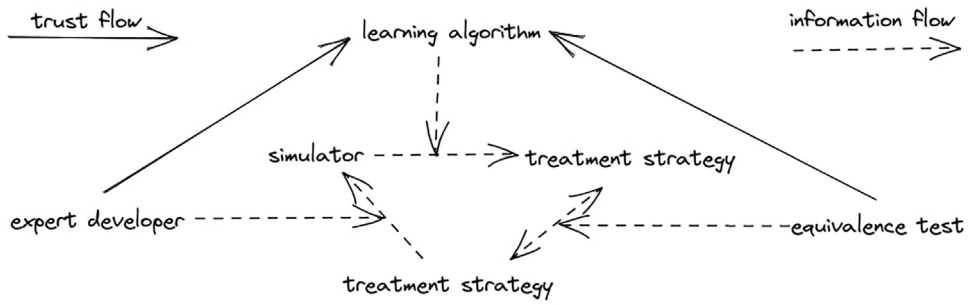

FIGURE 2 | Trust graph of simulator as a benchmark.

to true patient models, but unlike in a true patient model, treatments that these simulators evaluate as optimal are known beforehand. One can test a learning algorithm by applying it to a biased simulator and checking that the resulting strategy is equivalent to the strategy that simulator developers had in mind, as displayed on the trust graph in **Figure 2**.

A crucial metric for effective benchmarks is their *difficulty*. A *difficult* simulator is one where most learning algorithms fail to discover an effective treatment strategy, i.e. a strategy that leads to positive health outcomes. A benchmarks's purpose is separating stronger learning algorithms from weaker ones, hence a good benchmark has to be not too difficult and not too easy. To get a grasp on simulator difficulty we train a baseline reinforcement learning model on each, see **section 4**.

Another desideratum for benchmarks is *transparency*. When a learning algorithm fails to discover an effective treatment strategy, it is very useful to understand what went wrong in detail. Any information output by the simulator other than its predictions as well as having easy access to its internal logic can be useful for developers.

Didactic simulators can be particularly good benchmarks (Liventsev, 2021a), but synthetic games akin to CartPole (Barto et al., 1983) and MountainCar (Moore, 1990) (commonly used as benchmarks for Reinforcement Learning algorithms) that aim to mimic only certain aspects of real life healthcare tasks can also be used. A particularly illustrative example is *Healing MNIST* (Krishnan et al., 2015)—a modified version of the industry standard MNIST Handwritten Digits (Li Deng, 2012) dataset to which rotations and random noise have been added. The authors argue that this dataset reflects important properties of healthcare tasks: rotations represent evolution of patient state over time, while the randomised "squares within the sequences are intended to be analogous to seasonal flu or other ailments that a patient could exhibit that are independent of the actions and which last several timesteps." Due to the data shortage in Healthcare many proposals for data-driven clinical decision support systems are tested on such relatively unrealistic benchmarks with the assumption that the system can be retrained on real patient data and simulators shall they arrive in the future.

2 EXISTING SIMULATORS

2.1 Simglucose

UVA/Padova (Man et al., 2014) is a set of equations used to model type 1 diabetes. The equations, outlined on **Figure 3**, were developed by clinical experts and validated on a dataset of 32 people aged 38 ± 12 years. It is widely used in Healthcare and even approved in the United States as a replacement for clinical trials. It provides p_o ($o|s$, a) and p_s ($s_{\text{next}}|s_{\text{prev}}$, a) (see **section 1.1**), so to be a full-fledged Markov Decision Process it only need p_r ($r|s$, a). (Xie, 2018). solves exactly that by adding a reward function based on diabetes risk index as defined in (Clarke and Kovatchev, 2009) to the UVA/Padova simulator, providing a Reinforcement Learning environment for type 1 diabetes.

2.2 GYMIC

2.2.1 Scope

GYMIC (Kiani et al., 2019) is, unlike the previous examples, a fully data-driven simulator. It harnesses a subset of MIMIC (Johnson et al., 2021) dataset to address on one of the most challenging problems in emergency care—sepsis. The authors

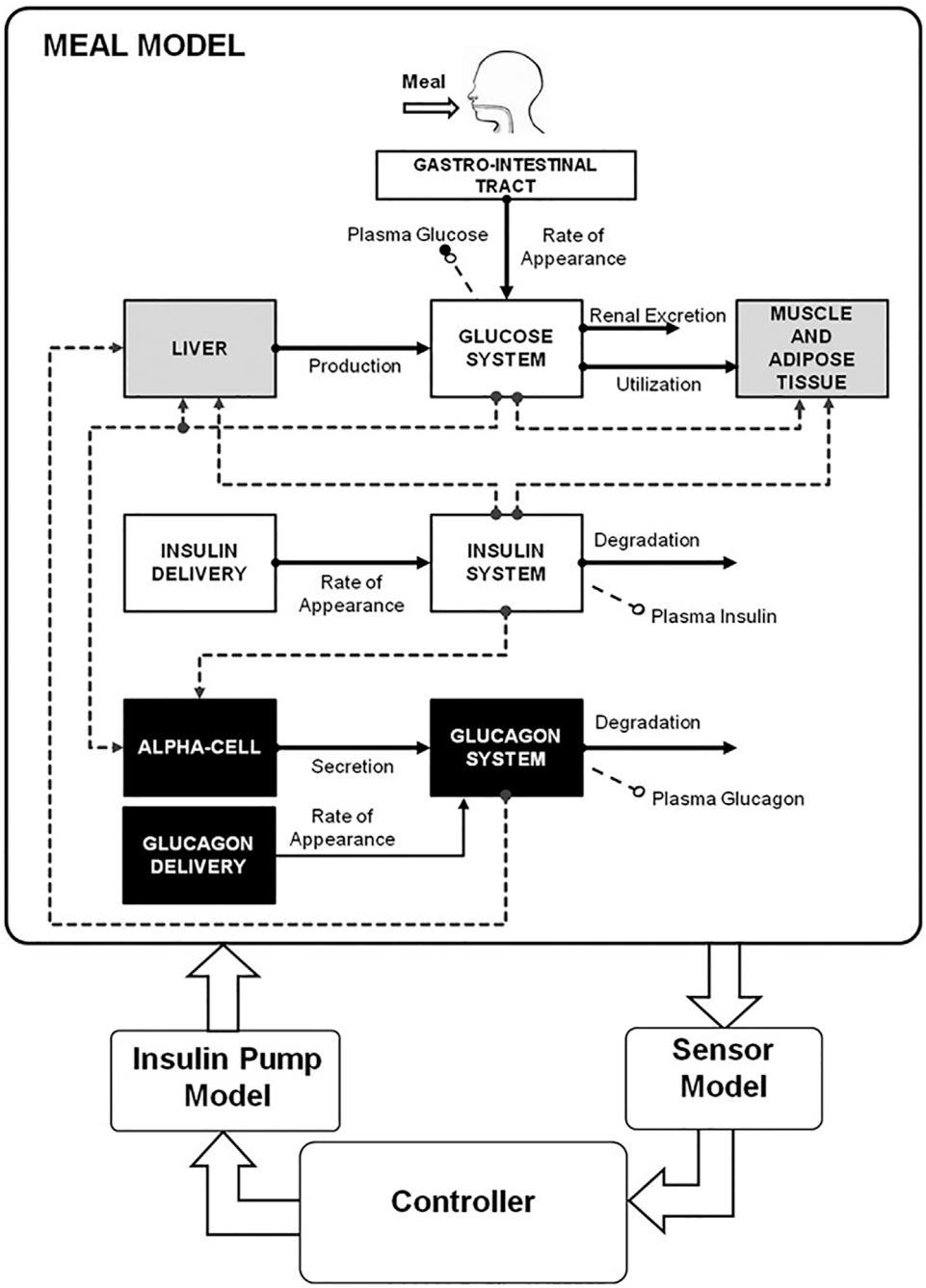

FIGURE 3 | UVA/PADOVA equations, visualised.

intentionally limit their scope to just sepsis in order to simplify the modelling task as well as because sepsis prevention has been identified as an area where doctors would particularly benefit from electronic decision support (Raghu et al., 2017; Walonoski et al., 2018).

2.2.2 Prediction Model

The prediction model of GYMIC simulator is defined as a solution to the following autoregression task:

1) A clinical history is a sequence of (s, a) tuples
2) $a \in 0, \ldots 24$ is one of 25 possible vasopressor or intravenous fluid interventions—a cartesian product of five types of interventions and five dosage quantiles.
3) $s \in R^{46}$ is the patient's state at the moment this intervention was administered.
4) Predict the conditional state distribution $p_s\ (s_t,\ a_t | s_{t-1},\ s_{t-2},\ \ldots,\ s_1)$

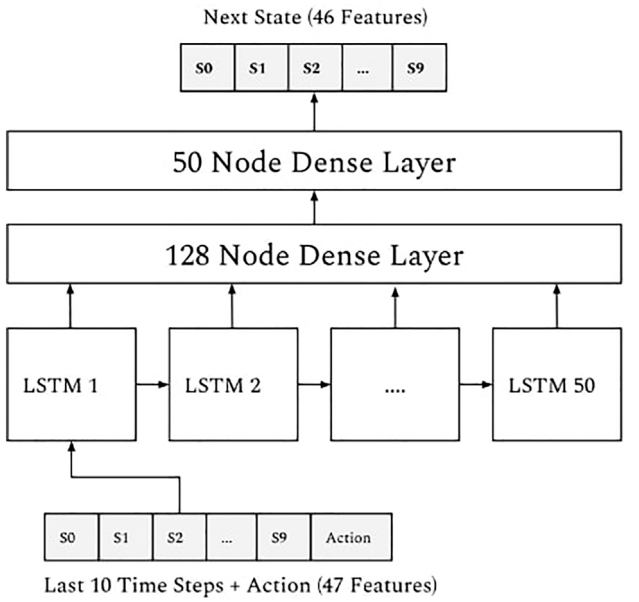

FIGURE 4 | Neural architecture of GYMIC.

The dataset of clinical histories is produced by a preprocessing algorithm combining together all clinical records from MIMIC that relate to sepsis patients.

Autoregressive tasks of this nature arise in many fields like stock market prediction (Cavalcante et al., 2016; Jiang, 2021) or language modelling (Jozefowicz et al., 2016) where state of the art solutions can be found. The authors of *GYMIC* solve it with an LSTM (Hochreiter and Schmidhuber, 1997) neural network with two additional dense layers attached, see **Figure 4** for the diagram. Faced with some of the mode collapse issues described in **section 2.2.4** the authors also experimented with semi-supervised learning [?]: they trained a variational autoencoder [?] on all patient states to replace the 46-vector representations of patient state s with learned representations from the latent space of the VAE encoder(s). The issues persisted.

2.2.3 Reward Model

Highlighting the gravity of contracting sepsis, *GYMIC* has only two outcomes: patient is discharged from intensive care or patient dies. Its reward model reflects that, giving the agent a large positive or negative reward at the end of the episode, depending on the outcome. However, in order to lower the difficulty of the simulator (delayed gratification makes training significantly harder (Mischel and EbbesenEbbesen, 1970; Gulwani et al., 2017)) an additional reward is provided during the episode, based on the evolution of the patient's SOFA score (Lambden et al., 2019)—a commonly used measure of sepsis severity:

$$r(s_t, s_{t+1}) = C_0 1\left(s_{t+1}^{\text{SOFA}} = s_t^{\text{SOFA}} \& s_{t+1}^{\text{SOFA}} > 0\right) + \\ + C_1\left(s_{t+1}^{\text{SOFA}} - s_t^{\text{SOFA}}\right) + C_2 \tanh\left(s_{t+1}^{\text{Lactate}} - s_t^{\text{Lactate}}\right) \quad (2)$$

A third reward component is proposed to negatively reinforce action severity and encourage the agent to use low doses of drugs - an instance of *confirmation bias* as discussed in **section 1.3**, but a necessary step given the issues in **section 2.2.4**.

2.2.4 Results and Issues

Unfortunately, the experiments performed by the authors of *GYMIC* indicate extreme overfitting. Due to *sampling bias* and simply inadequate size of the dataset there are treatments that have only occurred a few times in the training data and have always resulted in a positive health outcome. In *GYMIC* these treatments are silver bullets that guarantee a successful outcome while in real life they are risky and potentially very harmful.

2.3 Virtu-ALS

Virtu-ALS is a *didactic* emergency care simulator mainly targeted at students and junior healthcare professionals, although its application as a reinforcement learning *benchmark* was anticipated and accounted for by the authors (Brisk et al., 2018). Its most prominent feature is its visual nature (**Figure 5**): the user has access to a 3D-rendered virtual copy of a hospital room, view the monitor, press buttons on a defibrillator, etc. However, the visual modality means that its observation space

$$\mathcal{O} \subset R^{307200} \quad (3)$$

Such a high dimensionality of the observation space makes it an extremely challenging reinforcement learning task. Tasks from this family have been solved with deep neural networks (Mnih et al., 2013), however not only does it require a long and expensive training process, it also means that resulting treatment strategies are black box neural networks that no clinical expert understands. This approach to decision making is extremely hard to introduce into clinical practice (Price, 2018; Watson et al., 2019).

Like most *didactic* simulators, Virtu-ALS exhibits considerable *confirmation bias*—any decision that's not supported by the standard emergency care protocol (American Heart Association Staff, 2006; Thim et al., 2012) is considered a mistake and rewarded negatively.

3 PROPOSED SIMULATORS

3.1 Auto-ALS

As our first model, we propose a low-dimensional version of *Virtu-ALS*. *Auto-ALS* is a modification of Virtu-ALS that removes all the complexity of dealing with a visual 3D environment while retaining all the complexity of dealing with a patient that requires emergency care. This is achieved by attaching an event listener to Virtu-ALS that registers all observable events that can occure in the simulator in response to the user's actions. The events are listed in **Table 1**, organized by which agent action can trigger which event. *Tick* is a special event that occurs every time the simulator is advanced a timestep, and is negatively reinforced, which when used with reinforcement learning algorithms discourages clinicaly unnecessary actions.

FIGURE 5 | Virtu-ALS.

TABLE 1 | All actions and observations of Auto-ALS.

Agent actions	Patient reactions	Rewards
AssessResponse	ResponseVerbal, ResponseGroan, ResponseNone	0
AssessAirway	AirwayClear, AirwayVomit, AirwayBlood, AirwayTongue	
AssessBreathing	BreathingNone, BreathingSnoring, BreathingSeeSaw, BreathingEqualChestExpansion, BreathingBibasalCrepitations, BreathingWheeze, BreathingCoarseCrepitationsAtBase, BreathingPneumothoraxSymptoms, VentilationResistance, *MeasuredRespRate*	
AssessCirculation	RadialPulsePalpable, RadialPulseNonPalpable, *MeasuredHeartRate*	
AssessDisability	AVPU_A, AVPU_U, AVPU_V, PupilsPinpoint, PupilsNormal, *MeasuredCapillaryGlucose*	
AssessExposure	ExposureRash, ExposurePeripherallyShutdown, ExposureStainedUnderwear, *MeasuredTemperature*	
AssessDefibrillator		
AssessMonitor	HeartRhythm0, HeartRhythm1, HeartRhythm2, HeartRhythm3, HeartRhythm4, *MeasuredHeartRate*, *MeasuredMAP*, *MeasuredSats*, *MeasuredResps*	
DoNothing		
ABG, AirwayManoeuvres, GiveAtropine, GiveAdenosine, GiveAdrenaline, GiveAmiodarone, GiveMidazolam, Venflon, Yankeur, DrawBloods, BPCuffOn, BVM, Guedel, NRBMask, DefibOn, DefibAttachPads, DefibShock, DefibCharge, DefibChangePaceCurrentDown, DefibChangePaceCurrent, DefibEnergyDown, DefibEnergyUp, DefibChangePaceRateDown, DefibChangePaceRateUp, DefibPace	Blunder	r_{blunder}
Finish	Failure	−1
	Success	1
—	Tick	r_{tick}

MeasuredHeartRate, MeasuredRespRate, MeasuredCapillaryGlucose, MeasuredTemperature, MeasuredMAP, MeasuredSats, and MeasuredResps are *measurements*, events that have a value ($-\infty$; $+\infty$) associated with them.

The events in **Table 1** only get registered if the agent has *learnt* some piece of information, meaning that, for example, AirwayVomit will only occur if the patient has vomit in their airway *and* the agent checked the airway (which is part of the standard protocol (Thim et al., 2012)). Assessment skills (knowing where to look and how to establish the patient's state) are crucial for patient resuscitation, hence revealing all known health variables to the agent would jeopardize the simulation.

The observation vector in *Auto-ALS* is based on all observations that have occurred between the beginning of the episode and current time. However, more recent observations are more likely to still be relevant and should be given priority. This is done with the following formula proposed in (Liventsev, 2021b):

$$o^+ = \langle o_1 \in O_1, \exp(t_1 - t), \ldots, o_n \in O_n, \exp(t_n - t), \rangle \quad (4)$$

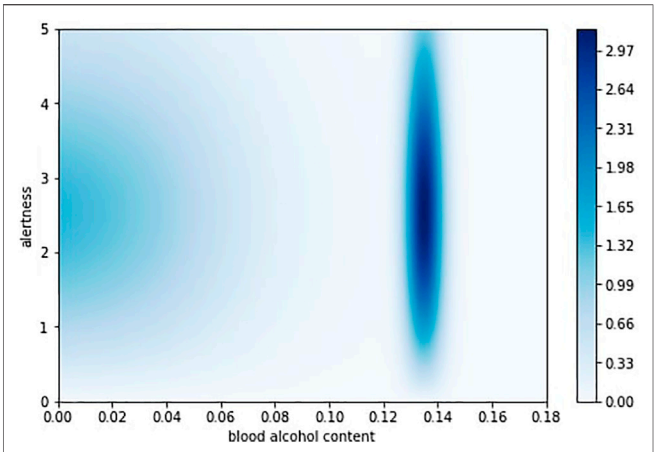

FIGURE 6 | Productivity function in HeartPole.

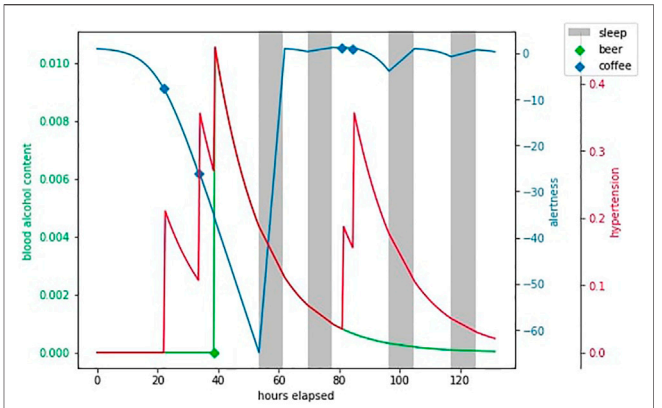

FIGURE 7 | An example episode in HeartPole.

where O_i is the value of the observation and t is current time and t_i is time when observation i (for $i = 5$, ResponseGroan) has *last* occurred and exp $(t_i - t)$ represents its decaying relevance. For *measurements*, the O_i equals the magnitude of the measurement, however, for binary obsevations O_i would always be equal to one. For memory efficiency, for all i that correspond to binary observations, O_i is skipped from the o^+ vector and the actual observation vector o has size $36 + 7*2 = 50$, as opposed to $(36 + 7)*2 = 86$.

See source code and documentation at (Liventsev, 2021c).

3.2 HeartPole

HeartPole focuses on simplicity and transparency at the expense of realism—it is based on a familiar scenario and a simple set of rules so that when a treatment performs badly it is easy to explore what exactly goes wrong. *HeartPole* simulates a creative professional trying to become more productive. However, many decisions that would help in the short term (not sleeping, consuming coffee and alcohol) can create long-term health issues that negate all short term gains.

In *HeartPole* state s consists of alertness s_t^{alert}, hypertension s^{hypert}, intoxication s^{tox} time since slept s^{tawake}, *total time elapsed* s^{ttotal} and *total work done* s^{done}.

Over these parameters, we define *productivity* function $\eta(s_t^{\text{alert}}, s_t^{\text{tox}})$ presented graphically on **Figure 6** and *heart attack probability* $r(s_t^{\text{hypert}}) = \frac{\text{sigmoid}(s_t^{\text{hypert}})}{2}$. The agent receives small positive rewards for productivity and a very large negative reward if a heart attack occurs.

As shown on **Figure 7**, every half an hour awake, the agent observes s_t and picks an action a_t from discrete action space of *just work*, *drink coffee* (increases s^{alert} and s^{hypert}), *drink beer* (decreases s^{alert}, increases s^{hypert} and s_t^{tox}) and *go to bed* (sleep takes a lot of time, but reduces s^{hypert} and s_t^{tox} and without it alertness starts to fall very fast).

See source code and documentation at (Härmä et al., 2021).

3.3 GraphSim

Our last model is trained on MIMIC (Johnson et al., 2021) to maximise *accuracy*. MIMIC can be represented as a set of patients where every patient is an oriented graph, its nodes are patient states, each state is a vector of various clinical measurements, such as blood pressure and oxygen saturation, whereas arcs that connect the patient states are doctors actions, each action a vector of administered drug doses. These oriented graphs can also be viewed as disjoint clusters in one graph of all possible patient states, a sequence of patient states connected by doctor agents.

$$G = \{\langle s_{\text{before}}, a, s_{\text{after}}\rangle\} \tag{5}$$

Some states in this oriented graph are very similar, it is reasonable to assume that two different patients have been in the same state at some point. Our algorithm is based on the idea that if two patients have been in the same state, their clinical histories represent two possible timelines of events after the state and the choice of timeline depends on doctor's actions.

We find all state pairs $\langle s_A, s_B\rangle$ below a similarity threshold

$$cos(s_A, s_B) < c_{\min} \tag{6}$$

and merge each into a single state, replacing occurences of s_A and s_B in G with $\frac{s_A+s_B}{2}$. The resulting oriented graph becomes the backbone of out simulator. The simulated patient is initialized in state s_0 equal to one of the initial states of real patients in MIMIC. When at timestep t the agent picks action a_t, transition to the next state depends on euclidean distance between the action in the graph and a_t via the softmax function:

$$p(s_{t+1}|s_t, a_t) = \frac{\sum_{\langle s_{\text{before}}, a, s_{\text{after}}\rangle}^{G} \mathbb{I}[s_{\text{before}} = s_t]\mathbb{I}[s_{\text{after}} = s_{t+1}]e^{|a_t - a|_2^2}}{\sum_{\langle s_{\text{before}}, a, s_{\text{after}}\rangle}^{G} \mathbb{I}[s_{\text{before}} = s_t]e^{|a_t - a|_2^2}} \tag{7}$$

where $\mathbb{I}$ is the indicator function. See **Figure 8** for a diagram of the resulting simulator.

The simulator's source code will be published soon.

4 EFFECTIVENESS

How do these simulators fare with respect to *accuracy* criteria we set out in the introduction? The factors that contribute to a

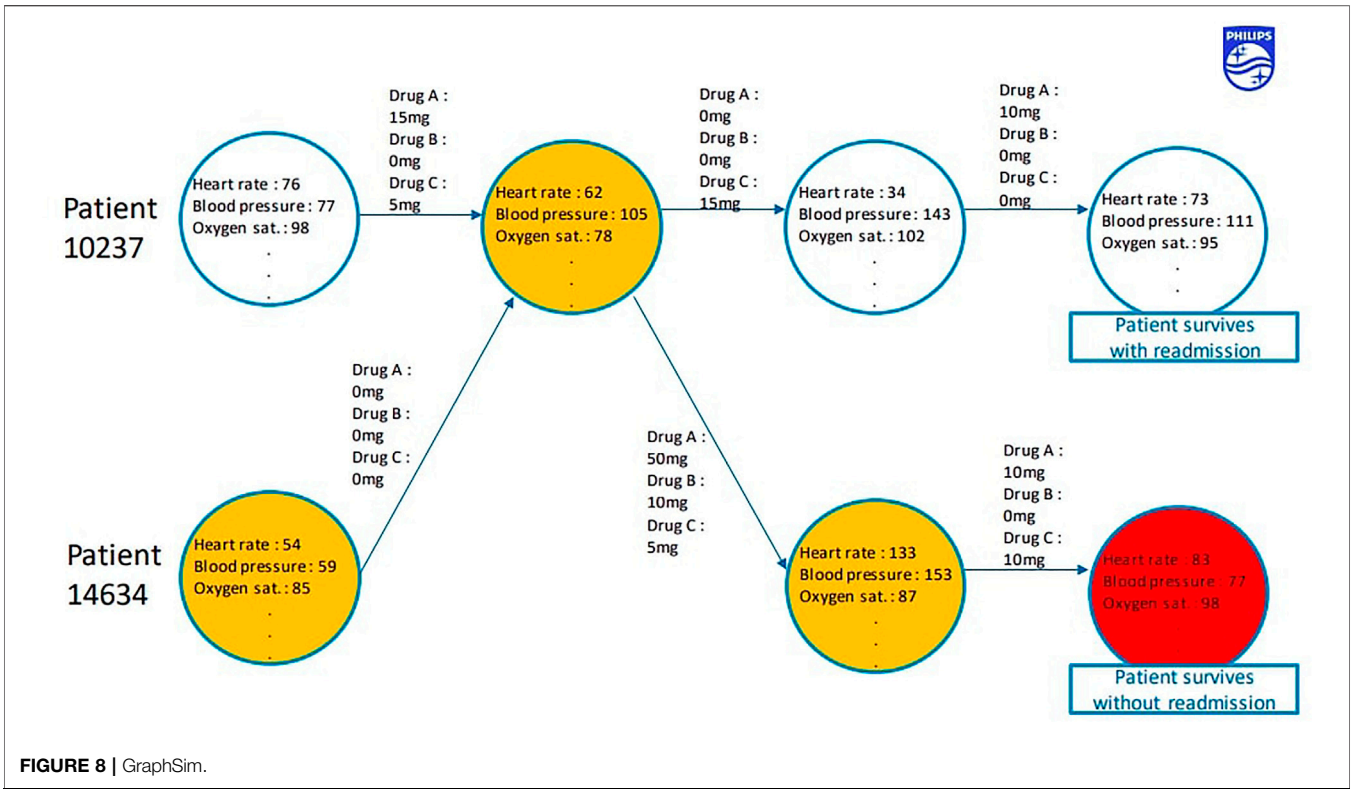

FIGURE 8 | GraphSim.

TABLE 2 | Summary of each simulator: trust view.

Simulator	Scope	Data source	Sample size	Learning algorithm	Known biases
simglucose (Xie, 2018)	type 1 diabetes	original study	32	expert model validated on data	Overfitting confirmation bias confirmation bias no factual basis
GYMIC (Kiani et al., 2019)	sepsis in intensive care	MIMIC (Johnson et al., 2021)	40 ,000	behavior cloning	
Virtu-ALS (Brisk et al., 2018)	emergency care				
Auto-ALS	emergency care				
HeartPole	healthy lifestyle				
GraphSim	intensive care	MIMIC (Johnson et al., 2021)	40 ,000	graph compression	

simulator's accuracy are reviewed in **Table 2**. *GYMIC* and *GraphSim* are the only simulators trained on a large dataset and GYMIC's accurracy has known overfitting issues. *GraphSim* is thus the most accurate of the simulators.

The most *transparent* simulator is clearly *HeartPole*. It does not aim to model any real clinical scenario accurately, but it can be a useful development tool to help scrutinize reinforcement learning algorithms.

As far as *difficulty* is concerned, *HeartPole* (Härmä et al., 2021), *simglucose* (Zhu et al., 2021), and *GYMIC* (Kiani et al., 2019) are known to be solvable with relatively small models and standard reinforcement learning algorithms like DQN (Mnih et al., 2015). Thus, the only simulators *difficult* enough to be benchmarks for novel approaches are *Virtu-ALS* and *Auto-ALS* and *Auto-ALS* is the more accessible of the two.

TABLE 3 | Summary of each simulator: POMDP view.

Simulator	O	A	$p_r(r = 0\|s) \neq 1$
simglucose	$[0; +\infty)$	[0; 35]	$S_{nt} \cup S_t$
GYMIC	$[0; 24]^{46}$	0, . . . , 24	$S_{nt} \cup S_t$
Virtu-ALS	$[0; 256]^{307\ 200}$	1, . . . , 307 200	$S_{nt} \cup S_t$
Auto-ALS	$[0; +\infty)^{36}$	1, . . . , 34	$S_{nt} \cup S_t$
HeartPole	$\mathcal{R}^6$	1, . . . , 4	$S_{nt} \cup S_t$
GraphSim	$(-\infty;+\infty)^{26}$	$[0; 1]^{317}$	S_t

Table 3 reviews the structural complexity of the simulators, a factor that directly contributes to *difficulty*. Note that *Virtu-ALS* is an unusually high-dimensional environment. As such,

solving it is likely to require more parameters and longer training times. *GraphSim* is the only simulator that doesn't provide non-zero rewards in non-terminal states S_{nt}, making it harder for the agent to attribute the results of the episode to particular actions. *GYMIC* (see **section 2.2**) solves this problem with an additional metric (SOFA score), but unlike *GYMIC GraphSim* covers a wide range of clinical conditions and there is no single health metric applicable to each.

5 CONCLUSION

Automatic discovery of clinical strategies is a nascent field of research that has a potential to considerably improve patient outcomes and become a new modus operandi in healthcare research. The goal of this paper is to provide a solid foundation for further development of this field with better patient simulators and better understanding thereof. We have reviewed the state of the art in patient simulators, identified some of the problems the field is facing and proposed novel simulators to address them. We believe that *HeartPole* and *Auto-ALS* can become new standard benchmarks for reinforcement learning in healthcare, while *GraphSim* can become a stepping stone to improved patient outcomes in intensive care.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: https://physionet.org/content/mimiciv/0.4/

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

FUNDING

This work was funded by the European Union's Horizon 2020 research and innovation programme under grant agreement n°812 882. This work is part of "Personal Health Interfaces Leveraging HUman-MAchine Natural interactionS" (PhilHumans) project: https://www.philhumans.eu

ACKNOWLEDGMENTS

We would like to thank *Rob Brisk* of (Brisk et al., 2018) for providing us with Virtu-ALS source code as well as his advice and support.

REFERENCES

American Heart Association Staff (2006). *Advanced Cardiovascular Life Support Provider Manual (International English)*. American Heart Association, Incorporated. ISBN 9781616690472. URL Available at: https://books.google.nl/books?id=oxdAygEACAAJ.

Barto, A. G., Sutton, R. S., and Anderson, C. W. (1983). Neuronlike Adaptive Elements that Can Solve Difficult Learning Control Problems. *IEEE Trans. Syst. Man. Cybern.* SMC-13 (5), 834–846. ISSN 2168-2909. doi:10.1109/TSMC.1983.6313077

Brisk, R., Bond, R. B., Liu, J. Y., Finlay, D., McLaughlin, J. A. D., and McEneaney, D. (2018). Ai to Enhance Interactive Simulation-Based Training in Resuscitation Medicine. In British HCI Conference 2018, BHCI2018 ; Conference date: 02-07-2018 Through 06-07-2018. 1–4. doi:10.14236/ewic/HCI2018.64

Cavalcante, R. C., Brasileiro, R. C., Souza, V. L. F., Nobrega, J. P., and Oliveira, A. L. I. (2016). Computational Intelligence and Financial Markets: A Survey and Future Directions. *Expert Syst. Appl.* 55, 194–211. doi:10.1016/j.eswa.2016.02.006

Chen, R. J., Lu, M. Y., Chen, T. Y., Williamson, D. F. K., and Mahmood, F. (2021). Synthetic Data in Machine Learning for Medicine and Healthcare. *Nat. Biomed. Eng.*, 1–5. doi:10.1038/s41551-021-00751-8

Clarke, W., and Kovatchev, B. (2009). Statistical Tools to Analyze Continuous Glucose Monitor Data. *Diabetes Technol. Ther.* 11 Suppl 1 (S1), S45–S54. doi:10.1089/dia.2008.0138

Gong, M., Xie, Y., Pan, K., Feng, K., and Qin, A. K. (2020). A Survey on Differentially Private Machine Learning [review Article]. *IEEE Comput. Intell. Mag.* 15 (2), 49–64. doi:10.1109/MCI.2020.2976185

Gulwani, S., Polozov, O., and Singh, R. (2017). Program Synthesis. *FNT in Programming Languages* 4 (1-2), 1–119. ISSN 2325-1107. doi:10.1561/2500000010

Härmä, A., Petković, M., Liventsev, V., and Simon, A. (2021). *Heartpole: A Transparent Task for Reinforcement Learning in Healthcare*. URL Available at: https://github.com/vadim0x60/heartpole/blob/master/HeartPole_abstract.pdf.

Hester, R. L., Brown, A. J., Husband, L., Iliescu, R., Pruett, D., Summers, R., et al. (2011). Hummod: A Modeling Environment for the Simulation of Integrative Human Physiology. *Front. Physio.* 2 (12). ISSN 1664-042X. doi:10.3389/fphys.2011.00012

Hochreiter, S., and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput.* 9 (8), 1735–1780. doi:10.1162/neco.1997.9.8.1735

James, J., Yoon, J., and Van Der Schaar, M. (2018). Pate-gan: Generating Synthetic Data with Differential Privacy Guarantees. In International conference on learning representations.

Jiang, W. (2021). Applications of Deep Learning in Stock Market Prediction: Recent Progress. *Expert Syst. Appl.* 184, 115537. doi:10.1016/j.eswa.2021.115537

Johnson, A., Bulgarelli, L., Pollard, T., Horng, S., Celi, L. A., and Mark, R. (2021). Mimic-iv (version 0.4).

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. (2016). Exploring the Limits of Language Modeling. arXiv preprint arXiv:1602.02410.

Kiani, A., Wang, C., and Xu, A. (2019). Sepsis World Model: A Mimic-Based Openai Gym" World Model" Simulator for Sepsis Treatment. arXiv preprint arXiv:1912.07127.

Koza, John. R. (1994). *Genetic Programming II*, 17. MA: MIT press Cambridge.

Krishnan, R. G., Shalit, U., and Sontag, D. (2015). Deep Kalman Filters. arXiv preprint arXiv:1511.05121.

Lambden, S., Laterre, P. F., Levy, M. M., and Francois, B. (2019). The SOFA Score-Development, Utility and Challenges of Accurate Assessment in Clinical Trials. *Crit. Care* 23 (1), 374–379. doi:10.1186/s13054-019-2663-7

Lee, J., Kim, H., Kim, K. H., Jung, D., Jowsey, T., and Webster, C. S. (2020). Effective Virtual Patient Simulators for Medical Communication Training: a Systematic Review. *Med. Educ.* 54 (9), 786–795. doi:10.1111/medu.14152

Li Deng, Li. (2012). The Mnist Database of Handwritten Digit Images for Machine Learning Research [best of the Web]. *IEEE Signal. Process. Mag.* 29 (6), 141–142. doi:10.1109/msp.2012.2211477

Liventsev, V., Aki, H., and Petković, M. (2021). Neurogenetic Programming Framework for Explainable Reinforcement Learning. doi:10.1145/3449726.3459537

Liventsev, V. (2021). Auto-als-v0. URL Available at: https://github.com/vadim0x60/auto-als.

Liventsev, V. (2021). Anthropodidactic Learning: a Modest Proposal. URL Available at: https://vadim.me/posts/anthropodidactic/.

Liventsev, V. (2021). Reinforcement Learning as Message Passing. URL Available at: https://vadim.me/posts/mpdp/.

Man, C. D., Micheletto, F., Lv, D., Breton, M., Kovatchev, B., and Cobelli, C. (2014). The UVA/PADOVA Type 1 Diabetes Simulator. *J. Diabetes Sci. Technol.* 8 (1), 26–34. ISSN 1932-2968. doi:10.1177/1932296813514502

McFetrich, J. (2006). A Structured Literature Review on the Use of High Fidelity Patient Simulators for Teaching in Emergency Medicine. *Emerg. Med. J.* 23 (7), 509–511. doi:10.1136/emj.2005.030544

Mischel, W., and EbbesenEbbesen, E. B. (1970). Attention in Delay of Gratification. *J. Personal. Soc. Psychol.* 16 (2), 329–337. doi:10.1037/h0029815

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., et al. (2013). Playing Atari with Deep Reinforcement Learning. arXiv preprint arXiv: 1312.5602.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level Control through Deep Reinforcement Learning. *nature* 518 (7540), 529–533. doi:10.1038/nature14236

Moore, A. W. (1990). *Efficient Memory-Based Learning for Robot Control*. Cambridge, UK: Technical report.

MühlbacherMühlbacher, A. C., and Juhnke, C. (2013). Patient Preferences versus Physicians' Judgement: Does it Make a Difference in Healthcare Decision Making? *Appl. Health Econ. Health Pol.* 11 (3), 163–180. doi:10.1007/s40258-013-0023-3

Nehring, W. M., and Lashley, F. R. (2004). Current Use and Opinions Regarding Human Patient Simulators in Nursing Education: An International Survey. *Nurs. Educ. Perspect.* 25 (5), 244–248.

Niazkar, M., and Niazkar, H. R. (2020). Covid-19 Outbreak: Application of Multi-Gene Genetic Programming to Country-Based Prediction Models. *Electron. J. Gen. Med.* 17 (5), 2020.

Price, W. N. (2018). Big Data and Black-Box Medical Algorithms. *Sci. Transl. Med.* 10 (471), eaao5333. doi:10.1126/scitranslmed.aao5333

Raghu, A., Komorowski, M., Ahmed, I., Celi, L., Peter, S., and Ghassemi, M. (2017). Deep Reinforcement Learning for Sepsis Treatment. arXiv preprint arXiv: 1711.09602.

Rankin, D., Black, M., Bond, R., Wallace, J., Mulvenna, M., and Epelde, G. (2020). Reliability of Supervised Machine Learning Using Synthetic Data in Health Care: Model to Preserve Privacy for Data Sharing. *JMIR Med. Inform.* 8 (7), e18910. doi:10.2196/18910

Street, R. L., Elwyn, G., and Epstein, R. M. (2012). Patient Preferences and Healthcare Outcomes: an Ecological Perspective. *Expert Rev. Pharmacoeconomics Outcomes Res.* 12 (2), 167–180. doi:10.1586/erp.12.3

Summers, R. L., Ward, K. R., Witten, T., Convertino, V. A., Ryan, K. L., Coleman, T. G., et al. (2009). Validation of a Computational Platform for the Analysis of the Physiologic Mechanisms of a Human Experimental Model of Hemorrhage. *Resuscitation* 80 (12), 1405–1410. doi:10.1016/j.resuscitation.2009.09.001

Sutton, R. S., and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT press.

Thim, T., Krarup, N. H., Grove, E., Rohde, C., and Lofgren, B. (2012). Initial Assessment and Treatment with the Airway, Breathing, Circulation, Disability, Exposure (Abcde) Approach. *Ijgm* 5, 117. doi:10.2147/ijgm.s28478

Walonoski, J., Kramer, M., Nichols, J., Quina, A., Moesel, C., Hall, D., et al. (2018). Synthea: An Approach, Method, and Software Mechanism for Generating Synthetic Patients and the Synthetic Electronic Health Care Record. *J. Am. Med. Inform. Assoc.* 25 (3), 230–238. doi:10.1093/jamia/ocx079

Wang, Z., Myles, P., and Tucker, A. (2021). Generating and Evaluating Cross-sectional Synthetic Electronic Healthcare Data: Preserving Data Utility and Patient Privacy. *Comput. Intelligence* 37 (2), 819–851. doi:10.1111/coin.12427

Watson, D. S., Krutzinna, J., Bruce, I. N., Griffiths, C. E., McInnes, I. B., Barnes, M. R., et al. (2019). Clinical Applications of Machine Learning Algorithms: beyond the Black Box. *BMJ* 364, l886, 2019 . ISSN 0959-8138. doi:10.1136/bmj.l886

Wendy, M. N., Ellis, W. E., and Lashley, F. R. (2001). Human Patient Simulators in Nursing Education: An Overview. *Simulation & Gaming* 32 (2), 194–204.

Xie, J. (2018). Simglucose V0. 2.1, 2018 URL Available at: https://github.com/jxx123/simglucose.

Yu, C., Liu, J., and Nemati, S. (2019). Reinforcement Learning in Healthcare: A Survey. *CoRR*. abs/1908.08796URL Available at: http://arxiv.org/abs/1908.08796.

Zhu, T., Li, K., and Georgiou, P. (2021). *Personalized Dual-Hormone Control for Type 1 Diabetes Using Deep Reinforcement Learning*. Cham: Springer International Publishing, 45–53. doi:10.1007/978-3-030-53352-6_5

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.