MACHINE REINFORCED PERTURBATION ON DRIFTED HUMAN LOGICAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Using deep neural networks as computational models to simulate cognitive process can provide key insights into human behavioral dynamics. This enables synthetic data generation to test hypotheses for neuroscience and guides adaptive interventions for cognitive regulation. Challenges arise when environments are highly dynamic, obscuring stimulus-behavior relationships. However, the majority of current research focuses on simulating human cognitive behaviors under ideal conditions, neglecting the influence of environmental disturbances. We propose ***ReactiveAgent***, integrating drift-diffusion with deep reinforcement learning to simulate granular effects of dynamic environmental stimuli on human logical reasoning process. This framework is built and evaluated upon our contributed large dataset of 21,157 logical responses of humans under various dynamic stimuli. Quantitatively, the framework improves cognition modelling by considering temporal effect of environmental stimuli on logical reasoning and captures both subject-specific and stimuli-specific behavioural differences. Qualitatively, it captures general trends in human logical reasoning under stress, better than baselines. Our approach is extensible to examining diverse environmental influences on cognitive behaviors. Overall, it demonstrates a powerful, data-driven methodology to simulate, align with, and understand the vagaries of human logical reasoning in dynamic contexts.

1 INTRODUCTION

Modeling human cognition is a fundamental challenge in understanding human behaviors (Jaffe et al. (2023)). In particular, modeling the effects of environmental dynamics (e.g., stress (Cheng (2017)) and feedback (Costa et al. (2019))) on cognitive performance could elucidate behavioral responses to tasks (Cheng (2017)) and inform the design of feedback mechanisms to augment cognition (Costa et al. (2019)). However, prior research (Jaffe et al. (2023); Ma & Peters (2020); Peterson et al. (2018); Battleday et al. (2021); Peterson et al. (2021)) predominantly concentrates on modeling human cognition under standard and ideal conditions, often neglecting the nuanced impact of environmental stimuli (Do et al. (2021); Park & Lee (2020)). Alternatively, some studies treat environmental stimuli as a constant presence throughout the cognitive process (Bourgin et al. (2019)).

We propose that a more nuanced modeling approach is imperative, particularly when dealing with dynamic stimuli that can fluctuate over time, contingent upon users' performance. This nuanced approach involves stimuli variation at fine timescales, exerting a continuous influence on human cognitive behaviors. To illustrate, consider an animated visual stimulus conveying time pressure (Slobounov et al. (2000)). Such stimuli inform users of the passage of time, evoking sensations of pressure. Representing these stimuli as a binary existence indicator would oversimplify their nuanced effects. Therefore, this paper raises a fundamental question: **How can we simulate the impact of dynamic environmental stimuli on the regulation of human cognitive behaviors with precision at a fine-grained level?**

We aim to address this question by examining how dynamic time pressure stimuli (Zur & Breznitz (1981)) influence cognitive performance, particularly within the context of a math arithmetic task—a widely utilized benchmark for evaluating human cognition and logical reasoning (Lin et al. (2011); Judd & Klingberg (2021); Daitch et al. (2016)). The *dynamism* inherent in time pressure feedback encompasses two primary facets. Firstly, the presentation of time pressure can be dynamic, involving the delivery of progressively changing visual frames over time (Fig. 4(a)), thereby instilling a sense of

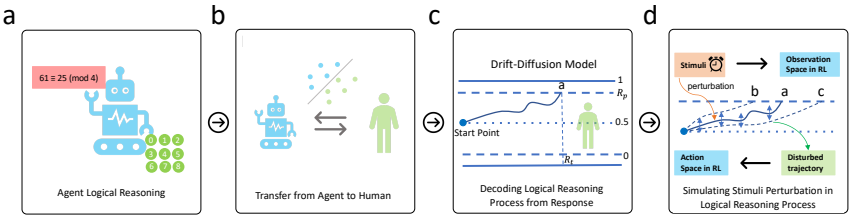

Figure 1: **Illustration of the overall framework**. First, we train a logical reasoning agent to solve cognitive tasks without considering users' response. Second, we transfer features extracted from the logical reasoning agent without time pressure to real user choice and response time (initial estimation). Third, the initial estimated response time and predicted choice probability generate evidence accumulation trajectory in the drift-diffusion model. Lastly, the DRL agent simulates influence of stimuli perturbation on cognitive process by taking dynamic environmental stimuli as input and take specific action to modulate evidence accumulation process. When evidence accumulator achieves boundary threshold, the final prediction of response time is generated and DRL agent achieves terminate state.

urgency. Secondly, the presence of time pressure may vary dynamically across different trials. Since time pressure stimuli represent a well-established feedback modality capable of modulating human cognitive performance (Cheng (2017); Slobounov et al. (2000); Moore & Tenney (2012); Edland & Svenson (1993); Whittaker et al. (2016)), the modeling of cognition performance under such dynamic time pressure holds the promise of offering valuable insights into cognitive behaviors and facilitating the development of adaptive intervention mechanisms for regulating user performance.

In this paper, we introduce a systematic hybrid framework (***ReactiveAgent***) depicted in Fig. 1. This framework integrates a classical closed-form cognitive model into a data-driven deep reinforcement learning (DRL) approach, allowing for a comprehensive and explainable simulation of the impacts of dynamic, fine-grained time pressure stimuli. While neural networks (NNs) are recognized for their proficiency in function approximation and have been applied to model cognitive behaviors (Bourgin et al. (2019)), their inherent black-box nature poses challenges in representing the internal mechanisms of the cognitive process. To address this limitation, our framework integrates DRL with the drift-diffusion model (DDM), a sequential sampling method widely employed in cognitive modeling (Ratcliff & McKoon (2008); Steyvers et al. (2019)). DDM posits that humans make decisions by accumulating evidence until reaching a boundary threshold (Fudenberg et al. (2020)). The simulated choice and response time are then determined based on the corresponding boundary and accumulation time. While DDM excels in representing the cognition process in an explicable and fine-grained manner, it primarily focuses on posterior estimation of user decisions rather than predicting users' future performance under stimuli. On the other hand, DRL, with NNs at its core, offers a step-by-step interaction environment. This environment enables the incorporation of the fine-grained cognition process inherent in DDM while retaining the function approximation capabilities of NNs. This hybrid approach bridges the gap between the transparency of classical cognitive models and the flexibility of data-driven methods, presenting a promising avenue for modeling the intricate dynamics of cognition under dynamic time pressure stimuli. Our contribution is three-folded:

- We proposed ***ReactiveAgent***, a hybrid framework to incorporate classical cognition models (drift-diffusion model) with deep reinforcement learning to simulate the perturbation of environmental stimuli on the evidence accumulation process in human logical reasoning.
- We comprehensively demonstrate and explain the effectiveness of our framework in reducing response time simulation error by comparing with a series of baseline models and running several ablation studies.
- We open-source a large dataset including 21,157 logical reasoning responses collected from humans who experienced various dynamic environmental visual feedback, as well as the task and stimuli information in both text and video format. All codes and datasets are available at: https://github.com/Reactive-Agent/ReactiveAgent

2 Related Work

Cognitive process models. The existing literature has amassed empirical evidence supporting feasibility of modeling human cognition (De Boeck & Jeon (2019)). Traditional cognitive models, exemplified by BEAST (Erev et al. (2017)) and the drift-diffusion model (DDM) (Ratcliff & McKoon (2008); Steyvers et al. (2019)), are characterized by closed-form structures. For example, DDM (Ratcliff & McKoon (2008)) treats cognitive process as an evidence accumulation process for humans to make decisions, so as to simulate the speed-accuracy tradeoff (Heitz (2014)) in human behaviors.

Cognitive simulation with machine learning. More recently, there has been a notable shift toward the integration of machine learning techniques (Cichy & Kaiser (2019)) for simulating human behaviors (Peysakhovich & Naecker (2017); Lake et al. (2017); Ma & Peters (2020)) across an array of tasks, including visual cognition (Cho et al. (2023); Wenliang & Seitz (2018); Mehrer et al. (2020)), categorization (Battleday et al. (2017; 2021)), decision making (Binz & Schulz (2024); Hosoya; Peterson et al. (2021); Bourgin et al. (2019)), game strategy (Hartford et al. (2016)), human exploration (Binz & Schulz (2022)), word learning (Ritter et al. (2017)), probabilistic inference (Orhan & Ma (2017)), point-and-click interactions (Do et al. (2021); Park & Lee (2020)), and others.

Response time simulation. Of particular note, recurrent neural networks (RNNs) (Jaffe et al. (2023); Song et al. (2017; 2016)) have been adapted to execute various cognitive tasks (Yang et al. (2019)) emulating human performance and the intricate balance between accuracy and response time observed in biological vision (Spoerer et al. (2020)). Recently, (Goetschalckx et al. (2024)) computed the human-like reaction time from convolutional RNN, leveraging evidential deep learning (Sensoy et al. (2018)). And task-DyVA (Jaffe et al. (2023)) modelled cognitive response time with RNN-based latent dynamical systems in task-switching games.

In spite of these existing models to simulate response time and human task performance, there is limited work focusing on modelling external stimuli perturbation (such as environmental stress) on task performance. One example is (Bourgin et al. (2019)), which treats environmental stimuli as a constant presence throughout the cognitive process. However, we believe a more nuanced modeling approach is imperative, particularly when dealing with dynamic external stimuli that can fluctuate over time, contingent upon users' performance. This nuanced approach involves stimuli variation at fine timescales, exerting a continuous influence on human cognitive behaviors.

3 Task and Dataset

As depicted in Section. 1, we used a math arithmetic task with time pressure visual stimuli as our model exploration context. The illustration of the task and stimuli is depicted in Appendix Fig. 4. In short, each math trial was composed of two two-digit numbers Num_1, Num_2 and one one-digit number Num_3, formatted as: $Num_1 \equiv Num_2 \,(\, mod\, Num_3)$. To solve this question, participants first used Num_1 to subtract Num_2 and judged whether the subtraction result could be divisible by Num_3. If it was divisible, they selected "True" button. Otherwise, they selected "False" button. When the time pressure stimuli happened, a progress bar was shown on top of the math question, which added one unit for each second and reset and added again when it accumulated five units. The human response time was then calculated from the time when the math task appeared per trial, to the time when the participants clicked one button to answer it.

We collected an extensive dataset encompassing 21,157 valid responses from 44 participants engaged in the math task (see Fig. 5(a)). To enhance dataset diversity and evaluate our model under dynamic environmental stress, participants were randomly and uniformly distributed across four distinct groups: ***None*** Group: Participants experienced no time pressure for any trial. ***Static*** Group: Time pressure was consistently applied for each trial. ***Random*** Group: There was a 50% probability of time pressure being applied for each trial. ***Rule*** Group: Time pressure was adaptively applied based on users' past performance using a rule-based strategy (more details of such strategy are in Appendix A.1.4). Each participant engaged in a two-day study, featuring one exercise session (20 trials) and one formal session (300 trials) per day, when we collected participants' choices and response time per trial. This collection has been approved by the Institutional Review Board (IRB) in our local institution. We do not anticipate any risk during data collection and we have obtained informed consent from all participants beforehand. More dataset details are in Appendix A.1.

4 MODEL AND METHODOLOGY

Similar to (Goetschalckx et al. (2024)), our framework aims to simulate human response time in the task instead of human choice accuracy. Our dataset analysis in Appendix A.2 also showed that human accuracy was not affected by the external stimuli since our experimenter asked participants to prioritize accuracy rather than answering speed to control speed-accuracy tradeoff (Heitz (2014)).

4.1 REACTIVEAGENT FRAMEWORK

Inspired by exploratory analysis (Appendix A.2) and existing cognitive theories (Roseboom et al. (2019); Yang et al. (2019); Mickey & McClelland (2014)), our framework comprises four key steps, as illustrated in Fig. 1. In the initial step, we train a long short-term memory (LSTM)-based logical reasoning agent to proficiently solve the designated cognitive task. Then the second step involves the knowledge transfer from these trained agents to establish mappings from the LSTM agent to human performance metrics. This results in human response time and accuracy for each trial. Moving to the third step, we employ a fine-grained Drift-Diffusion Model (DDM) to decode human performance, extracting detailed information about response time and accuracy. This step is pivotal in generating the evidence accumulation process (EA) reflective of the underlying cognitive mechanisms. In the final step, we introduce a deep reinforcement learning (DRL) agent to the framework. This agent plays a crucial role in simulating the impact of stimuli perturbation on the evidence accumulation process. By leveraging DRL, we can capture the nuanced dynamics of how external stimuli, such as time pressure, influence the intricate logical reasoning processes modeled by the DDM. We describe details of the first two steps in Section. 4.2. and the last two steps in Section. 4.3.

4.2 MATH LOGICAL REASONING AGENT AND TRANSFER TO HUMANS

To simulate the impact of time pressure, it is imperative to first predict users' baseline performance in ideal conditions without time pressure. Drawing inspiration from prior research that models human subjects' time perception by capturing internal activities in perceptual classification networks (Roseboom et al. (2019)), we have devised a baseline prediction model. Specifically, Roseboom et al. (Roseboom et al. (2019)) constructed a neural network **functionally akin to human visual processing for image classification**. The network was then exposed to input videos of natural scenes, causing changes in network activation. The accumulation of salient changes in activation was subsequently used to estimate duration, effectively gauging the perceived passage of time in the video through a Support Vector Machine (SVM).

Similarly, our baseline prediction model employs an LSTM neural network to address cognitive tasks (Yang et al. (2019)). In particular, we train an LSTM-based math answer agent (Fig. 1(a)) to learn and respond to math questions, thereby achieving **functional similarity with human cognition in math tasks** (Yang et al. (2019)). The intermediate output of the LSTM layer serves as input features for the SVM, establishing mappings between agents and humans to estimate user choice and response time (Fig. 1(b)). The rationale of this approach is that distinct math questions may pose varying levels of difficulty, leading to user choice biases and variations in response time (Hanich et al. (2001)). The LSTM-based agent has the capacity to capture these potential differences in difficulty levels (Mickey & McClelland (2014); Zaremba & Sutskever (2014)), and the SVM is employed to map these to user choice (via the SVC model) and response time (via the SVR model). More rationales of the math answer agent and SVM models are in Appendix A.3, A.4, Fig. 8.

4.3 HYBRID DRL AGENT TO SIMULATE STIMULI PERTURBATION

To simulate how dynamic time pressure perturbs human logical reasoning process, we conceptualize the logical reasoning process as an evidence accumulation (EA) process in line with the Drift-Diffusion Model (DDM) (Ratcliff & McKoon (2008)) (Fig. 1(c)). The EA process segments users' cognition into sequential steps, facilitating the fine-grained modeling of dynamic time pressure. The boundary threshold and accumulation time parameters in the DDM are derived from the predicted responses obtained from the previous SVM model. In order to simulate the dynamic impact of time pressure visual stimuli, we introduce a DRL agent. The visual stimuli are segmented into frames, aligning with the steps in the EA process. For each frame, the specific visual stimuli are applied to the DRL agent (Fig. 1(d)), which, akin to how participants' logical reasoning processes may be

Table 1: Examples of baseline model performance on response time simulation. For MAPE, we show its mean value (Mean), standard deviation (STD), 2.5th (Lower) and 97.5th (Upper) percentiles of the MAPE distribution (95% confidence interval). For results in all baseline models, see Table. 3.

		MAPE			
Model Input Type	Model Type Name	Mean	STD	Lower	Upper
Task: Video Feedback: Video	hGRU	0.3335	0.2486	0.0153	0.9406
	LSTM + ViT-B-16	0.3339	0.2573	0.0145	0.9852
	MLP + 3D ResNet	0.3330	0.2507	0.0121	0.9390
Task: Encoded String Feedback: Video	MLP + 3D ResNet	0.3331	0.2550	0.0125	0.9601
	Transformer + 3D ResNet	0.3306	0.2496	0.0145	0.9462
	ReactiveAgent	**0.2999**	**0.2318**	**0.0131**	**0.8029**
Task: Numeric Value Feedback: Video	LSTM-V1 + 3D ResNet	0.3341	0.2617	0.0152	0.9923
	LSTM-V2 + 3D ResNet	0.3286	0.2538	0.0147	0.9707
	Transformer + 3D ResNet	0.3315	0.2526	0.0152	0.9615
Task: Numeric Value Feedback: Numeric Value	MLP	0.3293	0.2441	0.0151	0.9257
	SVM	0.3299	0.3108	0.0113	1.1827
	XGBoost	0.3508	0.3469	0.0112	1.3075
Task: Encoded String Feedback: Numeric Value	Linear Regression	0.3512	0.3469	0.0105	1.3176
	LSTM	0.3278	0.2478	0.0142	0.9397
	Random Forest	0.3600	0.3630	0.0130	1.3620

influenced by each frame of stimuli, modulates the EA process. In particular, for each frame of time pressure stimuli, the DRL agent adjusts the EA process by introducing positive, neutral, or negative bias (action space of the DRL agent). This modulation may result in the evidence accumulator reaching the boundary threshold either earlier or later. The output from this DRL-modulated EA process serves as the final prediction for user response time. More details are in Appendix A.6.

5 Evaluation

5.1 Human response time simulation performance

We first demonstrate the effectiveness of our ***ReactiveAgent*** framework in human response time simulation by comparing with baseline models using different stimuli encoding schemes. The model input is composed of three parts: math task stimuli, time pressure environmental feedback stimuli, and task question ID. The question ID is in numeric value to indicate the trial number for participants in the math task. Our exploratory analysis in Appendix A.2 has depicted the relevance of question ID in human response time. In ***ReactiveAgent***, the math task stimuli are represented by one-hot encoded textual strings and the feedback stimuli are represented by videos. However, there are also different ways to extract features from the model input. For example, we can treat both task stimuli and feedback stimuli as numeric value directly or we can put both math task stimuli and feedback stimuli into the whole video as model input, just like what humans watch in the task. Therefore, we traverse five types of model input to represent the features of task stimuli and feedback stimuli and use corresponding baseline models, as depicted in Table. 1 and Table. 3. More details of each model input type and the training/testing hyperparameters/process are depicted in Appendix A.7. When encoding both task stimuli and feedback stimuli into a whole video, we use hGRU (Goetschalckx et al. (2024)), LSTM with pre-trained vision models (Jaffe et al. (2023)), and MLP with pre-trained 3D ResNet (Bourgin et al. (2019)) as the baseline. These models are adapted into our problem corresponding to the recent State-of-the-Art (SOTA) models in human decision making (Bourgin et al. (2019)) and response time prediction (Goetschalckx et al. (2024); Jaffe et al. (2023)). Similarly, for other types of model input, we also use related SOTA models in the specific input type domain. More details of baseline models and adaptation into our problems are depicted in Appendix A.7.

We use Mean Average Percentage Error (MAPE) instead of Mean Squared Error (MSE) to evaluate the response time difference between real humans and simulations because human response time

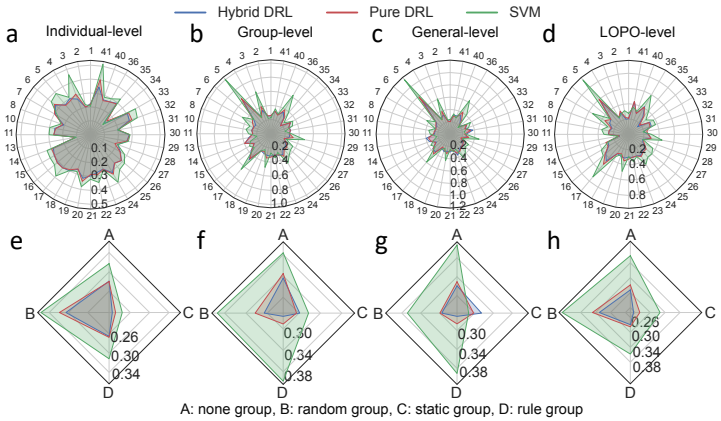

Figure 2: a,b,c,d,e,f,g,h: Average MAPE for each participant (a,b,c,d)/group (e,f,g,h) in predictions of testing set from Hybrid DRL agent, Pure DRL agent, and SVM model in four training strategies (a,e. Individual-Level, b,f. Group-Level, c,g. General-Level, d,h. LOPO-Level), respectively. (The number around the circle represents participant id in a,b,c,d).

comes with high individual differences (Faust et al. (1999)). Therefore, for deep learning models, we use MAPE loss function instead of MSE loss function. Training details are in Appendix A.7.

Results are depicted in Table. 1 and Table. 3, showing that ***ReactiveAgent*** has the best response time prediction performance (lowest MAPE) by comparing with other models in both the same and different model input types. Such performance improvement benefits from the whole framework including useful extracted features from the math logical reasoning agent and the integration of the drift-diffusion model in the DRL agent to simulate feedback stimuli in a fine-grained manner. In what follows, we run ablation studies to show the importance of each component in our framework.

5.2 Importance of task encoding with the math logical reasoning agent

To demonstrate that the math answer agent has indeed captured useful and representative features from the math questions, in the first ablation study, we compare the SVM models (the second step in our framework) with two additional settings where the SVM models do not take features captured from the math answer agent as input. Instead, they take raw three-digit numbers from the math questions or one-hot encoded vectors (same as the input of the math logical reasoning agent in Appendix A.3) of raw numbers as input, along with the question id. The SVM performance in the three settings is depicted in Table. 2. To evaluate the SVM-based classifier (SVC), we use accuracy, precision, recall, and F1-score to measure the accuracy to predict user choice. For the SVM-based regression model (SVR), we use MAPE to measure the error to predict user response time. Notably, SVM models with features from the math answer agent exhibit significantly higher accuracy (0.9613) and F1-score (0.8996) for user choice prediction and lower MAPE (0.3652) for response time estimation than the other two settings. These results underscore the effectiveness of the math answer agent in capturing representative features from math questions and the feasibility of predicting user baseline performance in ideal conditions without environmental stimuli with the SVM models.

5.3 Why does the logical reasoning agent work?

The second ablation study explores why the math logical reasoning agent could extract useful features from math questions (in the first step of our framework). We answer this question by exploring its math task solving performance under different number of output neurons from the LSTM layer.

Note that the math answer agent aims to solve math tasks correctly instead of predicting human choice. In short, given one math question as input, it could directly output the arithmetic reasoning answer. Therefore, its training and testing have no correlation with real users' response. Hence, we prepare a separate dataset that is independent from humans' dataset to train the agent. Finally, we traverse all possible combinations of three numbers in math questions and get a dataset including

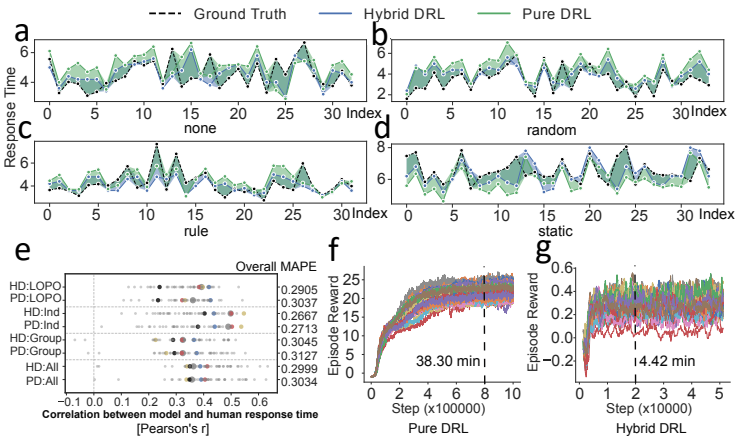

Figure 3: a,b,c,d: Examples of user response time in chronological order from one participant in each group predicted from Hybrid/ Pure DRL agent in LOPO-level training, compared with ground truth. e: Pearson correlation between predictions from Hybrid/ Pure DRL agent (HD: Hybrid DRL, PD: Pure DRL) and human real response time (ground truth) in four training strategies (All: General-level, Group: Group-level, Ind: Individual-level, LOPO: LOPO-level). Small gray dots, medium dots, and large gray dots represent Pearson correlation of prediction results from each participant's testing set, each group's testing set (red:*none*, yellow:*static*, black:*random*, blue:*rule*) and whole testing set, respectively. The right y axis depicts overall average MAPE of two agents in four training strategies. f,g: Training curve for Pure DRL (f) and Hybrid DRL (g) model.

20414 samples, which is split into training set (80%) and testing set (20%). Given that the first two numbers of math questions are both two-digits, the arithmetic reasoning result is chosen from 0 to 8. Consequently, the ground truth encompasses 9 classes.

We experimented with different numbers of output neurons (32, 64, 128, 256) from the LSTM layer. After 100 epochs of training, the logical reasoning agent with 256 neurons achieved remarkable results, attaining a training loss of 0.0001 and 100% accuracy (Fig.6(b)). The confusion matrix (Fig.6(a)) for the testing set also demonstrates that this neuron configuration yields over 99% accuracy for all classes, resulting in an overall test accuracy of 99.93%. Moreover, even for other neuron number, the test accuracy is also high enough (more than 95%). These outcomes affirm that the LSTM-based logical reasoning agent adeptly solves math arithmetic problems in the majority of cases. This aligns with existing work (Mickey & McClelland (2014); Zaremba & Sutskever (2014)), which demonstrated the capacity of neural networks to learn mathematical equivalence. The success of the logical reasoning agent in solving arithmetic problems lays a foundation and explains its capability for extracting representative features from math questions to construct cognition models.

5.4 Importance of integrating the DDM into DRL agents

The third ablation study examines the importance of DDM in DRL agents, to simulate the perturbation of external stimuli on human response time in a fine-grained manner. We introduce a baseline DRL model called "Pure DRL agent", which does not incorporate the DDM. Specifically, this Pure DRL agent does not segment time pressure visual stimuli into frames. Instead, for each trial from the dataset, if the time pressure stimuli exist, it directly takes the entire visual stimuli as input and outputs one action representing the overall change in response time due to time pressure. The final estimation of regulated response time is the sum of this action and the basic response time estimated by the SVR models (see details in Appendix A.5 and Fig. 7). Moreover, we also directly remove the whole Hybrid DRL agent and only use the SVR models to predict response time as another ablation baseline.

We employ both MAPE and Pearson correlation to compare the performance of the hybrid DRL and Pure DRL agents. Four model training strategies are used for comparison: **general-level**, **group-level**, **individual-level**, and **Leave-One-Participant-Out (LOPO)-level**. General-level involves splitting the entire dataset into training (80%) and testing (20%) sets for overall model evaluation. Group-level trains and tests a specific model using data from each group, revealing performance

across different time pressure stimuli. Individual-level trains and tests a model using data from a specific participant, assessing personalized model feasibility incorporating subject-specific behavioural differences. Shuffling is applied to the training and testing sets to prevent overfitting artifacts. As shuffled testing disrupts the temporal trend of user response time across different math trials, we incorporate LOPO-level as an additional training strategy. This strategy selects all data from one participant as the testing set and uses data from other participants in the same group as the training set. By traversing every participant's data as the testing set, we ensure a comprehensive assessment of the model's performance in capturing the temporal trend of response time.

Fig. 2 illustrates the average MAPE of the testing set for each individual user (a,b,c,d) and each group (e,f,g,h). Both the hybrid DRL and Pure DRL agents show improvement in response time estimation compared to SVM results. However, the hybrid DRL agent consistently achieves lower MAPE compared to the Pure DRL agent in most cases, indicating the superiority of the hybrid DRL agent in response time estimation. The overall average MAPE for the entire testing set by both agents is depicted on the right y-axis of Fig. 3(e), further supporting this conclusion. Fig. 3(e) also reveals that the hybrid DRL agent exhibits larger Pearson correlation in individual testing sets (small dots), group testing sets (medium dots), and the whole testing set (large dots) compared to the Pure DRL agent in most cases, across all four training strategies. Both MAPE and Pearson correlation demonstrate the superior performance of the hybrid DRL agent in modeling the effect of time pressure stimuli.

To compare which agent design better captures the trend of response time change in users' overall tasks, we visualize the prediction results and real user response time for the testing set from one participant of each group in LOPO-level in chronological order (Fig. 3(a,b,c,d)). The resulting curves clearly demonstrate that the hybrid DRL agent more accurately captures the trend of user response time compared to the Pure DRL agent.

5.5 Training efficiency

The training curves for both Hybrid and Pure agents are presented in Fig. 3(f,g). The Pure DRL and hybrid DRL agents converge at approximately 800,000 steps and 20,000 steps, respectively. It is important to note that the meanings of one step differ between the two agents. For the hybrid DRL agent, one step represents one frame of time pressure stimuli during one trial, whereas one step for the Pure DRL agent represents the entire trial. Consequently, a direct comparison of steps is not meaningful. Instead, we compare the training time required for both agents to achieve convergence on the same hardware (GeForce RTX 2080 Ti) and the same dataset. The results in Fig. 3(f,g) indicate that the hybrid DRL agent converges in approximately 1/10 of the time compared to the Pure DRL agent (4.42 minutes vs. 38.30 minutes). This outcome underscores the advantage of incorporating an explicit cognitive model (i.e., the DDM) in the hybrid DRL agent.

5.6 Interpretability

An essential advantage of the cognition-inspired hybrid DRL agent is its interpretability, compared with deep learning models and the pure DRL agent, which directly output estimated response time changes for each trial, obscuring the internal mechanism regarding how time pressure stimuli modulate the logical reasoning process. In contrast, the hybrid DRL agent can generate a trajectory of the time pressure effect on response time corresponding to the users' logical reasoning process. Therefore, visualizing the trajectories of the hybrid DRL agent enables the extraction of new insights into how time pressure stimuli affect the human logical reasoning process.

We explore this benefit in Fig. 9(a,b,c,d,e,f,g,h). Here, the *action trajectory* represents the trajectory of actions taken by the hybrid DRL agent during one episode, with each episode corresponding to one math trial of users. The *time pressure effect trajectory* is the accumulated actions multiplied by δ_p. δ_p represents one unit of evidence per step, transforming the normalized action value into the evidence accumulation process. We visualize the time pressure effect trajectories across the four groups in Fig. 9(a,b,c,d). Each curve represents one trajectory predicted by the hybrid DRL agent during one trial.

We observe that the time pressure effect trajectories are more concentrated in the *random* and *rule* groups but divergent in the *none* and *static* groups (Fig. 9(a,b,c,d)). This suggests that participants in the *random* and *rule* groups, especially the *random* group, are better regulated by the corresponding type of time pressure stimuli, resulting in similar trends in all time pressure effect trajectories in

this group. Quantitatively, the *random* group has the lowest standard deviation (STD) of action trajectories (Fig. 9(g)) and the highest average value and slope for the time pressure effect trajectories (Fig. 9(f,h)). These findings in the simulation results indicate that the *random* group experiences the most effective regulation of user cognition performance.

This observation aligns with the expectation that users may quickly adapt to *none* or *static* time pressure, ceasing to be regulated by them after a few trials. However, users may not anticipate the time pressure in *random* group, leading to a more prolonged regulation effect. This result in the hybrid DRL simulation is also consistent with real human results in our initial exploratory findings (Appendix A.2, Fig. 5(e)), where participants in *random* group demonstrated a significantly larger reduction in response time, compared with other groups. These experiments affirm the hybrid DRL agent's capability to explain and support observations in the real humans' response time performance.

The comparative analysis between the hybrid and pure DRL agent designs across three key aspects (response time estimation performance, agent training efficiency, and interpretability) highlights the advantages and effectiveness of the hybrid DRL approach in capturing the nuanced dynamics of time pressure stimuli on user response time in logical reasoning process.

6 Discussion, Limitations and Future Work

Our investigation contributes new insights and lays the groundwork for cognition models integrating machine learning within dynamic environments. These models hold promise for advancing our comprehension of cognitive and social behaviors in humans amid dynamic environmental contexts, thereby elucidating behavioral responses to tasks (Cheng (2017)) and propelling advancements in cognition modeling for diverse applications, including human decision prediction (Bourgin et al. (2019)), generating synthetic datasets (Zweifel et al. (2021)), and informing the design of feedback mechanisms aimed at enhancing cognition (Costa et al. (2019)). We further discuss limitations below.

One limitation is that our evaluation used our own dataset rather than public ones. Most existing datasets, such as Lumosity (Steyvers et al. (2019)), focus on user performance in ideal conditions, without environmental stimuli. Since these datasets don't account for the effects of external stimuli on human performance, they couldn't be used for our study. To address this, we collected a new, large dataset that accounts for external stimuli and contribute it to the research community.

Another limitation is that we only explored one math task to assess human logical reasoning under time pressure, although it is not rare to use one cognitive task to study deep learning models such as Bourgin et al. (2019). Moreover, our framework can be extended into more diverse tasks and external stimuli. Specifically, our methodology involves training a task-solving agent, such as the logical reasoning agent in this study, to mimic human task performance. Previous research has demonstrated the effectiveness of machine learning models in solving various cognitive tasks (Yang et al. (2019)). Machine learning models like SVM can then be used to link the agent's extracted features with real user responses. A DRL agent is used to simulate the effects of dynamic stimuli on the evidence accumulation process. This framework is adaptable to stimuli beyond environmental stress. For visual stimuli, images or videos can be introduced into the DRL agent's observation space, and other types of stimuli, like auditory signals, can also be incorporated by adjusting the input modalities.

Furthermore, DDM can be extended to multiple-choice tasks, as in Steyvers et al. (2019), by accumulating evidence for each choice and setting corresponding thresholds. For continuous choices, discretization can be applied to use the multi-choice model.

7 Conclusion

We propose ***ReactiveAgent***, a computational framework to simulate environmental stimuli perturbation on humans' logical reasoning process, in the context of a math arithmetic task under time pressure visual feedback. Our framework achieves more accurate simulation with higher training efficiency and interpretability by integrating the drift-diffusion model from cognitive science into deep reinforcement learning, to simulate the granular effect of the dynamic stimuli on human logical reasoning process. Our comprehensive experiment demonstrates the advantages and effectiveness of our framework. We believe that this framework could bring new insights in both machine learning and neuroscience to build computational models to understand human cognitive behaviors.

8 Ethics Statement

Our experiment for human data collection in logical reasoning tasks was approved by the Institutional Review Board (IRB) in our local institution. We do not anticipate any risk during data collection and we have obtained informed consent from all participants beforehand. Our work may provide insights to integrate classical cognitive theories into machine learning models. In neuroscience, effective computational models for response time could pave the way for understanding many key cognitive behaviors and neurobiological disorders(Goetschalckx et al. (2024); Huys et al. (2016)). We do not anticipate the negative impact on society in this context.

9 Reproducibility Statement

We have described high-level details about models and experiments in the main text and we have included more details in Appendix so that the results are reproducible and verifiable. We have also open-sourced our datasets and codes (https://github.com/Reactive-Agent/ReactiveAgent) to further enhance the reproducibility.

References

Collie Alexander, Maruff Paul, Mcstephen Michael, et al. The effects of practice on the cognitive test performance of neurologically normal individuals assessed at brief test–retest intervals. *Journal of the International Neuropsychological Society*, 9(3):419–428, 2003.

Ruairidh M Battleday, Joshua C Peterson, and Thomas L Griffiths. Modeling human categorization of natural images using deep feature representations. *arXiv preprint arXiv:1711.04855*, 2017.

Ruairidh M Battleday, Joshua C Peterson, and Thomas L Griffiths. Capturing human categorization of natural images by combining deep networks and cognitive models. *Nature communications*, 11 (1):1–14, 2020.

Ruairidh M Battleday, Joshua C Peterson, and Thomas L Griffiths. From convolutional neural networks to models of higher-level cognition (and back again). *Annals of the New York Academy of Sciences*, 1505(1):55–78, 2021.

Marcel Binz and Eric Schulz. Modeling human exploration through resource-rational reinforcement learning. *Advances in neural information processing systems*, 35:31755–31768, 2022.

Marcel Binz and Eric Schulz. Turning large language models into cognitive models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=eiC4BKypf1.

David D Bourgin, Joshua C Peterson, Daniel Reichman, Stuart J Russell, and Thomas L Griffiths. Cognitive model priors for predicting human decisions. In *International conference on machine learning*, pp. 5133–5141. PMLR, 2019.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Shyh-Yueh Cheng. Evaluation of effect on cognition response to time pressure by using eeg. In *International conference on applied human factors and ergonomics*, pp. 45–52. Springer, 2017.

Junmo Cho, Jaesik Yoon, and Sungjin Ahn. Spatially-aware transformers for embodied agents. In *The Twelfth International Conference on Learning Representations*, 2023.

François Chollet et al. Keras. https://keras.io, 2015.

Radoslaw M Cichy and Daniel Kaiser. Deep neural networks as scientific models. *Trends in cognitive sciences*, 23(4):305–317, 2019.

Jean Costa, François Guimbretière, Malte F Jung, and Tanzeem Choudhury. Boostmeup: Improving cognitive performance in the moment by unobtrusively regulating emotions with a smartwatch. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(2):1–23, 2019.

Amy L Daitch, Brett L Foster, Jessica Schrouff, Vinitha Rangarajan, Itır Kaşikçi, Sandra Gattas, and Josef Parvizi. Mapping human temporal and parietal neuronal population activity and functional coupling during mathematical cognition. *Proceedings of the National Academy of Sciences*, 113 (46):E7277–E7286, 2016.

Paul De Boeck and Minjeong Jeon. An overview of models for response times and processes in cognitive tests. *Frontiers in psychology*, 10:102, 2019.

Seungwon Do, Minsuk Chang, and Byungjoo Lee. A simulation model of intermittently controlled point-and-click behaviour. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–17, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Anne Edland and Ola Svenson. Judgment and decision making under time pressure. In *Time pressure and stress in human judgment and decision making*, pp. 27–40. Springer, 1993.

Ido Erev, Eyal Ert, Ori Plonsky, Doron Cohen, and Oded Cohen. From anomalies to forecasts: Toward a descriptive model of decisions under risk, under ambiguity, and from experience. *Psychological review*, 124(4):369, 2017.

Mark E Faust, David A Balota, Daniel H Spieler, and F Richard Ferraro. Individual differences in information-processing rate and amount: implications for group differences in response latency. *Psychological bulletin*, 125(6):777, 1999.

Drew Fudenberg, Whitney Newey, Philipp Strack, and Tomasz Strzalecki. Testing the drift-diffusion model. *Proceedings of the National Academy of Sciences*, 117(52):33141–33148, 2020.

Lore Goetschalckx, Lakshmi Narasimhan Govindarajan, Alekh Karkada Ashok, Aarit Ahuja, David Sheinberg, and Thomas Serre. Computing a human-like reaction time metric from stable recurrent vision models. *Advances in Neural Information Processing Systems*, 36, 2024.

Tal Golan, Prashant C Raju, and Nikolaus Kriegeskorte. Controversial stimuli: Pitting neural networks against each other as models of human cognition. *Proceedings of the National Academy of Sciences*, 117(47):29330–29337, 2020.

Jun Han and Claudio Moraga. The influence of the sigmoid function parameters on the speed of backpropagation learning. In *Proceedings of the International Workshop on Artificial Neural Networks: From Natural to Artificial Neural Computation*, IWANN '96, pp. 195–201, Berlin, Heidelberg, 1995. Springer-Verlag. ISBN 3540594973.

Laurie B Hanich, Nancy C Jordan, David Kaplan, and Jeanine Dick. Performance across different areas of mathematical cognition in children with learning difficulties. *Journal of educational psychology*, 93(3):615, 2001.

Jason S Hartford, James R Wright, and Kevin Leyton-Brown. Deep learning for predicting human strategic behavior. *Advances in neural information processing systems*, 29, 2016.

Richard P Heitz. The speed-accuracy tradeoff: history, physiology, methodology, and behavior. *Frontiers in neuroscience*, 8:86875, 2014.

Haruo Hosoya. A cognitive model for learning abstract relational structures from memory-based decision-making tasks. In *The Twelfth International Conference on Learning Representations*.

Quentin JM Huys, Tiago V Maia, and Michael J Frank. Computational psychiatry as a bridge from neuroscience to clinical applications. *Nature neuroscience*, 19(3):404–413, 2016.

Paul I Jaffe, Russell A Poldrack, Robert J Schafer, and Patrick G Bissett. Modelling human behaviour in cognitive tasks with latent dynamical systems. *Nature Human Behaviour*, pp. 1–15, 2023.

Nicholas Judd and Torkel Klingberg. Training spatial cognition enhances mathematical learning in a randomized study of 17,000 children. *Nature Human Behaviour*, 5(11):1548–1554, 2021.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Omkar Kumbhar, Elena Sizikova, Najib Majaj, and Denis G Pelli. Anytime prediction as a model of human reaction time. *arXiv preprint arXiv:2011.12859*, 2020.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

Angela M Legg and Lawrence Locker Jr. Math performance and its relationship to math anxiety and metacognition. *North American Journal of Psychology*, 11(3), 2009.

Chin-Teng Lin, Shi-An Chen, Tien-Ting Chiu, Hong-Zhang Lin, and Li-Wei Ko. Spatial and temporal eeg dynamics of dual-task driving performance. *Journal of neuroengineering and rehabilitation*, 8 (1):1–13, 2011.

Drew Linsley, Junkyung Kim, Vijay Veerabadran, Charles Windolf, and Thomas Serre. Learning long-range spatial dependencies with horizontal gated recurrent units. *Advances in neural information processing systems*, 31, 2018.

Wei Ji Ma and Benjamin Peters. A neural network walks into a lab: towards using deep nets as models for human behavior. *arXiv preprint arXiv:2005.02181*, 2020.

Johannes Mehrer, Courtney J Spoerer, Nikolaus Kriegeskorte, and Tim C Kietzmann. Individual differences among deep neural network models. *Nature communications*, 11(1):1–12, 2020.

Kevin W Mickey and James L McClelland. A neural network model of learning mathematical equivalence. In *Proceedings of the annual meeting of the cognitive science society*, volume 36, 2014.

Don A Moore and Elizabeth R Tenney. Time pressure, performance, and productivity. In *Looking back, moving forward: A review of group and team-based research*, volume 15, pp. 305–326. Emerald Group Publishing Limited, 2012.

Gali Noti, Effi Levi, Yoav Kolumbus, and Amit Daniely. Behavior-based machine-learning: A hybrid approach for predicting human decision making. *arXiv preprint arXiv:1611.10228*, 2016.

A Emin Orhan and Wei Ji Ma. Efficient probabilistic inference in generic neural networks trained with non-probabilistic feedback. *Nature communications*, 8(1):1–14, 2017.

Frank Pajares and M David Miller. Role of self-efficacy and self-concept beliefs in mathematical problem solving: A path analysis. *Journal of educational psychology*, 86(2):193, 1994.

Eunji Park and Byungjoo Lee. An intermittent click planning model. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Mads Lund Pedersen, Michael J Frank, and Guido Biele. The drift diffusion model as the choice rule in reinforcement learning. *Psychonomic bulletin & review*, 24(4):1234–1251, 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Joshua C Peterson, Joshua T Abbott, and Thomas L Griffiths. Evaluating (and improving) the correspondence between deep neural networks and human representations. *Cognitive science*, 42 (8):2648–2669, 2018.

Joshua C Peterson, David D Bourgin, Mayank Agrawal, Daniel Reichman, and Thomas L Griffiths. Using large-scale experiments and machine learning to discover theories of human decision-making. *Science*, 372(6547):1209–1214, 2021.

Alexander Peysakhovich and Jeffrey Naecker. Using methods from machine learning to evaluate behavioral models of choice under risk and ambiguity. *Journal of Economic Behavior & Organization*, 133:373–384, 2017.

Ori Plonsky, Ido Erev, Tamir Hazan, and Moshe Tennenholtz. Psychological forest: Predicting human behavior. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL `http://jmlr.org/papers/v22/20-1364.html`.

Roger Ratcliff and Gail McKoon. The diffusion decision model: theory and data for two-choice decision tasks. *Neural computation*, 20(4):873–922, 2008.

Samuel Ritter, David GT Barrett, Adam Santoro, and Matt M Botvinick. Cognitive psychology for deep neural networks: A shape bias case study. In *International conference on machine learning*, pp. 2940–2949. PMLR, 2017.

Pau Rodríguez, Miguel A Bautista, Jordi Gonzalez, and Sergio Escalera. Beyond one-hot encoding: Lower dimensional target embedding. *Image and Vision Computing*, 75:21–31, 2018.

Warrick Roseboom, Zafeirios Fountas, Kyriacos Nikiforou, David Bhowmik, Murray Shanahan, and Anil K Seth. Activity in perceptual classification networks as a basis for human subjective time perception. *Nature communications*, 10(1):1–9, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Pulkit Singh, Joshua C Peterson, Ruairidh M Battleday, and Thomas L Griffiths. End-to-end deep prototype and exemplar models for predicting human behavior. *arXiv preprint arXiv:2007.08723*, 2020.

SM Slobounov, K Fukada, R Simon, M Rearick, and W Ray. Neurophysiological and behavioral indices of time pressure effects on visuomotor task performance. *Cognitive Brain Research*, 9(3): 287–298, 2000.

Philip L Smith. Diffusion theory of decision making in continuous report. *Psychological Review*, 123(4):425, 2016.

H Francis Song, Guangyu R Yang, and Xiao-Jing Wang. Training excitatory-inhibitory recurrent neural networks for cognitive tasks: a simple and flexible framework. *PLoS computational biology*, 12(2):e1004792, 2016.

H Francis Song, Guangyu R Yang, and Xiao-Jing Wang. Reward-based training of recurrent neural networks for cognitive and value-based tasks. *Elife*, 6:e21492, 2017.

Courtney J Spoerer, Tim C Kietzmann, Johannes Mehrer, Ian Charest, and Nikolaus Kriegeskorte. Recurrent neural networks can explain flexible trading of speed and accuracy in biological vision. *PLoS computational biology*, 16(10):e1008215, 2020.

Mark Steyvers, Guy E Hawkins, Frini Karayanidis, and Scott D Brown. A large-scale analysis of task switching practice effects across the lifespan. *Proceedings of the National Academy of Sciences*, 116(36):17735–17740, 2019.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision, 2015. URL https://arxiv.org/abs/1512.00567.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Guillaume Viejo, Mehdi Khamassi, Andrea Brovelli, and Benoît Girard. Modeling choice and reaction time during arbitrary visuomotor learning through the coordination of adaptive working memory and reinforcement learning. *Frontiers in behavioral neuroscience*, 9:225, 2015.

Li K Wenliang and Aaron R Seitz. Deep neural networks for modeling visual perceptual learning. *Journal of Neuroscience*, 38(27):6028–6044, 2018.

Steve Whittaker, Vaiva Kalnikaite, Victoria Hollis, and Andrew Guydish. 'don't waste my time' use of time information improves focus. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 1729–1738, 2016.

Guangyu Robert Yang, Madhura R Joglekar, H Francis Song, William T Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, 22(2):297–306, 2019.

Wojciech Zaremba and Ilya Sutskever. Learning to execute. *arXiv preprint arXiv:1410.4615*, 2014.

Hasida Ben Zur and Shlomo J Breznitz. The effect of time pressure on risky choice behavior. *Acta Psychologica*, 47(2):89–104, 1981.

Nadina O Zweifel, Nicholas E Bush, Ian Abraham, Todd D Murphey, and Mitra JZ Hartmann. A dynamical model for generating synthetic data to quantify active tactile sensing behavior in the rat. *Proceedings of the National Academy of Sciences*, 118(27):e2011905118, 2021.

A Appendix

A.1 Dataset collection

A.1.1 Participants

We recruited 50 participants in total (age 21.44 $\pm$ 3.22 y (mean $\pm$ SD); 27 female) from our local institution to finish the math modular task (details in Fig. 4(a)). Participants were recruited by email groups in our local institution and came from a variety of majors including engineering, computer science, biology, and so on. Six participants took part in the preliminary study to explore potential configurations of study design, whose results were removed. Other 44 participants were randomly and uniformly divided into 4 groups in order to fully capture the potential effects of time pressure in cognition performance, as described before. Two participants withdrew from the study and three did not finish the study completely. We also removed another three participants' results whose study duration was longer than 3 hours. This was much longer than normal study duration of other

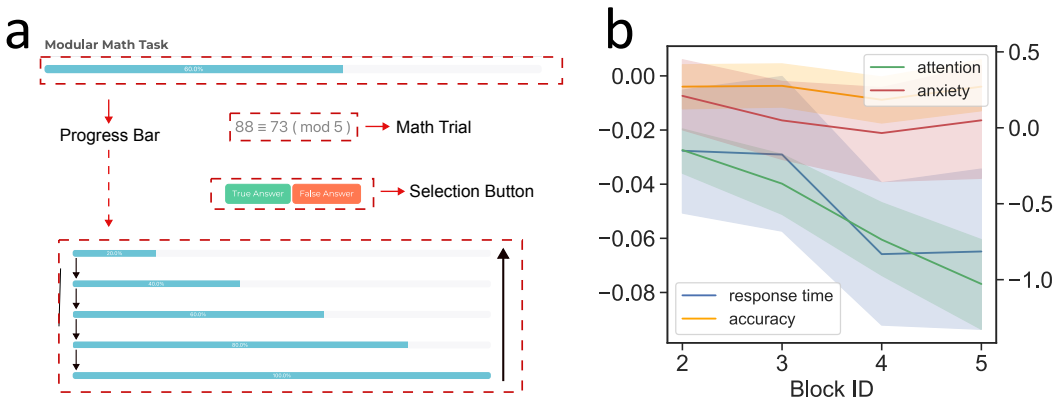

Figure 4: a: Math arithmetic task and time pressure feedback. Each math trial is composed of two two-digit numbers Num_1, Num_2 and one one-digit numer Num_3, formatted as: $Num_1 \equiv Num_2\,(\,mod\,Num_3)$. To solve this question, participants first use Num_1 to subtract Num_2 and judge whether the subtraction result could be divisible by Num_3. If it is divisible, they select "True" button. Otherwise, they select "False" button. When the time pressure feedback happens, a progress bar will be shown on top of the math question, which adds one unit for each second and reset and add again when it accumulates five units. b: Overall trend of relative change of response time/accuracy (left y axis), and attention/anxiety (right y axis), respectively, across 4 blocks.

participants (within 1 hour) and suggested that participants neither focused on the task nor took this experiment seriously. Finally, we had 36 participants: *None* group (10), *Static* group (9), *Random* group (7), *Rule* group (10). This study has been approved by the Institutional Review Board (IRB) in our local institution. We have obtained informed consent from all participants before study.

A.1.2 Procedure

All participants took part in a two-day study. For each day, they were asked to first finish an exercise session containing 20 math trials and then finish a formal session containing 300 math trials. The exercise session aimed to familiarize the users with tasks and measure users' baseline performance (without time pressure). In the formal session, different time pressure mechanisms were provided for different groups as mentioned above. Additionally, participants were requested to rate their current attention/anxiety status on a 7-point Likert scale every 30 trials. There was also a 5-min rest between exercise session and formal session. It took each participant an average of one hour for the study per day. In the study, participants were told to always take accuracy as the priority and then try their best to answer questions as soon as possible. The compensation rule for each participant (ranging from \$10 to \$100) also prioritized average accuracy over response time in order to encourage participants to follow our instructions. We finally obtained a large data set of 21,157 logical responses after removing invalid user response.

A.1.3 Math question generation and distribution

All math questions are composed of two two-digit numbers (Num_1, Num_2) and one one-digit number (Num_3). We denote the three numbers as $Num_1 = ab$, $Num_2 = cd$, $Num_3 = e$, respectively. So each math question could be denoted as $ab \equiv cd\,(\,mod\,e)$, where $a \in [1, 10)$, $b \in [2, 10)$, $c \in [1, 10)$, $d \in [1, b)$, $e \in [3, 10)$. All math questions are randomly generated for each trial. We have traversed all possible combinations of math digits in the math question format, which are distributed uniformly in the whole math space for the four groups. Participants' accuracy and the provided time pressure feedback are also distributed uniformly.

A.1.4 Groups

Here we describe details of four groups in dataset collection. *None* Group: Participants experienced no time pressure for any trial. *Static* Group: Time pressure was consistently applied for each trial.

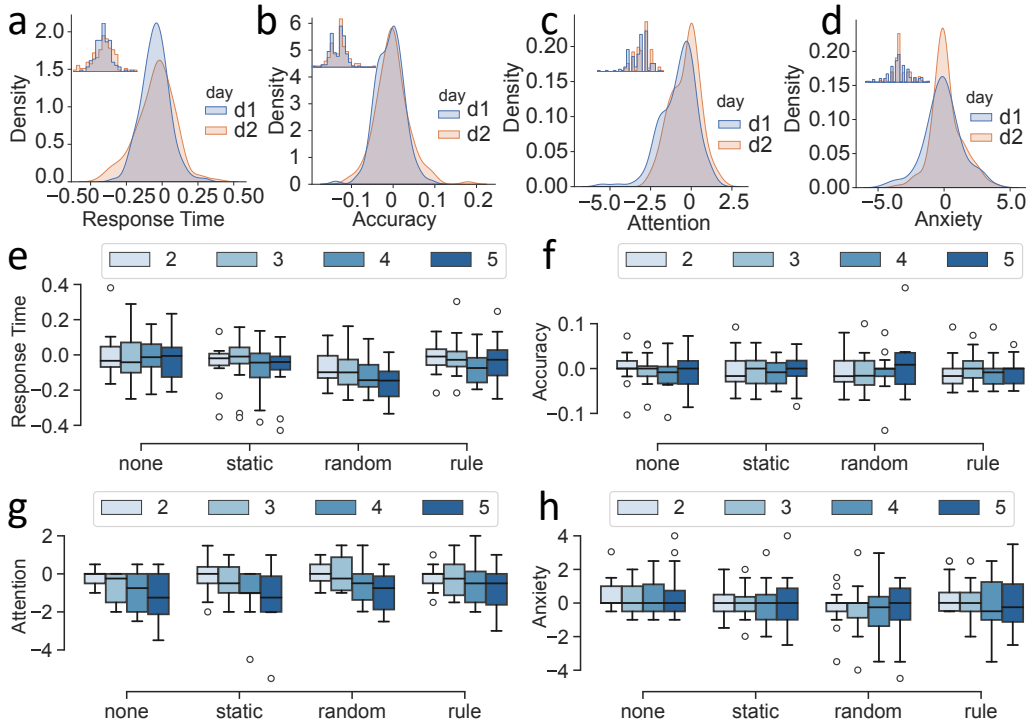

Figure 5: a,b,c,d: overall distribution of relative change of response time (a), accuracy (b), attention (c), and anxiety (d), respectively, across 2 days. e,f,g,h: box plot of relative change of response time (e), accuracy (f), attention (g), and anxiety (h), respectively, across 4 groups and 4 blocks.

Random Group: There was a 50% probability of time pressure being applied for each trial. *Rule* Group: Time pressure was adaptively applied based on users' past performance using a rule-based strategy. More details about such strategy are depicted below.

Rule-based strategy is designed to provide adaptive time pressure feedback for each trial according to participants' past performance in the *Rule* group. There is a response buffer to update and save user response of most recent 20 trials. For each new user response, it is updated in the response buffer. Then we calculate five metrics (mean response time, delta response time, mean accuracy, push counter, and tolerant counter) in the buffer to decide whether the time pressure feedback is delivered to participants in the next trial. The time pressure feedback only happens if: (a). Mean response time exceeds its threshold RT. Here we use the average response time in exercise session of each specific participant to be RT. (b). Delta response time exceeds its threshold deltaRT = 1 second. (c). Mean accuracy is lower than its threshold accuracy TA. Here we use the average accuracy in exercise session of each specific participant to be TA. (d). Push counter is lower than its threshold PC = 3. (e). Tolerant counter achieves its threshold TC = 2. When the time pressure feedback is decided to be delivered to the participant in the next trial, push counter adds 1 unit and tolerant counter is reset to 0.

These five metrics aim to ensure that time pressure feedback does not increase user response time but could increase user accuracy. Push counter and tolerant counter are designed to avoid introducing too much distraction to users. The strategy tolerates for a few trials and does not deliver time pressure feedback even if the first three metrics achieve the threshold. After the tolerant counter achieves the TC threshold, it delivers time pressure feedback. In addition, if the strategy delivers time pressure for too many times (exceeding PC threshold), the time pressure feedback is still not delivered to users. Therefore, rule-based strategy is a relatively conservative strategy which cares more about avoiding introducing additional distraction to users.

A.2 Dataset Exploration

To investigate the impact of different time pressure stimuli on cognition performance, we conducted an initial exploratory analysis on the dataset. To mitigate the influence of chance factors, we divided the 300 trials of the formal session into five blocks of equal size and calculated the block-wise averages for accuracy, response time, attention, and anxiety scores. Recognizing the inherent variability in users' baseline performance, we aimed to elucidate the impact of time pressure across different groups by comparing the *relative change* in user performance and status across the four groups. Specifically, let R_i denote the average result of $Block_i$, where R_1 ($Block_1$) represents the baseline performance. The final relative result for $Block_i$ ($i > 1$) is $(R_i - R_1)/R_1$ for accuracy and response time change and $R_i - R_1$ for attention and anxiety change. This adjustment accounts for the fact that attention/anxiety scores linearly reflect user status, while response time/accuracy changes need to be normalized against participants' individual baseline performances. The obtained results were then analyzed using repeated-measures ANOVA. To discern specific differences, Bonferroni-corrected paired post hoc t-tests were employed for pairwise comparisons between the groups, enabling a thorough exploration of the impact of different time pressure stimuli on cognition performance and user status.

A.2.1 Response time

In the analysis of between-subjects effects, the ANOVA revealed a significant effect of Group ($F_{3,32} = 3.015, P = 0.044 < 0.05$) (Fig. 5(e)). Specifically, a significant difference was identified between the *none* group (mean $\pm$ SD: -0.012 ± 0.021) and the *random* group (-0.105 ± 0.025) with $p = 0.039 < 0.05$. The *rule* group showed a larger reduction in response time (-0.034 ± 0.021) compared to the *none* group but a smaller reduction compared to the *static* group (-0.054 ± 0.022). Notably, the *random* group exhibited the most substantial reduction in response time. These results suggest that different types of time pressure stimuli may exert varying effects on response time.

Regarding within-subjects tests, a significant effect was observed across blocks ($F_{3,96} = 7.121, P < 0.001$) (Fig. 5(e)), specifically between the following blocks: $Block_2$ (-0.031 ± 0.011) vs. $Block_4$ (-0.070 ± 0.014): $p = 0.023 < 0.05$, $Block_2$ vs. $Block_5$ (-0.072 ± 0.014): $p = 0.026 < 0.05$, $Block_3$ (-0.033 ± 0.013) vs. $Block_4$: $p = 0.008 < 0.01$, $Block_3$ vs. $Block_5$: $p = 0.025 < 0.05$.

No interaction was found between Block and Group ($F_{9,96} = 0.958, P = 0.48$). Furthermore, there was no significant effect of Date ($F_{1,32} = 0.003, P = 0.959$) (Fig. 5(a)), and no other significant interaction effects were identified (all $P > 0.05$). These findings provide valuable insights into the differential impact of time pressure stimuli on response time and underscore the significance of within-subject variations across different blocks.

A.2.2 Accuracy

No significant effect was observed in Group ($F_{3,32} = 0.081, P = 0.97 > 0.05$), Block ($F_{3,30} = 0.313, P = 0.816 > 0.05$) (Fig. 5(f)), or Date ($F_{1,32} = 0.861, P = 0.36 > 0.05$) (Fig. 5(b)). Additionally, no other significant interaction effects were identified (all $P > 0.05$). This outcome aligns with expectations, as participants were instructed to prioritize accuracy over response time consistently. Consequently, the accuracy of users' choices should generally be high, while response time may vary depending on the stimuli. The lack of significant effects in these factors supports the study design and participants' adherence to the specified priority in their decision-making process.

The above results suggest that both time pressure stimuli and block number (not experiment date) may impact users' response time. This evidence contributes valuable insights and aligns with prior theory (Slobounov et al. (2000); Alexander et al. (2003)), providing a foundation to inform the design of our cognition model. The observed effects underscore the relevance of considering both math task and question ID in modeling and understanding the dynamics of user response time under varying conditions.

A.3 Math Logical Reasoning Agent

Existing work revealed humans' varied performance on different cognitive tasks of diverse difficulty levels (Hanich et al. (2001)). Therefore, it is essential to first encode features such as difficulty

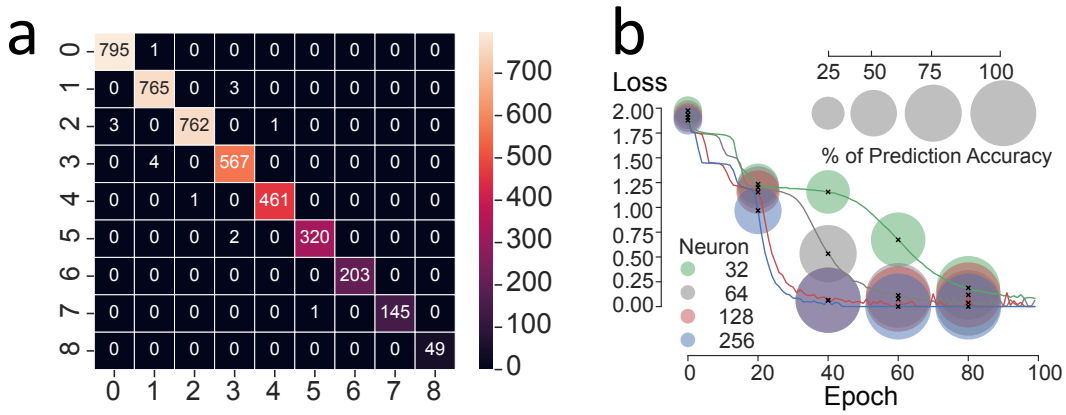

Figure 6: a. Confusion matrix (x axis: ground truth, y axis: prediction) for testing set prediction of the logical reasoning agent (LSTM neuron = 256). b. Training loss and accuracy with training epochs across four kinds of LSTM neurons of the logical reasoning agent.

levels of cognitive tasks so that we could model participants' varied responses to different math questions stem from features inherent in the questions. These features may influence user choice and response time even in ideal conditions (i.e., without external stimuli). To capture such features, we train a logical reasoning agent capable of solving math questions in a manner similar to humans. Subsequently, feature representations are extracted from the intermediate output of this logical reasoning agent.

Illustrated in Fig. 1 and Fig. 8, we employ an LSTM-based logical reasoning agent that takes a math question as input and outputs the corresponding answer. For example, given the sequence "$61 \equiv 26(\mathrm{mod}\ 4)$" as input, the agent outputs "3" (the remainder of the subtraction result, "35," of "61" and "26," divided by "4"). It is essential to note the distinction from the data collection process, where users are required to choose whether the subtraction result ("35") of "61" and "26" is divisible by "4"–a binary selection task.

In other words, the logical reasoning agent is trained to answer math arithmetic tasks correctly, rather than to predict user responses. This design choice ensures that the agent learns the potential arithmetic reasoning process and generates representative features of math questions, rather than performing a binary classification task.

The logical reasoning agent is a sequence-to-sequence model based on an LSTM model. Before inputting the math question into the LSTM, the math question is encoded into sequence vectors from original string format. Each math question is denoted as $ab \equiv cd\,(\,mod\,e)$, comprising 11 characters. We use one-hot encoding to deal with the characters. Specifically, each character is mapped into a 1×17 vector, where the location of this character in a pre-built character dictionary (["0","1","2","3","4","5","6","7","8","9","≡","(","m","o","d",")"," "]) is denoted as 1, and other locations are denoted as 0. So we finally obtain the 11×17 vector for each math question.

For each math question string (1×11), we use sequence encoding mentioned above to encode it into a sequence vector (11×17), which is then fed into the LSTM model. The hidden unit is 256 neurons, which is then connected with 17 neurons with softmax activation function. Finally, the neuron with the highest probability is the final output answer. We use Keras(Chollet et al. (2015)) to implement the model (loss function: categorical cross entropy, optimizer: Adam, learning rate: 0.001).

The logical reasoning agent aims to solve math tasks correctly. In short, given one math question as input, it directly outputs the arithmetic reasoning answer. Therefore, the training and testing of logical reasoning agent have no correlation or connection with real users' response. Hence, we prepare a separate dataset that is independent with users' dataset to train the logical reasoning agent. Finally, we have traversed all possible combinations of three numbers in math questions and gotten a dataset including 20414 samples, which is split into training set (80%) and testing set (20%).

Table 2: Performance of user choice classification of SVC models and response time estimation of SVR models across three math question representations: *Feature* label: SVM (both SVC and SVR) takes features extracted from logical reasoning agent as input, *String* label: SVM (both SVC and SVR) takes encoded vectors of raw math numbers as input, *Digits* label: SVM (both SVC and SVR) takes raw numeric math numbers as input.

	Choice Classification				Response Time Regression (MAPE)			
Input	Accuracy	F1-Score	Precision	Recall	Mean	STD	Lower	Upper
Digits	0.8107	0.0000	0.0000	0.0000	0.3740	0.3772	0.0121	1.4185
String	0.8174	0.0724	0.9333	0.0377	0.3813	0.3847	0.0135	1.4891
Feature	**0.9613**	**0.8996**	**0.8833**	**0.9166**	**0.3652**	**0.3648**	**0.0108**	**1.3612**

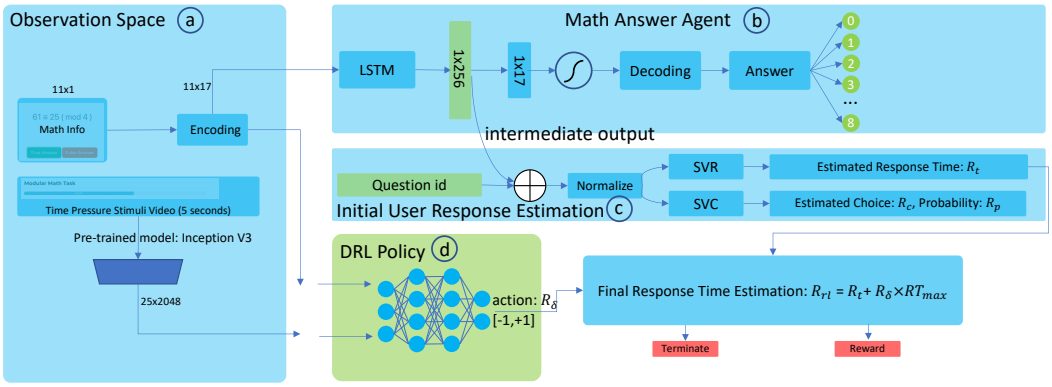

Figure 7: The detailed architecture of the pure DRL agent without drift-diffusion model.

A.4 SVM MODEL CONFIGURATION

As previously mentioned, the second step in our simulation framework (Fig. 1) involves transferring features captured by the logical agent to real responses of humans by utilizing SVM models to predict users' baseline performance without time pressure. The features comprise the intermediate output of the LSTM layer, with the output neuron number set to 256, resulting in 256 features captured by the math answer agent. During cognition performance analysis, we observed that users' performance is influenced by the block number. Therefore, for each trial, we introduce the question id as an additional input feature, concatenated with the previous 256 features for SVM models. The question id denotes the corresponding trial number in the dataset, resulting in a total of 257 features for predicting user response for each sample/trial. Users' response encompasses both user choice and response time. Consequently, the SVM models consist of a binary SVM classifier (SVC) to predict user choice (True or False selection) and an SVM regressor (SVR) to estimate users' response time.

The SVM model is implemented with scikit-learn(Pedregosa et al. (2011)). We use default regularization parameter, kernel, and other parameters for both SVM classifier (SVC) and regressor (SVR). The SVR takes 256 features from LSTM layer of math logical reasoning agent as well as question id for input and predicts user response time. The SVC not only predicts user response (choice) but also the probability R_p for each possible response, which serves as the boundary threshold in the drift-diffusion model.

A.5 PURE DEEP REINFORCEMENT LEARNING (DRL) AGENT

The pure DRL model is implemented with PyTorch(Paszke et al. (2019)), Stable Baselines3(Raffin et al. (2021)), and Gym(Brockman et al. (2016)).

Most parts of the pure DRL agent is the same as the hybrid DRL agent. The main difference lies in the way to represent effect of time pressure in human cognition performance. The hybrid DRL agent segments cognition process of each trial into frames and each action represents specific effect on each

frame/step. However, for the pure DRL agent, it directly takes the whole visual stimuli as input and output one action which represents the whole response time change due to time pressure. The final estimation of regulated response time is the sum of this action and basic response time estimated by SVR models.

A.5.1 DRL TRAINING LOOP

The DRL training loop is similar with the hybrid DRL agent, which is still composed of observation space, action space, reward, terminal state, and learning policy. More details are depicted below.

A.5.2 OBSERVATION SPACE

The observation space still consists of two parts: math question information and dynamic time pressure visual stimuli. For each math trial, the math question encoding is the same as the hybrid DRL agent. For time pressurem, different from the hybrid DRL agent, the pure DRL agent does not segment visual stimuli into frames. Instead, it takes whole time pressure stimuli video (lasting 5 seconds) as input. We first use a pre-trained Inception-V3 model(Szegedy et al. (2015)) in Keras(Chollet et al. (2015)) to extract features from this video. The dimension of output features from each frame of the video is 1×2048. For the whole video, we use the same frame rate as the hybrid DRL agent ($f = 5$). So finally we have $5\ seconds \times 5 = 25$ frames. The final feature dimension of this time pressure visual stimuli in observation space of the pure DRL agent is 25×2048.

A.5.3 ACTION SPACE

The action space contains one action (R_δ) with continuous numeric value which is normalized into the range from -1 to 1. Different from the hybrid DRL where each step is one frame of user cognition process, here each step of the pure DRL agent is just one trial of users' response. For each trial, user baseline performance is obtained from SVM models. The action of the pure DRL agent represents perturbation for baseline response time (R_t) because of time pressure stimuli. Therefore, the final estimation of user response time is $R_{rl} = R_t + R_\delta \times RT_{max}$, where $RT_{max} = 10$ is the maximum of user response time in the dataset.

A.5.4 TERMINAL STATE

The terminal state happens when final estimated response time R_{rl} exceeds normal range (smaller than 0 or larger than $RT_{max} = 10$) or the pure DRL agent achieves maximum steps in one episode. Here, one step represents one math trial in the dataset. Here we set the maximum step number to be 60 steps, which is the same as the trial number of each block in our user study result analysis.

A.5.5 REWARD

Different from the hybrid DRL agent that could only obtain reward in terminate state, for the pure DRL agent, it gets reward during each step (each trial in user dataset). The reward mainly aims to encourage the pure DRL agent to simulate effect of time pressure visual stimuli that is similar with real users' response. Therefore, the reward function is:

$$r_i = \begin{cases} |E_{rl} - E_{svm}|/E_{svm} + P^*, & E_{rl} < E_{svm} \\ 0, & E_{rl} \geq E_{svm} \end{cases} \tag{1}$$

where E_{rl} and E_{svm} are the estimated error rate of the pure DRL's predicting response time (R_{rl}) and the SVM's predicting response time ($R_{svm} = R_t$) compared with real response time (R_u) respectively, i.e. $E_{rl} = |R_{rl} - R_u|/R_u$, $E_{svm} = |R_{svm} - R_u|/R_u$. P^* is the penalty caused by terminal state if the pure DRL agent's estimated response time exceeds the normal range (0 to 10 seconds) ($P^* = -1$). Otherwise, $P^* = 0$.

A.5.6 LEARNING ALGORITHM AND POLICY

We use Proximal Policy Optimization (PPO)(Schulman et al. (2017)) as the learning algorithm and multilayer perceptron (MLP) to be the policy for agent training. All hyperparameters and network architectures follow default settings in Stable Baselines3(Raffin et al. (2021)).

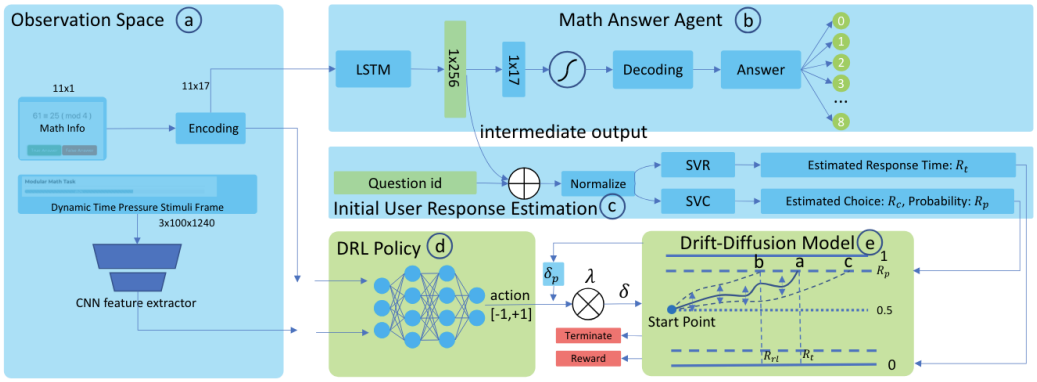

Figure 8: The detailed architecture of our ReactiveAgent framework. First, we use math questions to train a math answer agent to solve them without considering users' response. Second, for each math question, we transfer features extracted from LSTM layer in math answer agent without time pressure to make predictions of user choice and response time using SVM (initial estimation). The initial estimated response time and predicted choice probability will generate evidence accumulation trajectory in the drift-diffusion model. Third, the DRL agent will take math question and each frame of dynamic time pressure stimuli as input and take specific action to modulate evidence accumulation process. When evidence accumulator achieves boundary threshold, the final prediction of response time is generated and DRL agent achieves terminate state.

A.6 Hybrid DRL Agent with Drift-Diffusion Model

A.6.1 Drift-diffusion model (DDM)

The DDM assumes that users make decision by accumulating evidence for each choice and make the final selection when the evidence accumulator achieves the threshold. Our framework incorporates the SVM model's predicting results into the DDM. Specifically, we use the output probability of SVC as the accumulated evidence, whose start point is 0.5. The boundary threshold is R_p, which is the probability when SVC makes the predictions. Different from traditional DDM that uses Bayesian modelling to draw a distribution of user response time, we need to have a fine-grained trajectory from start point to end point for each math trial to support our reinforcement learning process. Here we use Sigmoid function(Han & Moraga (1995)) to represent the trajectory from the start point to the end point. When users are solving math questions, they are usually more confident given more time to answer (Legg & Locker Jr (2009); Pajares & Miller (1994)). Therefore, we could use a monotonous function to represent the trajectory T, i.e. the Sigmoid function. Moreover, we use Brownian motion (Smith (2016)) to add noise into the Sigmoid curve in order to introduce the randomness in decision making trajectory (Smith (2016)). Note that the final simulated trajectory is not always monotonous because such trajectory is modulated and modified by the DRL agent adaptively according to the environmental stimuli.

A.6.2 DRL training loop

The DRL training loop is composed of observation space, action space, reward, terminal state, and learning policy. The observation space serves as the model entrance to accept math question information and external stimuli as input. The action space contains a set of potential actions that the DRL agent could take to perform simulation. The reward is used to guide the DRL agent to update its strategy powered by the learning policy to take the optimal action so as to achieve highest possible reward. Terminal state represents the end of one training episode.

A.6.3 Observation space

The observation space consists of two parts: math question information and dynamic time pressure visual stimuli. For each math trial, the math question is encoded as a sequence vector (11×17) just like the logical reasoning agent. The dynamic time pressure visual stimuli is segmented into visual

frames just like what users perceive in the study. Given frame rate f, for each frame i, we can obtain the specific image S_i of the visual stimuli for input in the observation space (we set $f = 5$). In order to encode the frame for input, we use a default CNN feature extractor in Stable Baselines3 (Raffin et al. (2021)) to extract features automatically from the time pressure image.

A.6.4 ACTION SPACE

The action space contains one action with continuous numeric value from -1 to 1. The hybrid DRL agent takes one step for each frame i. When the output action a is 0, it means that the current time pressure frame has no effect on evidence accumulator in drift-diffusion model. When the output action a is from -1 to 0 or 0 to +1, then it means current time pressure frame leads to negative or positive change δ on evidence accumulator. The change δ is obtained from the trajectory of drift diffusion model. Given boundary threshold R_p, start point S_p, response time R_t and frame rate f, the change δ of evidence accumulation in each frame is $\delta = \lambda \times \delta_p$, $\delta_p = |R_p - S_p|/(f \times R_t)$, where λ is the discounting factor to avoid the DRL agent introducing too aggressive bias.

A.6.5 TERMINAL STATE

Terminal state happens when the evidence accumulator achieves boundary threshold (R_t) or the hybrid DRL agent achieves maximum steps in one episode. Here, one episode represents one math trial in the dataset. Here we set the maximum response time to 10 seconds, consistent with the largest response time in our dataset. So the maximum step number $N = RT_{max} \times f = 10 \times 5 = 50$ steps. If the DRL agent takes S_n steps when the evidence accumulator achieves R_t, then the new predicted response time is $R_{rl} = S_n/f$.

A.6.6 REWARD

For each step during per episode, the hybrid DRL agent only gets reward in the terminal state. For other situations, the reward is 0. The reward mainly aims to encourage the hybrid DRL agent to behave similarly with real users. Therefore, the reward function is:

$$r_i = \begin{cases} |E_{rl} - E_{svm}|/E_{svm} + P^*, & E_{rl} < E_{svm} \\ 0, & E_{rl} \geq E_{svm} \end{cases} \tag{2}$$

where E_{rl} and E_{svm} are the estimated error rate of the hybrid DRL's predicting response time (R_{rl}) and the SVM's predicting response time ($R_{svm} = R_t$) compared with real response time (R_u) respectively, i.e. $E_{rl} = |R_{rl} - R_u|/R_u$, $E_{svm} = |R_{svm} - R_u|/R_u$. P^* is the penalty caused by terminal state if the hybrid DRL agent's step number exceeds the maximum step threshold ($P^* = -1$). Otherwise, $P^* = 0$.

A.6.7 LEARNING ALGORITHM AND POLICY

We use Proximal Policy Optimization (PPO) (Schulman et al. (2017)) as the learning algorithm and multilayer perceptron (MLP) to be the policy for agent training. All hyperparameters and network architectures follow the default settings in Stable Baselines3(Raffin et al. (2021)). The hybrid DRL model is implemented with PyTorch(Paszke et al. (2019)), Stable Baselines3(Raffin et al. (2021)), and Gym(Brockman et al. (2016)).

A.7 BASELINE MODELS

Our baseline models are adapted into our problem corresponding to the recent State-of-the-Art (SOTA) computational models in human decision making (Bourgin et al. (2019)) and response time prediction (Goetschalckx et al. (2024); Jaffe et al. (2023)).

The whole dataset is first split into raw training (80%) and test set (20%). The raw training set is then split into model training set (80%) and validation set (20%). The validation set is used to select the best epoch.

All neural network-based models use MAPE loss function, Adam optimizer (Kingma & Ba (2014)) with learning rate of 0.001 and batch size of 16. All models are trained on 2 Nvidia RTX A6000

GPUs (48GB GPU memory). All neural network models are implemented by PyTorch (Paszke et al. (2019)) and other machine learning models are implemented by scikit-learn (Pedregosa et al. (2011)).

- Baseline Type 1: Model Input Format: Task: Video, Feedback: Video, Question ID: Numeric Value.
  - hGRU (Linsley et al. (2018)): This model comes from (Goetschalckx et al. (2024)) to simulate human response time in visual tasks. We use (Goetschalckx et al. (2024); Linsley et al. (2018)) to implement this model. The original hGRU model accepts image as input. We adjust the dimensions to accept video (including both task and time pressure visual feedback) as input. This model is trained from beginning without pre-trained models. The output of hGRU model is then concatenated with question ID for input into a linear layer (64 neurons) to predict response time. Each epoch takes about 40 minutes for training. We report the results for the best epoch out of 30 (based on performance on the validation set).
  - LSTM + AlexNet: This model is based on (Jaffe et al. (2023)) that uses LSTM to simulate human response time in cognitive tasks. Here we use the same LSTM configurations as (Jaffe et al. (2023)). To adapt it to accept video as input, we first use pre-trained AlexNet (Krizhevsky et al. (2012)) from TorchVision (Paszke et al. (2019)) to extract features from each frame of the video. The sequence of features from all frames are then input into LSTM layer. The output of the LSTM layer is then concatenated with question ID for input into a linear layer (64 neurons) to predict response time. Each epoch takes about 40 minutes for training. We report the results for the best epoch out of 30 (based on performance on the validation set).
  - LSTM + VGG-16: This model is similar with LSTM + AlexNet but we replace the AlexNet with pre-trained VGG-16 (Simonyan & Zisserman (2014)) in TorchVision (Paszke et al. (2019)) to extract visual features from video frames. Each epoch takes about 40 minutes for training. We report the results for the best epoch out of 30 (based on performance on the validation set).
  - LSTM + ViT-B-16: This model is similar with LSTM + AlexNet but we replace the AlexNet with pre-trained ViT-B-16 (Dosovitskiy et al. (2020)) in TorchVision (Paszke et al. (2019)) to extract visual features from video frames. Each epoch takes about 60 minutes for training. We report the results for the best epoch out of 30 (based on performance on the validation set).
  - MLP + 3D ResNet: This model is based on (Bourgin et al. (2019)) that uses MLP to predict human decision making. We follow the same MLP architecture as (Bourgin et al. (2019)). To adapt it to accept video input, we first use pre-trained 3D ResNet (Tran et al. (2018)) in TorchVision (Paszke et al. (2019)) to extract features from the video directly (instead of each video frame). The extracted features are then concatenated with question ID for input into the MLP model. Each epoch takes about 25 minutes for training. We report the results for the best epoch out of 30 (based on performance on the validation set).
- Baseline Type 2: Model Input Format: Task: Encoded String, Feedback: Video, Question ID: Numeric Value
  - LSTM-V1 + 3D ResNet: This model is based on (Jaffe et al. (2023)) that uses LSTM to simulate human response time in cognitive tasks. Here we use the same LSTM configurations as (Jaffe et al. (2023)). To adapt it to accept video input, we first use pre-trained 3D ResNet (Tran et al. (2018)) in TorchVision (Paszke et al. (2019)) to extract features from the video directly (instead of each video frame). The extracted feedback video features are then concatenated with both math task string with one-hot encoding and question ID for input into the LSTM model. The output of the LSTM layer is then passed into a linear layer (64 neurons) to predict response time. Each epoch takes about 12 minutes for training. We report the results for the best epoch out of 30 (based on performance on the validation set).
  - LSTM-V2 + 3D ResNet: This model is similar with LSTM-V1 + 3D ResNet. The difference is that the extracted feedback video features are first fed into the LSTM layer and then the output is concatenated with both math task string with one-hot encoding and question ID to predict response time. Each epoch takes about 12 minutes for

training. We report the results for the best epoch out of 30 (based on performance on the validation set).

  - MLP + 3D ResNet: This model is based on (Bourgin et al. (2019)) that uses MLP to predict human decision making. We follow the same MLP architecture as (Bourgin et al. (2019)). To adapt it to accept video input, we first use pre-trained 3D ResNet (Tran et al. (2018)) in TorchVision (Paszke et al. (2019)) to extract features from the video directly (instead of each video frame). The extracted features are then concatenated with both math task string with one-hot encoding and question ID for input into the MLP model. Each epoch takes about 15 minutes for training. We report the results for the best epoch out of 30 (based on performance on the validation set).
  - Transformer + 3D ResNet: This model is similar with MLP + 3D ResNet. The difference is that we replace the MLP model with the transformer model. We follow the default architecture of transformer in (Vaswani et al. (2017)). Each epoch takes about 12 minutes for training. We report the results for the best epoch out of 30 (based on performance on the validation set).

- Baseline Type 3: Model Input Format: Task: Numeric Value, Feedback: Video, Question ID: Numeric Value.
  - LSTM-V1 + 3D ResNet: This model is based on (Jaffe et al. (2023)) that uses LSTM to simulate human response time in cognitive tasks. Here we use the same LSTM configurations as (Jaffe et al. (2023)). To adapt it to accept video input, we first use pre-trained 3D ResNet (Tran et al. (2018)) in TorchVision (Paszke et al. (2019)) to extract features from the video directly (instead of each video frame). The extracted feedback video features are then concatenated with both math task digits and question ID for input into the LSTM model. The output of the LSTM layer is then passed into a linear layer (64 neurons) to predict response time. Each epoch takes about 12 minutes for training. We report the results for the best epoch out of 30 (based on performance on the validation set).
  - LSTM-V2 + 3D ResNet: This model is similar with LSTM-V1 + 3D ResNet. The difference is that the extracted feedback video features are first fed into the LSTM layer and then the output is concatenated with both math task digits and question ID to predict response time. Each epoch takes about 12 minutes for training. We report the results for the best epoch out of 30 (based on performance on the validation set).
  - MLP + 3D ResNet: This model is based on (Bourgin et al. (2019)) that uses MLP to predict human decision making. We follow the same MLP architecture as (Bourgin et al. (2019)). To adapt it to accept video input, we first use pre-trained 3D ResNet (Tran et al. (2018)) in TorchVision (Paszke et al. (2019)) to extract features from the video directly (instead of each video frame). The extracted features are then concatenated with both math task digits and question ID for input into the MLP model. Each epoch takes about 15 minutes for training. We report the results for the best epoch out of 30 (based on performance on the validation set).
  - Transformer + 3D ResNet: This model is similar with MLP + 3D ResNet. The difference is that we replace the MLP model with the transformer model. We follow the default architecture of transformer in (Vaswani et al. (2017)). Each epoch takes about 12 minutes for training. We report the results for the best epoch out of 30 (based on performance on the validation set).

- Baseline Type 4: Model Input Format: Task: Numeric Value, Feedback: Numeric Value, Question ID: Numeric Value. For this baseline type, all input features (task, feedback, question ID) are directly concatenated into 1D array for input into models. The baseline models in this type are mainly based on (Bourgin et al. (2019)), which presents several machine learning models to predict human decision making with similar model input.
  - Decision Tree: We use scikit-learn (Pedregosa et al. (2011)) to implement this model and follow all default settings in scikit-learn. The training process takes within 10 minutes.
  - Linear Regression: We use scikit-learn (Pedregosa et al. (2011)) to implement this model and follow all default settings in scikit-learn. The training process takes within 10 minutes.

  - LSTM: This model is based on (Jaffe et al. (2023)) that uses LSTM to simulate human response time in cognitive tasks. Here we use the same LSTM configurations as (Jaffe et al. (2023)). Each epoch takes about 2 minutes for training. We report the results for the best epoch out of 100 (based on performance on the validation set).
  - MLP: This model is based on (Bourgin et al. (2019)) that uses MLP to predict human decision making. We follow the same MLP architecture as (Bourgin et al. (2019)). Each epoch takes about 2 minutes for training. We report the results for the best epoch out of 100 (based on performance on the validation set).
  - Random Forest: We use scikit-learn (Pedregosa et al. (2011)) to implement this model and follow all default settings in scikit-learn. The training process takes within 10 minutes.
  - SVM: We use scikit-learn (Pedregosa et al. (2011)) to implement this model and follow all default settings in scikit-learn. The training process takes within 10 minutes.
  - Transformer: We follow the default architecture of transformer in (Vaswani et al. (2017)). Each epoch takes about 2 minutes for training. We report the results for the best epoch out of 100 (based on performance on the validation set).
  - XGBoost: We use XGBoost (Chen & Guestrin (2016)) to implement this model and follow all default settings in XGBoost. The training process takes within 10 minutes.
- Baseline Type 5: Model Input Format: Task: Encoded String, Feedback: Numeric Value, Question ID: Numeric Value. For this baseline type, the math task questions come with textual string format and get encoded with one-hot encoding (Rodríguez et al. (2018)), which are then concatenated with feedback and question ID for input into models. The baseline models in this type are mainly based on (Bourgin et al. (2019)), which presents several machine learning models to predict human decision making with similar model input.
  - Decision Tree: We use scikit-learn (Pedregosa et al. (2011)) to implement this model and follow all default settings in scikit-learn. The training process takes within 10 minutes.
  - Linear Regression: We use scikit-learn (Pedregosa et al. (2011)) to implement this model and follow all default settings in scikit-learn. The training process takes within 10 minutes.
  - LSTM: This model is based on (Jaffe et al. (2023)) that uses LSTM to simulate human response time in cognitive tasks. Here we use the same LSTM configurations as (Jaffe et al. (2023)). Each epoch takes about 2 minutes for training. We report the results for the best epoch out of 100 (based on performance on the validation set).
  - MLP: This model is based on (Bourgin et al. (2019)) that uses MLP to predict human decision making. We follow the same MLP architecture as (Bourgin et al. (2019)). Each epoch takes about 2 minutes for training. We report the results for the best epoch out of 100 (based on performance on the validation set).
  - Random Forest: We use scikit-learn (Pedregosa et al. (2011)) to implement this model and follow all default settings in scikit-learn. The training process takes within 10 minutes.
  - SVM: We use scikit-learn (Pedregosa et al. (2011)) to implement this model and follow all default settings in scikit-learn. The training process takes within 10 minutes.
  - Transformer: We follow the default architecture of transformer in (Vaswani et al. (2017)). Each epoch takes about 2 minutes for training. We report the results for the best epoch out of 100 (based on performance on the validation set).
  - XGBoost: We use XGBoost (Chen & Guestrin (2016)) to implement this model and follow all default settings in XGBoost. The training process takes within 10 minutes.

A.8 Further Discussion

Modelling dynamics in human cognitive responses to external stimuli is fundamental to understand how the brain dynamically reacts to the environment. However, the prevailing trend in contemporary research (Jaffe et al. (2023); Peysakhovich & Naecker (2017); Lake et al. (2017); Ma & Peters (2020); Mehrer et al. (2020); Golan et al. (2020); Kumbhar et al. (2020); Battleday et al. (2017; 2020); Singh et al. (2020); Peterson et al. (2018); Battleday et al. (2021); Peterson et al. (2021); Noti

Table 3: Results for all baseline model performance on response time simulation. For MAPE, we show its mean value (Mean), standard deviation (STD), 2.5th (Lower) and 97.5th (Upper) percentiles of the MAPE distribution (95% confidence interval).

		MAPE			
Model Input Type	Model Type Name	Mean	STD	Lower	Upper
Task: Video Feedback: Video	hGRU	0.3335	0.2486	0.0153	0.9406
	LSTM + AlexNet	0.3344	0.2602	0.0132	0.9954
	LSTM + VGG-16	0.3355	0.2708	0.0128	1.0393
	LSTM + ViT-B-16	0.3339	0.2573	0.0145	0.9852
	MLP + 3D ResNet	0.3330	0.2507	0.0121	0.9390
Task: Encoded String Feedback: Video	LSTM-V1 + 3D ResNet	0.3334	0.261	0.0151	0.9866
	LSTM-V2 + 3D ResNet	0.3376	0.2169	0.0185	0.7618
	MLP + 3D ResNet	0.3331	0.2550	0.0125	0.9601
	Transformer + 3D ResNet	0.3306	0.2496	0.0145	0.9462
	ReactiveAgent	**0.2999**	**0.2318**	**0.0131**	**0.8029**
Task: Numeric Value Feedback: Video	LSTM-V1 + 3D ResNet	0.3341	0.2617	0.0152	0.9923
	LSTM-V2 + 3D ResNet	0.3286	0.2538	0.0147	0.9707
	MLP + 3D ResNet	0.3333	0.2579	0.0147	0.9731
	Transformer + 3D ResNet	0.3315	0.2526	0.0152	0.9615
Task: Numeric Value Feedback: Numeric Value	Decision Tree	0.3617	0.364	0.015	1.3729
	Linear Regression	0.3595	0.3608	0.0113	1.3399
	LSTM	0.3059	0.2434	0.0141	0.9253
	MLP	0.3293	0.2441	0.0151	0.9257
	Random Forest	0.3650	0.3684	0.0117	1.3448
	SVM	0.3299	0.3108	0.0113	1.1827
	Transformer	0.3052	0.2446	0.0112	0.9309
	XGBoost	0.3508	0.3469	0.0112	1.3075
Task: Encoded String Feedback: Numeric Value	Decision Tree	0.3639	0.3639	0.0112	1.3917
	Linear Regression	0.3512	0.3469	0.0105	1.3176
	LSTM	0.3278	0.2478	0.0142	0.9397
	MLP	0.3333	0.2577	0.0145	0.9724
	Random Forest	0.3600	0.3630	0.0130	1.3620
	SVM	0.3245	0.3101	0.0123	1.1952
	Transformer	0.3299	0.2481	0.0142	0.9350
	XGBoost	0.3406	0.3363	0.0111	1.2611

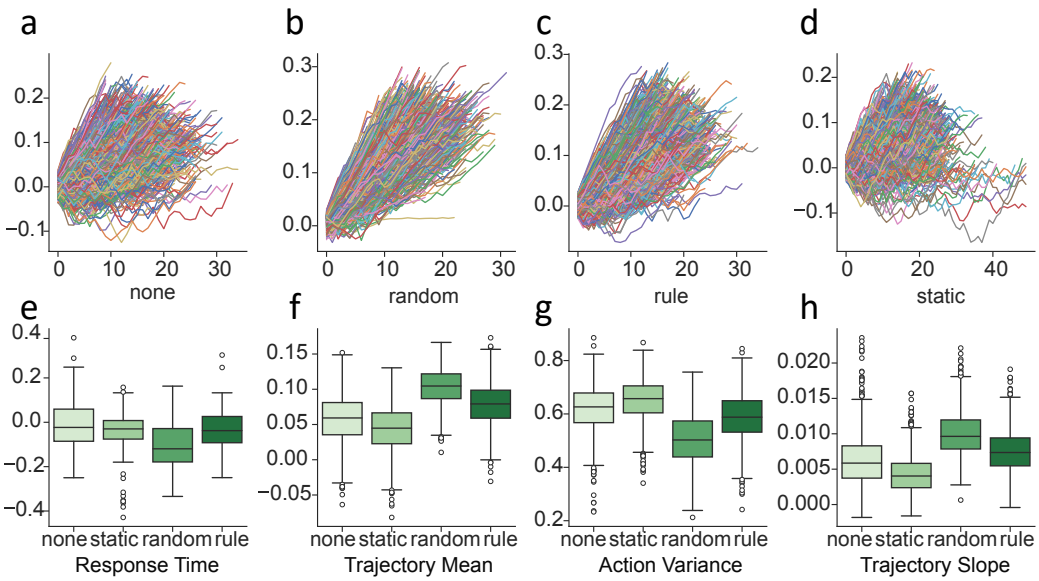

Figure 9: a,b,c,d: Time pressure effect trajectories of four groups, respectively. e: Box plot of relative response time change across four groups in the whole dataset. f,g,h: Box plot of mean value of time pressure effect trajectories (f), standard deviation of action trajectories (g), slope of time pressure effect trajectories (h) of four groups in predicted testing dataset by Hybrid DRL agent. The slope of one trajectory is calculated from the start point to the end point of the trajectory.

et al. (2016); Bourgin et al. (2019); Plonsky et al. (2017)) predominantly centers on the modeling of human cognition within standardized and idealized contexts, thereby often neglecting the nuanced influence exerted by external stimuli (Do et al. (2021); Park & Lee (2020)). Conversely, certain investigations adopt an oversimplified perspective by treating external stimuli as a persistent and unchanging factor throughout the cognitive processes (Bourgin et al. (2019)). A more sophisticated modeling methodology is deemed essential, particularly when addressing dynamic environmental stimuli that exhibit temporal fluctuations contingent upon user performance. This refined approach advocates for a nuanced consideration of stimuli variation at fine temporal scales, thereby perpetuating a continuous impact on human cognitive behaviors. Our hybrid modeling approach, characterized by the incorporation of Deep Reinforcement Learning (DRL) to emulate external stimuli within the explainable drift-diffusion model at a granular level, takes into account subject-specific and stimuli-specific behavioral distinctions. This distinctive feature sets our framework apart from antecedent studies, which predominantly concentrated on the coarse-grained posterior estimation of decision-making through reinforcement learning (Viejo et al. (2015); Pedersen et al. (2017)). The elucidative nature of our framework significantly augments our capacity to comprehend and interpret the intricate interplay between environmental stimuli and cognitive behaviors.

Note: This manuscript has included all authors who make contributions to this work. Our study has been approved by the Institutional Review Board (IRB) in our local institution. The roles and responsibilities were agreed among all authors. We have considered citations of existing research which is related to this work. We have removed all personal identifiers of participants in our dataset for responsible usage of our datasets and codes.