CogReact: A Reinforced Framework to Model Human Cognitive Reaction Modulated by Dynamic Intervention

Songlin Xu [1] **Xinyu Zhang** [1]

Abstract

Using deep neural networks as computational models to simulate cognitive processes can provide key insights into human behavioral dynamics. 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 ***CogReact***, which integrates drift-diffusion with deep reinforcement learning to simulate granular effects of dynamic environmental stimuli on the human cognitive process. Quantitatively, it improves cognition modeling by considering the temporal effect of environmental stimuli on the cognitive process and captures both subject-specific and stimuli-specific behavioral differences. Qualitatively, it captures general trends in the human cognitive process under stimuli. We examine our approach under diverse environmental influences across various cognitive tasks. Overall, it demonstrates a powerful, data-driven methodology to simulate, align with, and understand the vagaries of human cognitive response 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; 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). Alternatively, some studies treat environmental stimuli as a constant factor 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 user performance. This nuanced approach involves stimuli variation at fine-grained 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 to simulate the impact of dynamic environmental stimuli on the regulation of human cognitive behaviors with precision at a fine-grained level?**

We address this question starting 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. 5(a)), thereby instilling a sense of urgency. Secondly, the presence of time pressure may vary dynamically across different trials. Since time pressure stimuli represent a well-established feedback modality to modulate human cognitive performance (Cheng, 2017; Slobounov et al., 2000; Moore & Tenney, 2012; Edland & Svenson, 1993; Whittaker et al., 2016), modeling such a modulation effect holds the promise to offer valuable insights in understanding human cognition (Jaffe et al., 2023) and facilitating adaptive intervention design for regulating user performance (Costa et al., 2019).

In this paper, we introduce a systematic hybrid framework (***CogReact***) depicted in Fig. 1. This framework integrates

[1]Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093, USA. Correspondence to: Songlin Xu <soxu@ucsd.edu>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

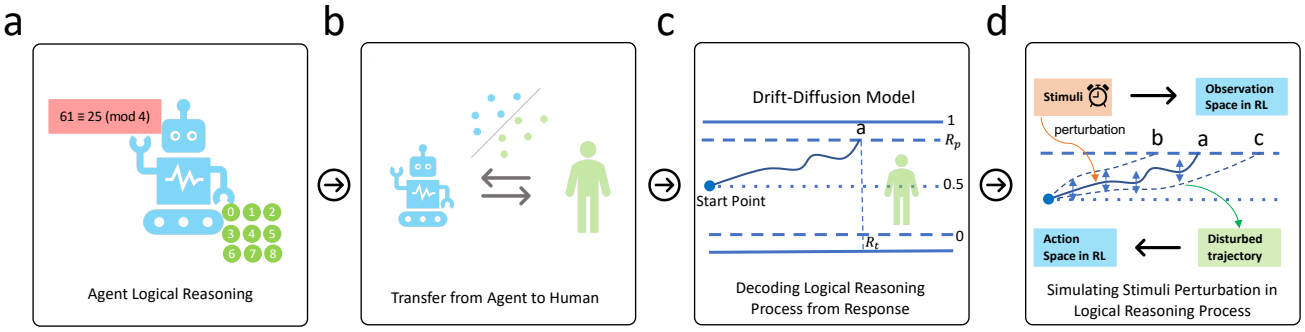

Figure 1. **Illustration of the overall framework**. First, we train a logical reasoning agent to solve cognitive tasks without considering user response. Second, we transfer features extracted from the logical reasoning agent without time pressure to real user choice and response time (initial estimation). Third, the initial estimated response time and predicted choice probability generate an 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 takes a specific action to modulate evidence accumulation process. When evidence accumulator achieves the boundary threshold, the final prediction of response time is generated and the DRL agent achieves a terminate state.

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 user 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 cognitive 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.

In addition, we further show the generalization ability of our framework by extending it into two additional public datasets in more diverse task (decision making, learning) and feedback modalities (numeric, textual). In summary, our contribution is three-folded:

- We propose ***CogReact***, a hybrid framework to incorporate classical cognition models (drift-diffusion model) with deep reinforcement learning to simulate perturbation of environmental stimuli on evidence accumulation process during human cognitive response.
- We comprehensively validate the effectiveness of our framework by comparing with a range of baselines and extensive ablation studies. Additionally, we are open-sourcing our code and a newly collected large dataset[1], comprising 21,157 logical reasoning responses, as a contribution to the research community.
- We demonstrate how our framework is adapted to three representative cognitive tasks: mathematical reasoning, decision making, and learning. This adaptation for both discrete user inputs and continuous behaviors establishes a foundation for extending the framework to accommodate diverse user cognitive responses.

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,

[1]https://github.com/cogreact/CogReact

2008) treats the cognitive process as an evidence accumulation process for humans to make decisions, thus simulating the speed-accuracy tradeoff (Heitz, 2014).

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), categorization (Battleday et al., 2017), decision making (Binz & Schulz, 2024; 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), etc.

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 balance between accuracy and response time observed in biological vision (Spoerer et al., 2020). Recently, (Goetschalckx et al., 2024) computed human-like reaction time from convolutional RNN using evidential deep learning (Sensoy et al., 2018). Furthermore, task-DyVA (Jaffe et al., 2023) modeled cognitive response time with RNN-based latent dynamical systems.

These existing models predominantly simulate response time under ideal conditions. In contrast, limited work has focused on modeling the impact of external stimuli perturbations, such as environmental stress, on task performance. We argue that a more nuanced modeling approach is essential, especially when addressing dynamic external stimuli that fluctuate over time based on user performance. This refined approach requires capturing stimuli variations at fine-grained timescales, which exert a continuous and evolving influence on human cognitive behaviors.

3. Model and Methodology

3.1. Math Reasoning Task and Dataset

We used a math arithmetic task with time pressure visual stimuli as our initial model exploration context. The illustration of the task and stimuli is depicted in Fig. 5 and in Appendix A.2. In each math trial, participants were presented with two two-digit numbers and tasked with determining whether their subtraction result was divisible by a given one-digit number. Participants made a binary decision for each trial, with varying settings of time pressure stimuli described below.

We collected an extensive dataset encompassing 21,157 valid responses (choice accuracy and response time) from 44 participants engaged in the task (see Fig. 6(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 user past performance using a rule-based strategy (more details of such strategy are provided in Appendix A.3.4). This collection has been approved by the Institutional Review Board (IRB) in our local institution. More details are in Appendix A.2.

Our dataset analysis in Appendix A.4 revealed that human accuracy remained unaffected by external stimuli, as participants were instructed to prioritize accuracy over speed to control the speed-accuracy tradeoff (Heitz, 2014). Consequently, to model cognitive response due to external stimuli, we focus on simulating response time rather than choice accuracy, aligning with (Goetschalckx et al., 2024).

3.2. CogReact Framework

Inspired by exploratory analysis (Appendix A.4) 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 (termed math agent) to proficiently solve the designated cognitive task. The second step involves the knowledge transfer from these trained agents to establish mappings from the LSTM agent to human performance metrics. This yields predictions for 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 3.3 and the last two steps in Section 3.4.

3.3. Math Agent and Transfer to Humans

To simulate the impact of time pressure, it is imperative to first predict user 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 net-

Table 1. Evaluation of selected baselines in math reasoning task. For MAPE, we show its mean value (Mean), standard deviation (STD). More complete results with more baselines are in Table 4.

		MAPE	
Model Input Type	Model Type Name	Mean	STD
I. Task: Video, Feedback: Video	hGRU	0.3335	0.2486
	LSTM + AlexNet	0.3344	0.2602
	LSTM + VGG-16	0.3355	0.2708
	LSTM + ViT-B-16	0.3339	0.2573
	MLP + 3D ResNet	0.3330	0.2507
II. Task: Encoded String, Feedback: Video	LSTM-V1 + 3D ResNet	0.3334	0.261
	LSTM-V2 + 3D ResNet	0.3376	0.2169
	MLP + 3D ResNet	0.3331	0.2550
	Transformer + 3D ResNet	0.3306	0.2496
	CogReact	**0.2999**	**0.2318**
III. Task: Numeric, Feedback: Video	LSTM-V1 + 3D ResNet	0.3341	0.2617
	LSTM-V2 + 3D ResNet	0.3286	0.2538
	MLP + 3D ResNet	0.3333	0.2579
	Transformer + 3D ResNet	0.3315	0.2526
IV. Task: Numeric, Feedback: Numeric	Decision Tree	0.3617	0.3640
	Linear Regression	0.3595	0.3608
	LSTM	0.3059	0.2434
	MLP	0.3293	0.2441
	Random Forest	0.3650	0.3684
	SVM	0.3299	0.3108
	Transformer	0.3052	0.2446
	CogReact	**0.2703**	**0.2224**
V. Task: Encoded String, Feedback: Numeric	Decision Tree	0.3639	0.3639
	Linear Regression	0.3512	0.3469
	LSTM	0.3278	0.2478
	MLP	0.3333	0.2577
	Random Forest	0.3600	0.3630
	SVM	0.3245	0.3101
	Transformer	0.3299	0.2481

works (Roseboom et al., 2019), we have devised a baseline prediction model. Specifically, (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 activations. 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 SVC, *a classification model in SVM*) and response time (via SVR, *a regression model in SVM*). More details on the rationale of the math answer agent and SVM models are provided in Appendix A.5, A.6, Fig. 7.

3.4. Hybrid DRL to Simulate Stimuli Perturbation

To simulate how dynamic time pressure perturbs human logical reasoning process, we conceptualize this 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 user 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 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 a 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 of user response time (Details in Appendix A.7).

4. Evaluation

4.1. Human Response Time Simulation Performance

We first demonstrate the effectiveness of our ***CogReact*** framework in human response time simulation by comparing with baselines using different stimuli encoding schemes.

The model input is composed of three parts: math task stimuli, environmental feedback stimuli (time pressure), and task question ID. The question ID is a numeric value indicating the trial number for participants in the math task. Our exploratory analysis in Appendix A.4 has depicted the relevance of question ID in human response time.

In ***CogReact***, 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,

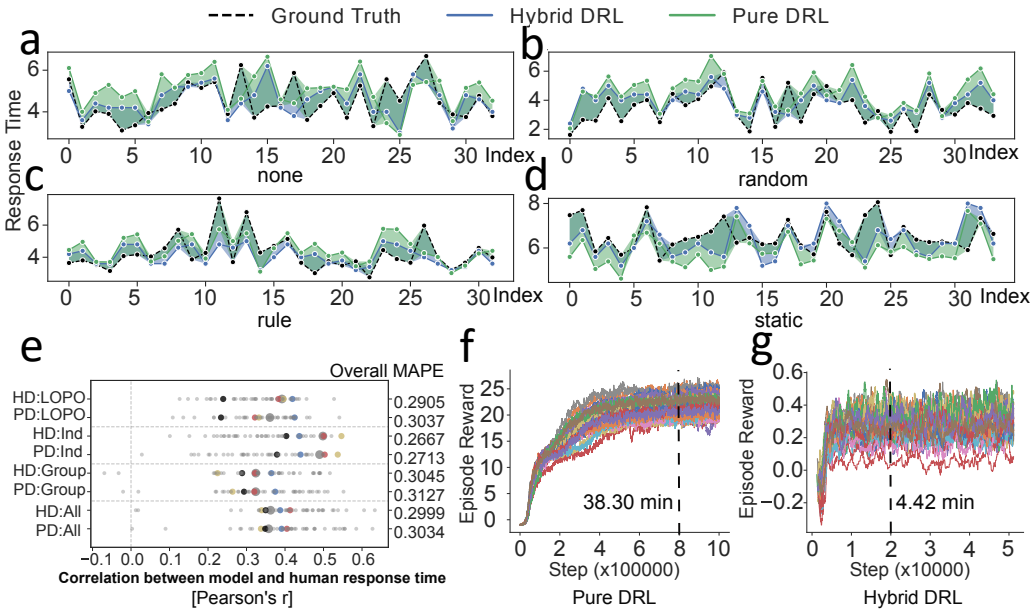

Figure 2. Experimental results in the logical reasoning task. a,b,c,d: Examples of user response time in chronological order from one participant in each group predicted by the Hybrid/Pure DRL agent in LOPO-level training, compared with ground truth. e: Pearson correlation between predictions from Hybrid/ 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) models.

we can treat both task stimuli and feedback stimuli as numeric values directly or we can put both math task stimuli and feedback stimuli into the whole video as model input, just as humans watch them in the task. Therefore, we evaluate 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 4. More details of each model input type and the training/testing hyperparameters/process are depicted in Appendix A.9.

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.9.

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 times exhibit 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.9.

Results are depicted in Table 1 and Table 4, showing that ***CogReact*** in both Type II and Type IV has the best response time prediction performance (lowest MAPE) by comparing with other models in both the same and different model input types. Specifically, CogReact in Type IV has a more efficient representation (numeric encoding) during the model inference process and achieves the best performance. We also perform statistical analysis in both Kolmogorov-Smirnov test and Permutation test because they do not necessarily assume the data to be normally distributed. We applied them in all baselines. Results in Table 5 show that CogReact in Type II achieves significantly lower MAPE ($p < 0.001$) than most baselines except for LSTM/Transformer in Type IV (numeric for task/feedback). This is due to different task/feedback modality encoding since CogReact uses Type II (Task: String, Feedback: Video). When we apply CogReact to numeric encoding of Type IV as well, results in Table 6 show that ours can also achieve significantly lower MAPE ($p < 0.001$) than LSTM/Transformer in Type IV.

MAPE Variance. Despite the superiority, we observed high variance in MAPE, largely driven by individual differences across users and variability in the math trials they completed.

These factors interact, amplifying prediction error variance. To isolate their effects, we held one factor constant while averaging over the other: (a) Fixing Users: Averaging predicted and actual response times across all trials per user, then computing MAPE, yielded a mean of 0.1388 (STD: 0.0641, 95% CI: [0.0316, 0.2378]). (b) Fixing Trials: Averaging across users per trial and computing MAPE resulted in a mean of 0.1403 (STD: 0.0865, 95% CI: [0.0292, 0.2855]). Both approaches significantly reduced variance compared to the overall results (Mean: 0.2703, STD: 0.2224, 95% CI: [0.0093, 0.7631]), suggesting that the high variability stems from the interaction of user and trial differences.

The performance improvement stems from the entire framework, including useful 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 unique importance of each component.

4.2. Importance of Task Encoding with Math Agent

To demonstrate that the math answer agent indeed captures representative features from math questions, in the first ablation study, we compare SVM models (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.5) of raw numbers as input, along with the question ID. The SVM performance in the three settings is depicted in Table 3. 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 other settings, underscoring the effectiveness of the math answer agent in capturing representative math question features and the feasibility of predicting user baseline performance in ideal conditions without environmental stimuli using SVM.

4.3. Why Does the Logical Reasoning Agent Work?

The second ablation study explores why the math logical reasoning agent can extract useful features from math questions (first step of our framework). We answer this question by exploring its math task solving performance under different numbers of output neurons from 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-user responses. Hence, we prepare a separate dataset that is independent of the human dataset to train the agent. Finally, we traverse all possible combinations of three numbers in math questions and got a dataset containing 20,414 samples, which is split into training set (80%) and testing set (20%). Given that the first two numbers of math questions are both two-digit, 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.9(b)). The confusion matrix (Fig.9(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 numbers, 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.

4.4. Importance of Integrating DDM into DRL Agents

The third ablation 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 the *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, it directly takes the entire time pressure 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 (details in Appendix A.8, Fig. 8). Moreover, we also directly remove the whole hybrid DRL agent and only use 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 below: (a). **General-level** involves splitting the entire dataset into training (80%) and testing (20%) sets for overall model evaluation. (b). **Group-level** trains and tests a specific model using data from each group, revealing performance across different time pressure stimuli. (c). **Individual-level** trains and tests a model using data from a specific participant, assessing personalized model feasibility incorporating subject-specific

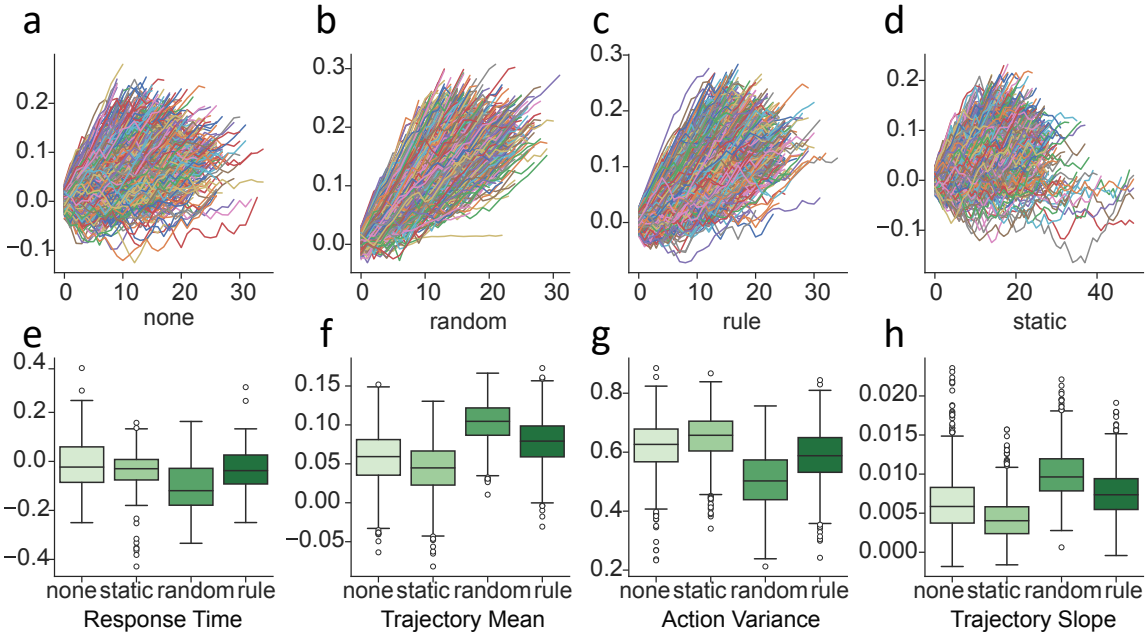

Figure 3. a,b,c,d: Time pressure effect trajectories of four groups, respectively. e: Box plot of relative response time change across four groups in the whole dataset. f,g,h: Box plot of mean value of time pressure effect trajectories (f), standard deviation of action trajectories (g), slope of time 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.

behavioral differences. Shuffling is applied to training and testing sets to prevent overfitting artifacts. (d). As shuffled testing disrupts the temporal trend of user response time across different math trials, we incorporate **Leave-One-Participant-Out (LOPO) training** 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 model performance in capturing temporal trends of response time.

Fig. 10 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 an 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. 2(e), further supporting this conclusion. Fig. 2(e) also reveals that the hybrid DRL agent exhibits a 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 user 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. 2(a,b,c,d) clearly demonstrate that the hybrid DRL agent more accurately captures the trend of user response time compared to the pure DRL agent.

4.5. Training Efficiency

The training curves for both hybrid and pure agents are presented in Fig. 2(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 meaning of one step differs 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. 2(f,g) indicate that the hybrid DRL agent converges in approximately one-tenth 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 to improve training efficiency.

4.6. Interpretability

An essential advantage of the cognition-inspired hybrid DRL agent is its interpretability, compared to deep learning 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 user logical reasoning process. Therefore, visualizing the trajectories of the hybrid DRL agent enables extracting new insights of how time pressure stimuli affect the human logical reasoning process.

We explore this benefit in Fig. 3(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. 3(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. 3(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. 3(g)) and the highest average value and slope for the time pressure effect trajectories (Fig. 3(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 the *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.4, Fig. 6(e)), where participants in the *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, training efficiency, and interpretability) highlights the advantages of the hybrid DRL approach in capturing the nuanced dynamics of time pressure stimuli on user response time in the logical reasoning process.

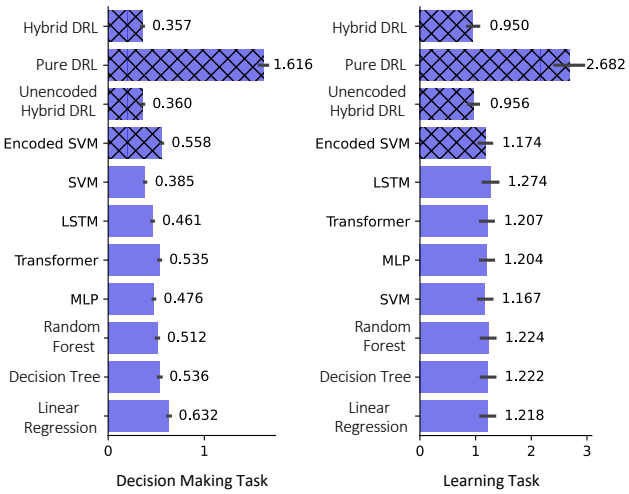

Figure 4. Results in decision making(left) and learning(right) task.

5. Generalization

We further evaluate the generalization ability of our model in two additional public datasets: CPC18[2] for decision making and PeerEdu for learning (Xu et al., 2025). Table 2 depicts the datasets' properties and summarizes diverse tasks and feedback modalities for modeling. Dataset details and our framework adaptation are depicted in Appendix A.11.

Baselines: As both datasets lack video input, we adopt the same baselines as Type IV/V in Table 1, aligning with baselines commonly used in prior work (Bourgin et al., 2019) that benchmarks the CPC18 dataset. For fair comparison, in the PeerEdu dataset, we use the same embeddings from OpenAI's *text-embedding-3-small* model to encode textual data for model input of baselines.

Performance: Consistent with prior experiments, we use MAPE to evaluate modeling error. As shown in Fig. 4, our hybrid DRL model achieves the lowest MAPE across both datasets compared to all baselines, highlighting its effectiveness in generalizing across diverse tasks and feedback modalities. Moreover, the statistical analyses (same with previous evaluations) also show significant improvement of our model over all baseline models in Table 7 and Table 8.

Ablation Study: We further explore the role of each component using variants of our framework, as shown in the

[2]https://cpc-18.com/data/

Table 2. Task/feedback information and dataset properties.

Task Information				Simulation Modality			Dataset Information		
Task Type	Response Type	User Action	Cognitive Response	Task	Feedback	Stage 1	Source	Size	User
Math Reasoning	Active	Binary	Response Time	String	Visual	Math Agent	Ours	21,157	50
Decision Making	Active	Binary	Response Time	Numeric	Numeric	Risk Agent	Public	30,489	240
Learning	Passive	Continuous	Curiosity	Textual	Textual	LLM Agent	Public	12,804	300

textured bar results in Fig. 4. The pure DRL variant removes the DDM component from the DRL loop, where the action space directly represents response time/curiosity changes without segmenting the evidence accumulation process (similar to Section 4.4). The unencoded hybrid DRL variant removes the first step of the framework (risk/LLM agent for embedding extraction), while the encoded SVM variant only includes the first two steps (risk/LLM agent + SVM) without the DDM or the DRL loop.

Fig. 4 shows that all variants perform worse than the full framework (hybrid DRL), underscoring the unique and critical roles of each component. Moreover, the statistical analyses (same with previous evaluations) also show significant improvement of our model over most ablation models in Table 7 and Table 8. Removing the first step (unencoded hybrid DRL) causes a slight performance drop, whereas removing the DDM (pure DRL) leads to a significant error increase, performing worse than all baselines. This highlights the dominant role of incorporating DDM into the DRL loop in modeling feedback modulation effects on cognitive responses. Additionally, removing both DDM and DRL (encoded SVM) also degrades performance, emphasizing the importance of the hybrid DRL loop.

Interestingly, the encoded SVM variant (risk/LLM agent + SVM) performs worse than the straightforward SVM baseline (without risk/LLM agent), suggesting potential drawbacks of using risk/LLM embeddings for SVM. However, removing the risk/LLM agent (unencoded hybrid DRL) results in worse performance compared to the full hybrid DRL model. This discrepancy suggests that, while the extracted features from the risk/LLM agent may not directly benefit the SVM, they still improve the hybrid DRL model by enhancing and expanding the observation space during the evidence accumulation process.

DDM in Deep Learning Models. Our previous findings demonstrated the advantage of integrating DDM into DRL. To investigate whether this performance gain arises primarily from the deep learning (DL) architecture within DRL or from the reinforcement learning (RL) component itself, we conducted an additional study that directly incorporated DDM into DL models without RL. Specifically, we adapted LSTM, MLP, and Transformer architectures-identical to those used in Baseline Model Type IV-with unchanged hyperparameters, modifying them to predict DDM parameters instead of response times. The final response times were then derived from these predicted parameters. Additionally, we introduced a variant of MLP (MLPv2), which shares the same neural network architecture as our Hybrid DRL model, to assess whether the observed performance gains could be attributed solely to the combination of DL and DDM. This experiment was conducted across three datasets, with the same statistical analyses performed in previous evaluations. The results (Fig. 11) indicate that while DL+DDM integration sometimes outperforms standalone DL models, its performance remains significantly inferior to that of the Hybrid DRL model ($p < 0.001$). These findings highlight that the improvement observed in the hybrid DRL model cannot be attributed solely to DL+DDM integration; rather, the RL component plays a critical and complementary role in achieving superior performance.

6. Discussion, Limitations, and Conclusion

We propose a computational framework for simulating environmental stimuli perturbations on human cognitive processes, including logical reasoning, decision-making, and learning, across diverse task and feedback modalities. By integrating the drift-diffusion model from cognitive science with deep reinforcement learning, our framework achieves higher simulation accuracy, improved training efficiency, and enhanced interpretability, capturing the granular effects of dynamic stimuli on cognitive processes. The successful adaptation of our framework for continuous behaviors (e.g., curiosity in learning tasks) lays a foundation for extending it to handle continuous user inputs beyond binary responses. This advancement has the potential to offer new insights into machine learning and neuroscience by fostering computational models that better understand human cognition.

One limitation is our focus on response time simulation. Future extensions could incorporate additional cognitive measures. A potential path involves training task-solving agents, like the logical reasoning agent in math tasks, to emulate human task performance. Building on prior research that highlights the effectiveness of machine learning models in more than 20 cognitive tasks (Yang et al., 2019), our framework could be extended to other domains by linking extracted features from task-solving agents to real user responses using models like SVM. Finally, by dynamically adapting the action and observation spaces to task-specific feedback, the DRL agent could simulate the nuanced effects of stimuli across diverse cognitive scenarios.

Acknowledgments

This work is supported by National Science Foundation CNS-2403124, CNS-2312715, CNS-2128588 and the University of California San Diego Center for Wireless Communications.

Impact Statement

Our framework bridges cognitive science and reinforcement learning by modeling human-like decision-making in dynamic environments. It extends prior work by capturing how environmental stimuli shape cognitive responses, a key challenge in human-in-the-loop AI. This approach enables neuroscientists to simulate cognitive adaptation through reinforcement-based models and supports biologically inspired AI designs and intervention strategies. It also provides cognitive scientists with tools to design behavioral experiments, while helping AI researchers align algorithms with human reasoning. By improving transparency, trust, and usability in adaptive systems, the framework advances human-centered AI and serves as a practical foundation for interdisciplinary research across AI, cognitive science, and neuroscience. Future research inspired by this framework could refine its scope to encompass additional cognitive measures, expand its application to diverse task-solving agents, and deepen its alignment with real-world user data. The framework's adaptability to continuous behaviors and dynamic feedback underscores its potential to drive innovation across cognitive modeling, neuroscience, and machine learning. We anticipate that this research will encourage interdisciplinary exploration into cognitive and behavioral modeling, with minimal societal risks. We adhered to appropriate licenses in using public datasets. Our experiment for human data collection in logical reasoning tasks was approved by the Institutional Review Board (IRB) at our local institution. The data we model is strictly limited to anonymous behavioral responses under experimental tasks, with no links to real-world identities, nor demographics. As such, the model itself does not encode or have access to demographic-sensitive features, reducing the risk of biased outputs, and also reducing the risk of potential usage for justifying employment, acceptance to college, and so on for targeted people.

References

Alexander, C., Paul, M., Michael, M., 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.

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

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

Battleday, R. M., Peterson, J. C., and Griffiths, T. L. 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.

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

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

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

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Cheng, S.-Y. 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.

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

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

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

Costa, J., Guimbretière, F., Jung, M. F., and Choudhury, T. 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.

Daitch, A. L., Foster, B. L., Schrouff, J., Rangarajan, V., Kaşikçi, I., Gattas, S., and Parvizi, J. 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.

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

Do, S., Chang, M., and Lee, B. 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.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

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

Erev, I., Ert, E., Plonsky, O., Cohen, D., and Cohen, O. From anomalies to forecasts: Toward a descriptive model of decisions under risk, under ambiguity, and from experience. *Psychological review*, 124(4):369, 2017.

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

Fudenberg, D., Newey, W., Strack, P., and Strzalecki, T. Testing the drift-diffusion model. *Proceedings of the National Academy of Sciences*, 117(52):33141–33148, 2020.

Goetschalckx, L., Govindarajan, L. N., Karkada Ashok, A., Ahuja, A., Sheinberg, D., and Serre, T. Computing a human-like reaction time metric from stable recurrent vision models. *Advances in Neural Information Processing Systems*, 36, 2024.

Golan, T., Raju, P. C., and Kriegeskorte, N. 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.

Han, J. and Moraga, C. 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.

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

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

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

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

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

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

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

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Kumbhar, O., Sizikova, E., Majaj, N., and Pelli, D. G. Anytime prediction as a model of human reaction time. *arXiv preprint arXiv:2011.12859*, 2020.

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

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

Lin, C.-T., Chen, S.-A., Chiu, T.-T., Lin, H.-Z., and Ko, L.-W. Spatial and temporal eeg dynamics of dual-task driving performance. *Journal of neuroengineering and rehabilitation*, 8(1):1–13, 2011.

Linsley, D., Kim, J., Veerabadran, V., Windolf, C., and Serre, T. Learning long-range spatial dependencies with horizontal gated recurrent units. *Advances in neural information processing systems*, 31, 2018.

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

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

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

Moore, D. A. and Tenney, E. R. 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.

Noti, G., Levi, E., Kolumbus, Y., and Daniely, A. Behavior-based machine-learning: A hybrid approach for predicting human decision making. *arXiv preprint arXiv:1611.10228*, 2016.

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

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., Red Hook, NY, USA, 2019.

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

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

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

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

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

Plonsky, O., Erev, I., Hazan, T., and Tennenholtz, M. Psychological forest: Predicting human behavior. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. 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.

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

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

Rodríguez, P., Bautista, M. A., Gonzalez, J., and Escalera, S. Beyond one-hot encoding: Lower dimensional target embedding. *Image and Vision Computing*, 75:21–31, 2018.

Roseboom, W., Fountas, Z., Nikiforou, K., Bhowmik, D., Shanahan, M., and Seth, A. K. Activity in perceptual classification networks as a basis for human subjective time perception. *Nature communications*, 10(1):1–9, 2019.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

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

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

Singh, P., Peterson, J. C., Battleday, R. M., and Griffiths, T. L. End-to-end deep prototype and exemplar models for predicting human behavior. *arXiv preprint arXiv:2007.08723*, 2020.

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

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

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

Song, H. F., Yang, G. R., and Wang, X.-J. Reward-based training of recurrent neural networks for cognitive and value-based tasks. *Elife*, 6:e21492, 2017.

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

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

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision, 2015. URL https://arxiv.org/abs/1512.00567.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. 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.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Viejo, G., Khamassi, M., Brovelli, A., and Girard, B. 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.

Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., and Wei, F. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*, 2023.

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

Whittaker, S., Kalnikaite, V., Hollis, V., and Guydish, A. '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.

Xu, S., Hu, D., Wang, R., and Zhang, X. Peeredu: Bootstrapping online learning behaviors via asynchronous area of interest sharing from peer gaze. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2025.

Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T., and Wang, X.-J. Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, 22(2):297–306, 2019.

Zaremba, W. and Sutskever, I. Learning to execute. *arXiv preprint arXiv:1410.4615*, 2014.

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

A. Appendix

A.1. Ethics Statement

Our experiment for human data collection in logical reasoning tasks was approved by the Institutional Review Board (IRB) at 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.

A.2. 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. 5. 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 subtracted Num_2 from Num_1 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. 6(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.3.4). 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) at 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.3.

A.3. Dataset Collection

A.3.1. 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. 5(a)). Participants were recruited by email groups at 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 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) at our local institution. We have obtained informed consent from all participants before study.

A.3.2. 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.3.3. 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.3.4. 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. *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.4. 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.4.1. 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. 6(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. 6(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):

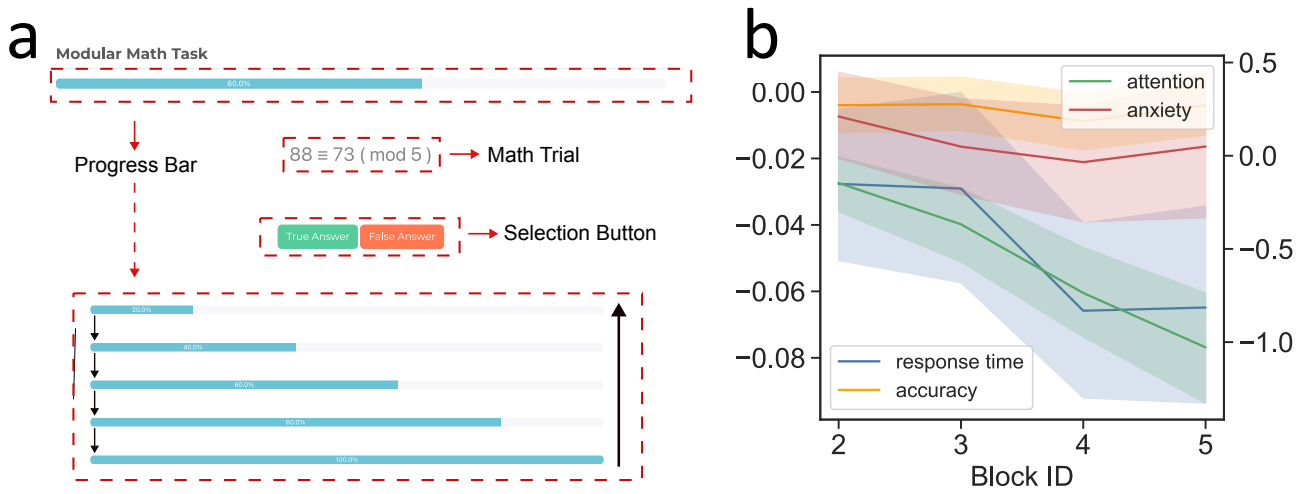

Figure 5. a: Math arithmetic task and time pressure feedback. Each math trial is composed of two two-digit numbers Num_1, Num_2 and one one-digit numer Num_3, formatted as: $Num_1 \equiv Num_2 \,(\, mod\, Num_3)$. To solve this question, participants first subtract Num_2 from Num_1 and judge whether 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.

$p = 0.026 < 0.05$, $Block_3$ (-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. 6(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.4.2. 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. 6(f)), or Date ($F_{1,32} = 0.861, P = 0.36 > 0.05$) (Fig. 6(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 user 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.5. 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 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. 7, 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 ≡ 26(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.

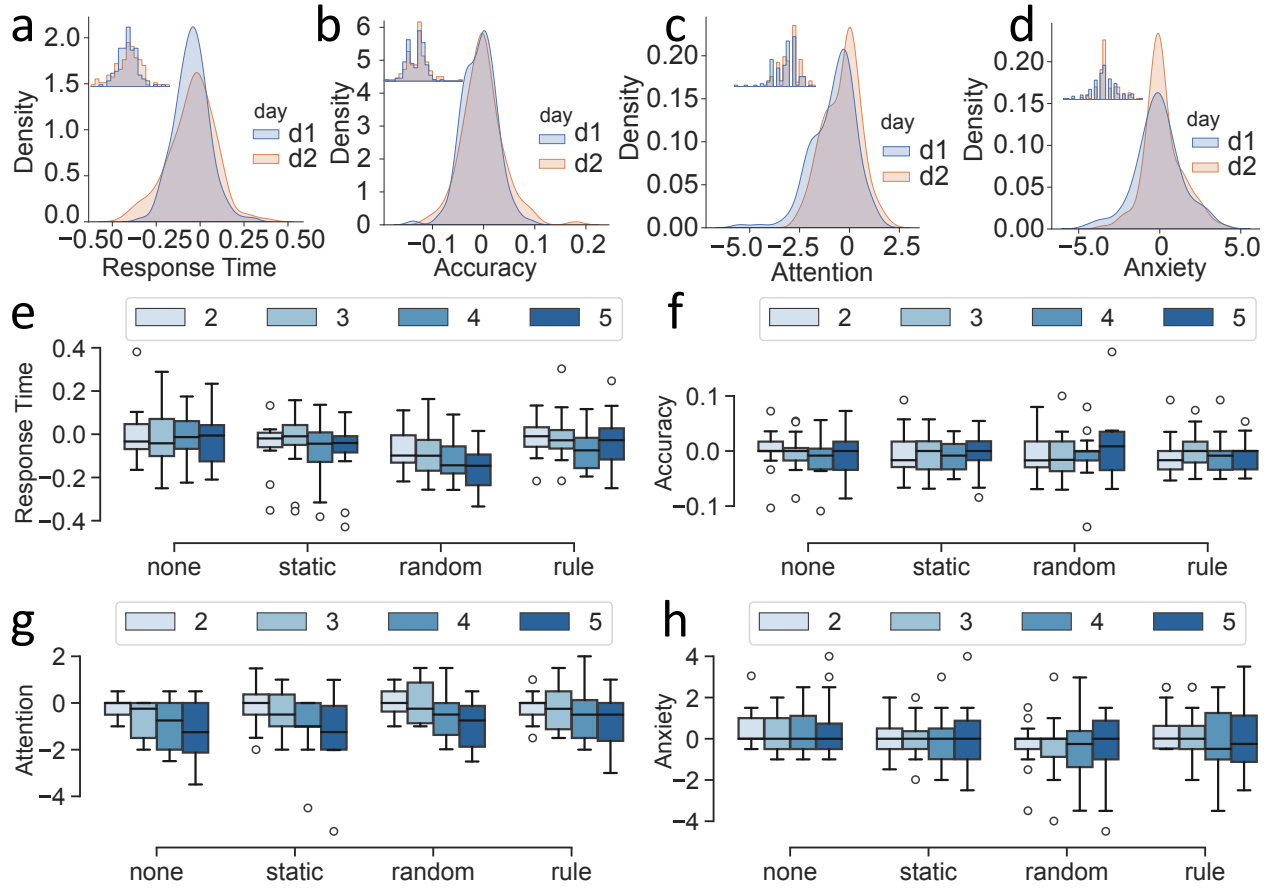

Figure 6. a,b,c,d: overall distribution of relative change of response time (a), accuracy (b), attention (c), and anxiety (d), respectively, across 2 days. e,f,g,h: box plot of relative change of response time (e), accuracy (f), attention (g), and anxiety (h), respectively, across 4 groups and 4 blocks.

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%).

A.6. 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 user 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.7. Hybrid DRL Agent with Drift-Diffusion Model

A.7.1. 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 monotonic 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 monotonic because such trajectory is modulated and modified by the DRL agent adaptively according to the environmental stimuli.

A.7.2. 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

Table 3. 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**

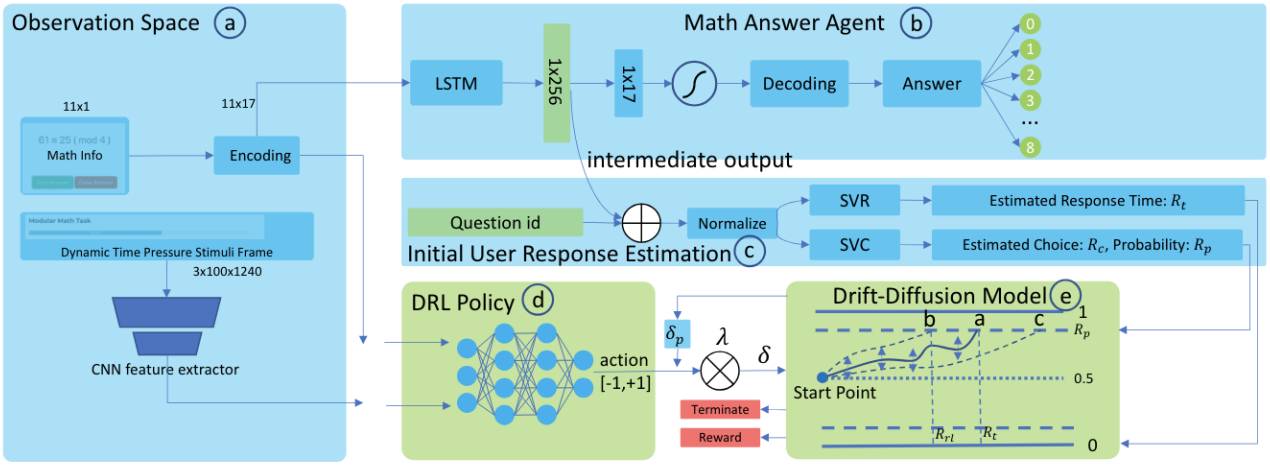

Figure 7. The detailed architecture of our CogReact framework. First, we use math questions to train a math answer agent to solve them without considering users' response. Second, for each math question, we transfer features extracted from LSTM layer in math answer agent without time pressure to make 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.

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.7.3. 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.7.4. 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.7.5. 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.7.6. 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:

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

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.7.7. 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.8. 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.8.1. 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.8.2. 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 pressure, different from

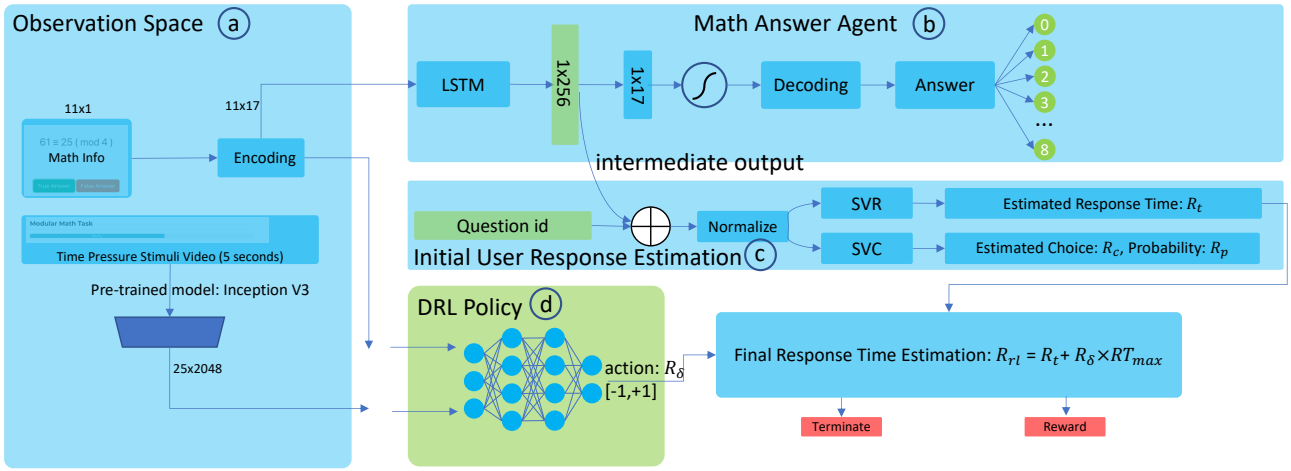

Figure 8. The detailed architecture of the pure DRL agent without drift-diffusion model.

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.8.3. 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.8.4. 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.8.5. 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:

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

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.8.6. 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).

A.9. 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).

- 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. The settings of these baseline models (Decision Tree, Linear Regression, LSTM, MLP, Random Forest, SVM, Transformer) are the same as Baseline Type 4.

A.10. 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 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. 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.

The principles underlying CogReact may be extended to the analysis of neural and physiological responses to external stimuli. Although such data—whether derived from neural activity or wearable sensors—pose significant challenges for direct modeling due to their high-frequency, noisy time-series nature, they hold considerable potential. Specifically, these recordings can serve as proxies for the trajectory of evidence accumulation in human decision-making and other cognitive processes. Mapping these novel forms of evidence accumulation within specific cognitive tasks offers a promising avenue for capturing human cognition at a highly fine-grained temporal and representational level.

A.11. Generalization in New Tasks and Datasets

A.11.1. Datasets

We further evaluate the generalization ability of our model in two additional public datasets. The first is CPC18, a widely used benchmark for modeling human cognition in a decision making task (Bourgin et al., 2019). In this dataset, participants engaged in a gambling game where they made binary decisions in each trial. Each choice offers different reward/loss with certain probabilities, and feedback indicates the alternative reward or loss if the participant had selected the other option. Consistent with previous experiments, the model predicts user response times per trial based on the model input including task information (reward/loss probability configurations of two choices) and feedback information (alternative reward/loss), represented as numeric arrays. We obtained a total of 30,489 trials from 240 participants with valid response time from the raw dataset.

The second public dataset, PeerEdu (Xu et al., 2025), captures students' cognitive states during learning with external peer feedback. Students watched online video lectures while their cognitive states, were passively recorded using sensors. Specifically, we use one cognitive state named curiosity (continuous value from 0 to 1) for evaluation, which is the most significantly impacted by peer feedback from (Xu et al., 2025). Unlike previous tasks requiring active but binary human choice input, PeerEdu focuses on passive yet continuous cognitive states without students' active input. Peer feedback was delivered by highlighting specific regions on video lecture slides that peer students focused on, updated continuously based on lecture progress.

To simulate curiosity in this learning task, each lecture was

divided into transcripts, with each transcript representing a sentence spoken by the lecturer. Task information includes transcript content, while feedback information comprises the highlighted text in peer feedback regions on the slides. The model predicts curiosity for each transcript by taking both task and feedback information (textual format) as input. We obtained a total of 12,804 samples from 300 students in PeerEdu, where each sample corresponds to the curiosity of one student during one transcript with specific feedback.

A.11.2. Framework Adaptation

To adapt our framework for the decision making task, we replace the math agent with an LSTM-based risk agent in the first step in Fig. 1. This risk agent, using the same architecture as before, predicts potential reward/loss (feedback information) from task inputs in each gambling trial, extracting risk features for the SVM model in the second step. The second step keeps the same as Fig. 1 to predict user choice and response time without feedback, which are used to generate the evidence accumulation process in the DDM step. The DRL loop incorporates an adjusted observation space to handle the task and feedback information in decision making, while maintaining the same action space and reward functions as the previous logical reasoning task for simulating response time changes.

For the learning task adaptation, we replace the math agent in our framework with a large language model (LLM), referred to as the LLM agent, to extract features from textual task and feedback information, leveraging the strong textual data mining capabilities of LLMs (Wang et al., 2023). Specifically, we use OpenAI's *text-embedding-3-small* model to generate embeddings from both task and feedback information, which are then input into SVM models in the second step of Fig. 1. To handle the absence of binary user input, we categorize curiosity values into high or low levels based on their position relative to the median, enabling SVM models to predict curiosity level (SVC: high or low) and actual value (SVR), similar to predicting binary choice and response time in previous tasks. This approach enables adaptation to continuous response modeling without significant changes to the framework and can be extended to other continuous behaviors in the future. The DRL loop incorporates an updated observation space to process embeddings from task and feedback information, with action space and reward functions adjusted to align with the curiosity scale.

Table 4. Results for all baseline model performance on response time simulation in Math Task. 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
	CogReact	**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
	CogReact	**0.2703**	**0.2224**	**0.0093**	**0.7631**
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

Table 5. Statistical results by comparing our CogReact model in Type II (Task: Encoded String, Feedback: Video) with each of baseline model respectively in the Math Task.

		Statistical Tests for CogReact in Type II	
Model Input Type	Model Type Name	Kolmogorov-Smirnov	Permutation
Task: Video Feedback: Video	hGRU	$p < 0.001$	$p < 0.001$
	LSTM + AlexNet	$p < 0.001$	$p < 0.001$
	LSTM + VGG-16	$p < 0.001$	$p < 0.001$
	LSTM + ViT-B-16	$p < 0.001$	$p < 0.001$
	MLP + 3D ResNet	$p < 0.001$	$p < 0.001$
Task: Encoded String Feedback: Video	LSTM-V1 + 3D ResNet	$p < 0.001$	$p < 0.001$
	LSTM-V2 + 3D ResNet	$p < 0.001$	$p < 0.001$
	MLP + 3D ResNet	$p < 0.001$	$p < 0.001$
	Transformer + 3D ResNet	$p < 0.001$	$p < 0.001$
Task: Numeric Value Feedback: Video	LSTM-V1 + 3D ResNet	$p < 0.001$	$p < 0.001$
	LSTM-V2 + 3D ResNet	$p < 0.001$	$p < 0.001$
	MLP + 3D ResNet	$p < 0.001$	$p < 0.001$
	Transformer + 3D ResNet	$p < 0.001$	$p < 0.001$
Task: Numeric Value Feedback: Numeric Value	Decision Tree	$p < 0.001$	$p < 0.001$
	Linear Regression	$p < 0.001$	$p < 0.001$
	LSTM	$p = 0.819$	$p = 0.263$
	MLP	$p < 0.001$	$p < 0.001$
	Random Forest	$p < 0.001$	$p < 0.001$
	SVM	$p < 0.001$	$p < 0.001$
	Transformer	$p = 0.920$	$p = 0.332$
Task: Encoded String Feedback: Numeric Value	Decision Tree	$p < 0.001$	$p < 0.001$
	Linear Regression	$p < 0.001$	$p < 0.001$
	LSTM	$p < 0.001$	$p < 0.001$
	MLP	$p < 0.001$	$p < 0.001$
	Random Forest	$p < 0.001$	$p < 0.001$
	SVM	$p < 0.001$	$p < 0.001$
	Transformer	$p < 0.001$	$p < 0.001$

Table 6. Statistical results by comparing our CogReact model in Type IV (Task: Numeric Value, Feedback: Numeric Value) with each of baseline model respectively in the Math Task.

		Statistical Tests for CogReact in Type IV	
Model Input Type	Model Type Name	Kolmogorov-Smirnov	Permutation
Task: Video Feedback: Video	hGRU	$p < 0.001$	$p < 0.001$
	LSTM + AlexNet	$p < 0.001$	$p < 0.001$
	LSTM + VGG-16	$p < 0.001$	$p < 0.001$
	LSTM + ViT-B-16	$p < 0.001$	$p < 0.001$
	MLP + 3D ResNet	$p < 0.001$	$p < 0.001$
Task: Encoded String Feedback: Video	LSTM-V1 + 3D ResNet	$p < 0.001$	$p < 0.001$
	LSTM-V2 + 3D ResNet	$p < 0.001$	$p < 0.001$
	MLP + 3D ResNet	$p < 0.001$	$p < 0.001$
	Transformer + 3D ResNet	$p < 0.001$	$p < 0.001$
Task: Numeric Value Feedback: Video	LSTM-V1 + 3D ResNet	$p < 0.001$	$p < 0.001$
	LSTM-V2 + 3D ResNet	$p < 0.001$	$p < 0.001$
	MLP + 3D ResNet	$p < 0.001$	$p < 0.001$
	Transformer + 3D ResNet	$p < 0.001$	$p < 0.001$
Task: Numeric Value Feedback: Numeric Value	Decision Tree	$p < 0.001$	$p < 0.001$
	Linear Regression	$p < 0.001$	$p < 0.001$
	LSTM	$p < 0.001$	$p < 0.001$
	MLP	$p < 0.001$	$p < 0.001$
	Random Forest	$p < 0.001$	$p < 0.001$
	SVM	$p < 0.001$	$p < 0.001$
	Transformer	$p < 0.001$	$p < 0.001$
Task: Encoded String Feedback: Numeric Value	Decision Tree	$p < 0.001$	$p < 0.001$
	Linear Regression	$p < 0.001$	$p < 0.001$
	LSTM	$p < 0.001$	$p < 0.001$
	MLP	$p < 0.001$	$p < 0.001$
	Random Forest	$p < 0.001$	$p < 0.001$
	SVM	$p < 0.001$	$p < 0.001$
	Transformer	$p < 0.001$	$p < 0.001$

Table 7. Statistical results by comparing CogReact with each of baseline model and ablation model (our model variants in ablation studies) respectively in PeerEdu dataset.

		Statistical Tests for CogReact in PeerEdu	
Model Input Type	Model Type Name	Kolmogorov-Smirnov	Permutation
Task: Numeric Value Feedback: Numeric Value	Pure DRL	$p < 0.001$	$p < 0.001$
	Unencoded Hybrid DRL	$p = 0.037$	$p < 0.001$
	Encoded SVM	$p < 0.001$	$p < 0.001$
	LSTM	$p < 0.001$	$p < 0.001$
	Transformer	$p < 0.001$	$p < 0.001$
	MLP	$p < 0.001$	$p < 0.001$
	SVM	$p < 0.001$	$p < 0.001$
	Random Forest	$p < 0.001$	$p < 0.001$
	Decision Tree	$p < 0.001$	$p < 0.001$
	Linear Regression	$p < 0.001$	$p < 0.001$

Table 8. Statistical results by comparing CogReact with each of baseline model and ablation model (our model variants in ablation studies) respectively in CPC dataset.

		Statistical Tests for CogReact in CPC	
Model Input Type	Model Type Name	Kolmogorov-Smirnov	Permutation
Task: Numeric Value Feedback: Numeric Value	Pure DRL	$p < 0.001$	$p < 0.001$
	Unencoded Hybrid DRL	$p = 0.058$	$p = 0.677$
	Encoded SVM	$p < 0.001$	$p < 0.001$
	LSTM	$p < 0.001$	$p < 0.001$
	Transformer	$p < 0.001$	$p < 0.001$
	MLP	$p < 0.001$	$p < 0.001$
	SVM	$p < 0.001$	$p < 0.001$
	Random Forest	$p < 0.001$	$p < 0.001$
	Decision Tree	$p < 0.001$	$p < 0.001$
	Linear Regression	$p < 0.001$	$p < 0.001$

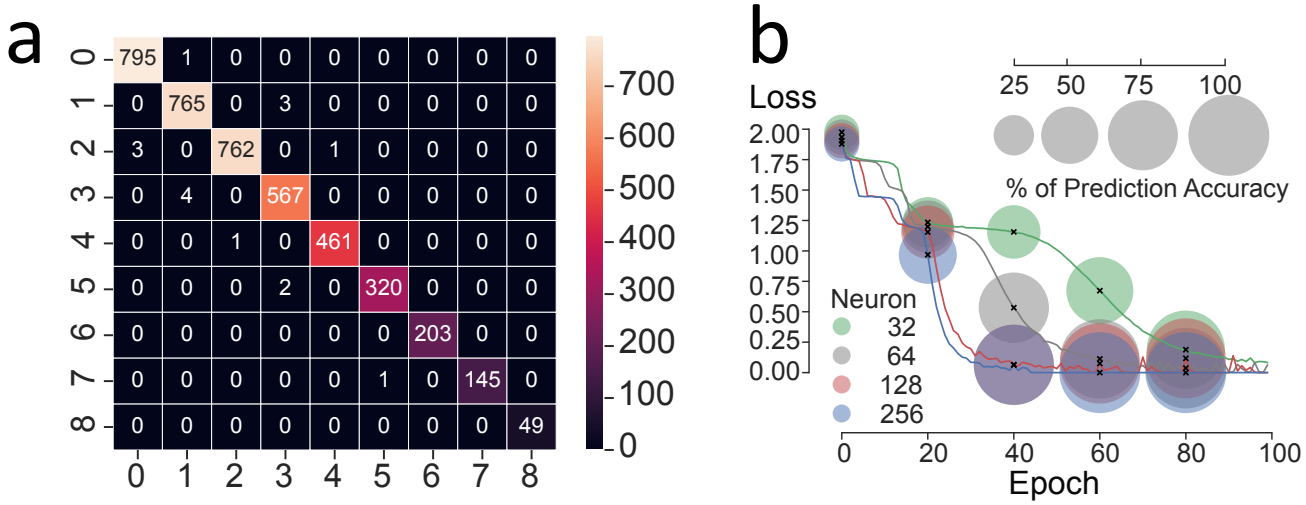

Figure 9. a. Confusion matrix (x axis: ground truth, y axis: prediction) for testing set prediction of the logical reasoning agent (LSTM neuron = 256). b. Training loss and accuracy with training epochs across four kinds of LSTM neurons of the logical reasoning agent.

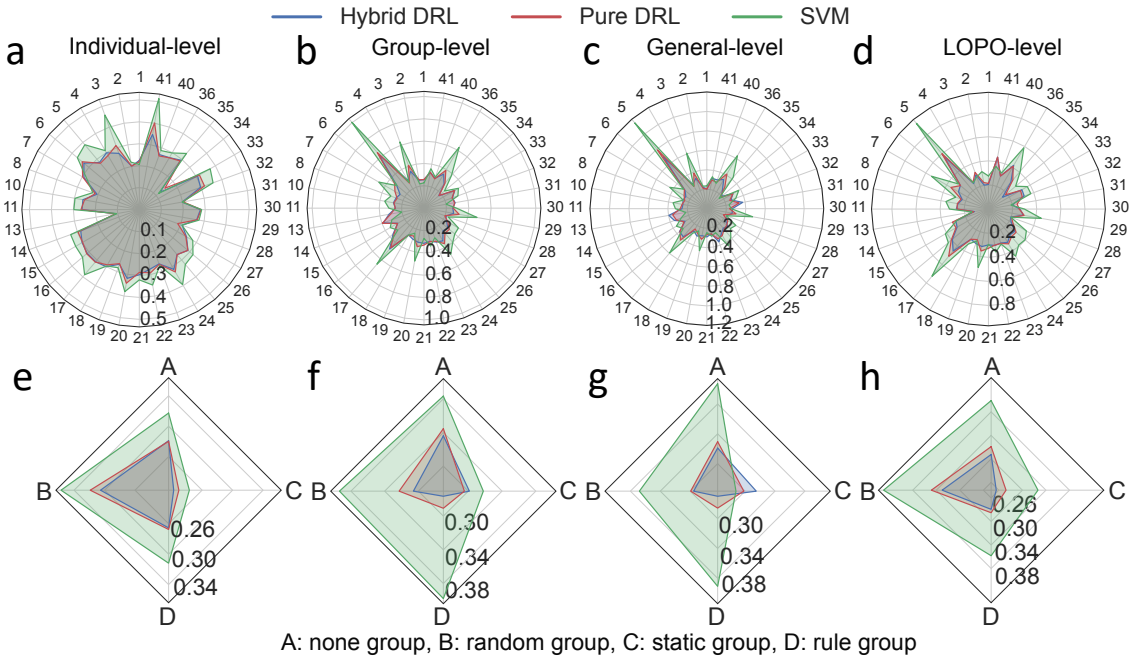

Figure 10. Evaluation results in the logical reasoning task for MAPE in different levels. a,b,c,d,e,f,g,h: Average MAPE for each participant (a,b,c,d)/group (e,f,g,h) in predictions of testing set from Hybrid DRL agent, Pure DRL agent, and SVM model in four training strategies (a,e. Individual-Level, 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).

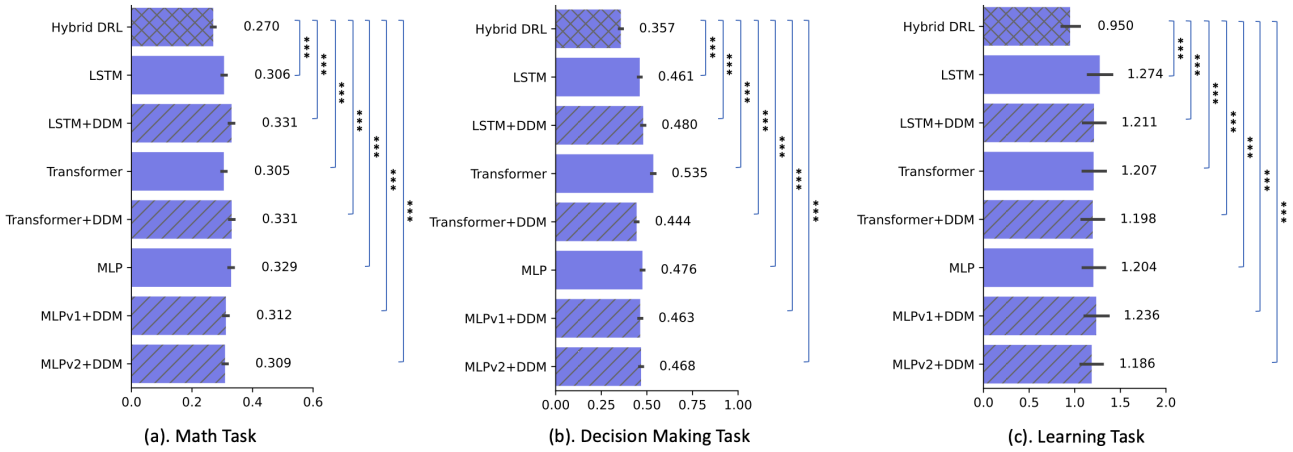

Figure 11. Comparison results between our CogReact Model (Hybrid DRL) and baseline deep learning models with / without DDM in three datasets. For statistical analysis with both Kolmogorov-Smirnov test and Permutation test, $*$ indicates $p < 0.05$, $**$ indicates $p < 0.01$, $***$ indicates $p < 0.001$. The results show that the MAPE of our model is significantly ($p < 0.001$) smaller than all deep learning models with / without DDM in all three datasets.