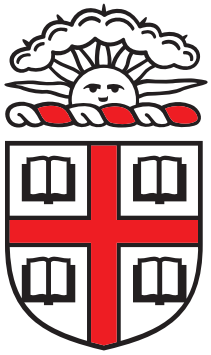
Abstract of "Computational and Behavioral Mechanisms of In-Context Learning", by Abdullah P. Rashed Ahmed, Ph.D., Brown University, May 2026

Sensory features of the environment change continuously, often in context-dependent ways that alter their underlying statistical structure. Recognizing and responding appropriately to such changes requires organisms to identify and track features of the local context, using this information to interpret ongoing sensory experience and update their predictions accordingly. To investigate this, we introduce a novel predictive inference paradigm, the Bouncing Ball task, in which observers predict the partially observable color of a moving ball. Critically, the probability of a color change depends on two latent variables that participants must estimate on a trial-by-trial basis. Using the ideal Bayesian observer, we define the optimal behavioral signatures of in-context learning and demonstrate that both humans and recurrent neural networks trained on the task, exhibit these signatures. However, we observe asymmetries in learning outcomes for humans and models, suggesting differences in capacity for evidence accumulation of each latent variable. To identify the computational mechanisms learned by the recurrent models, we probed their learned representations using elastic net regularization, temporal dynamics analysis, and targeted state interventions to identify the minimal neural sub-circuit that drives adaptive, in-context learning. These analyses revealed that just a single LSTM unit encoded each task parameter through functionally independent leaky integrator circuits. Our findings reveal fundamental computational components necessary for in-context learning and provide a bridge between normative Bayesian theories of human flexibility, interpretable dynamics in recurrent neural networks, and human neurophysiology.

Computational and Behavioral Mechanisms of In-Context Learning



BROWN

Abdullah P. Rashed Ahmed

B.S., Boston College, 2014

M.S., Georgia Institute of Technology, 2016

Thesis dissertation submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in the Department of Neuroscience at Brown University

Providence, Rhode Island

May 2026

This dissertation by Abdullah P. Rashed Ahmed is accepted in its present form by the Department of Neuroscience as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date _____ _____

Thomas Serre, Advisor

Recommended to the Graduate Council

Date _____ _____

Matthew Nassar, Reader

Date _____ _____

David Sheinberg, Reader

Date _____ _____

Michael Frank, Reader

Date _____ _____

Joshua Gold, Reader

Approved by the Graduate Council

Date _____ _____

Janet Blume, Interim Dean of the Graduate School

Curriculum Vitae

Education

2018–2025 Ph.D., Neuroscience
Brown University (Providence, RI, USA)

2015-2016
M.Sc., Computer Science
Georgia Institute of Technology (Atlanta, GA, USA)

2010–2014
B.Sc., Physics
Boston College (Newton, MA, USA)

Publications

C. Cosmo, Y.A. Berlow, K.A. Grisanzio, S.L. Fleming, **A.P. Rashed Ahmed**, M.C. Brennan, L.L. Carpenter, N.S. Philip, "Transdiagnostic Symptom Subtypes to Predict Response to Therapeutic Transcranial Magnetic Stimulation in Major Depressive Disorder and Posttraumatic Stress Disorder." Journal of Personal Medicine. Published February 06, 2022. doi:10.3390/jpm12020224

K.A. Grisanzio, A.N. Goldstein-Piekarski, M.Y. Wang, **A.P. Rashed Ahmed**, Z. Samara, L.M. Williams, "Clustering identifies symptom-brain-behavior subtypes that cut across mood, anxiety, and trauma disorders." JAMA Psychiatry. Published online December 03, 2017. doi:10.1001/jamapsychiatry.2017.3951

A.P. Rashed Ahmed, M.C. Browne, D.L. Flath, K.L. Gumerlock, T.K. Johnson, L. Lee, Z. Lentz, T.H. Rendahl, H. Shi, H.H. Slepicka, Y. Sun, T.A. Wallace, D. Zhu, "Automated Controls for the Hard X-Ray Split & Delay System at the Linac Coherent Light Source." ICALEPCS 2017, Barcelona (2017)

A.H. Rose, B.M. Wirth, R.E. Hatem, **A.P. Rashed Ahmed**, M.J. Burns, M.J. Naughton, and K. Kempa, "Nanoscope based on nanowaveguides." Optics Express 22, 5228-5233 (2014)

Research Experience

Graduate Student **Brown University**
Serre, Nassar Labs, Providence, RI September 2018 - Present
Computational and Behavioral Mechanisms of In-Context Learning

- Developed novel experimental paradigm isolating temporal vs event-contingent in-context learning
- Revealed sparse LSTM (2 units) circuits sufficient for near-optimal in-context learning
- Manuscript in prep

Research Assistant **Stanford University**
PanLab, Stanford, CA December 2017
Unsupervised Learning Identifies Symptom Biotypes

- Applied unsupervised learning techniques to *BrainNet* data to identify symptom-brain-behavior subtypes spanning neurophysiological, neurocognitive, and symptom domains
- Symptom biotypes explain more variance on cognitive, EEG, and functioning variables than the Diagnostic and Statistical Manual of Mental Disorders (DSM), and can be used to supplement treatment intervention
- Results published in *JAMA Psychiatry*

Research Intern **SLAC National Accelerator Lab**
XCS Instrument, Menlo Park, CA October 2017
Automated Controls for Hard X-Ray Split Delay

- Designed and implemented software architecture for controls automation of the hard x-ray split and delay system at the x-ray correlation spectroscopy (XCS) instrument at the Linac Coherent Light Source (LCLS)
- Presented poster and paper published at International Conference on Accelerator and Large Experimental Control Systems (ICALEPCS) 2017

Teaching

Jan 2024 – Present
Mentor, *InspiritAI*, (Remote)
Course: Independent Projects

- Mentor students one-on-one on a research question in AI

Jan 2022 – May 2022
Teaching Assistant, *Neuroscience*, (Brown University, Providence, RI)
Course: Advanced Systems Neuroscience (Grad)

- Led weekly sections and office hours

May 2021 – Dec 2022
Instructor, *InspiritAI*, (Remote)
Course: Deep Learning

- Taught introductory machine learning and deep learning to high school students

Sep 2019 – Jan 2020
Teaching Assistant, *Neuroscience*, (Brown University, Providence, RI)
Course: Neuropracticum - Computational Module

- Co-taught two-week course at the Marine Biological Laboratory

Sep 2019 – Jan 2020
Teaching Assistant, *Neuroscience*, (Brown University, Providence, RI)
Course: Neural Systems

- Led weekly sections and office hours

Professional Development

Certificates

Sep 2024 – Dec 2024, **Sheridan Teaching Seminar**
Brown University, Sheridan Center for Teaching and Learning (Providence, RI)

Oct 2024 – Nov 2024, **Mentoring Up**
Brown University, Division of Biology and Medicine (Providence, RI)

Awards and Fellowships

Date: NIH 3R01MH126971-03S
$204,261, NIMH

Acknowledgments

Every Ph.D. is said to be a journey unique to the individual, and this one is no exception. In truth, when I look back at the last seven years, I can see that my journey stretches back much further than when I first arrived at Brown. And for that, many more people than those I've met during my time here deserve acknowledgment, but to start I'd like to thank the advisors I've had throughout my time at Brown, Matthew Nassar and Thomas Serre. I know my PhD has not been the easiest to supervise, but it was through your mentorship, support, and patience that I was able to complete the program, and for that I am so grateful. For my dissertation work specifically, I want to thank Matthew Nassar for taking what was an initial curiosity and running with it for the last few years. I also want to thank Steven Liao for being a pleasure to work with while building the system that actually ran the human experiments. I'd like to also thank my committee, David Sheinberg, Michael Frank, and Joshua Gold for your support throughout the program and the defense.

Over the years I've been a part of or affiliated with the Serre, Nassar, and Frank labs and I want to thank everyone who was there for their help and support throughout my time. In the Serre lab, I particularly want to thank Minju Yung who took on some of my more out-there ideas with event cognition and keeping me grounded when experiments weren't going as expected. In the Nassar lab, I'd like to thank Niloufar Razmi for always being willing to go for a walk to talk about things unrelated to labwork. In the Frank lab, I want to thank Alana Jaskir, Guillaume Pagnier, and Aneri Soni not for any particular

event, but all the time we spent outside of lab ranging from climbing, to dancing, to having a night out in the city.

Looking back at my time at Brown, it's hard not to bring up how much my cohort set the tone for the whole experience. I want to thank all the members of the 2018 neuroscience program for making our first year and beyond such a fun time. I want to shoutout Tariq Brown and Mor Alkaslasi for all the punches we threw, Julia Licholai for all the art we shared/hosted, Max Seppo for all the problems we sent, Pablo Iturralde for all the ethereum we mined, Isabella Penido for all the wild conversations, and Audrey Madeiros for the late night jaunts back from the GCB. Beyond my cohort, I'd like to thank all the members of the neuroscience department and Brown community who have been a part of the journey. In particular, Brian Kim for being the best neuro bestie, Scott Susi for turning hyperfixations into some of the best conversations I've had, Olivia McKissick and Remy Mier for being the funnest duo to run Brown with, and Nina Friedman for being one of the best friends I've had.

Outside of Brown, I had the privilege of discovering a lifelong passion in salsa dancing, which has changed so much of how I view the world and what I want with my life. For that I want to thank the broader salsa community for being so fun and welcoming. I want to also thank Carlos Gonzalez and Rieku Rivera for fostering such a community and creating the space for me to grow as a dancer. Beyond the dancing itself, I've had the honor of building a found-family in a way that I didn't know I wanted, and for that I want to thank Jessica Pierre, Jimmy Phann, Alex Madrid, Hellen Grejada, and Brieana Hanzel.

Besides my academic advisors, friends, and dance partners, one of the most important people for directly making the Ph.D. happen was Jamall Pollock. We met a few months into the program, but I had no idea how pivotal our time would be to not just my ability to succeed in the Ph.D., but to grow as a person. Words do not capture it, but I can only hope that I can have such a positive impact on those around me, the way you have impacted me.

Throughout the journey, many people have seen my progression through the program, but only one person saw the mad dash to the finish line up close. I'd like to thank my partner Emily Salois for being the most supportive, loving, and patient partner as I wrapped up this arc of my life. My hope is we can both look back and cherish the time we had together, especially all of our wonderful dances.

A few years into the program I had the privilege of being a recipient of the cat distribution system. And what I got was the most rambunctious, athletic, vegetarian queen to walk this side of the Mississippi. Mila, our time together were some of the happiest and joyous years of my life, and I wish there was a way to tell you how much you meant to me. I always imagined we'd get to celebrate together, but I'm grateful I got to love you in the time we had.

If there was a single person in my life that I'd blame for having to write this book on, it would be my dad, Rashed Ahmed Ali. It's entirely your fault for showing me how much fun can come from following your curiosities, and how much there is to be curious about in the world. Thank you for being the first to teach me how to look. This dissertation and Ph.D. is a fulfillment of both our dreams, and I hope it can be for you what you imagined it would be. If my dad is to blame for having write the book, my mom, Chuchi Pablico, is responsible for me actually writing it. It's entirely your fault that I would develop the grit and resilience to handle all the challenges life would throw my way. Thank you for loving me so fiercely and inspiring me to be the best man that I can be.

And finally, if my parents were the reasons I was on this journey, my little sister, Zeinab Rashed Ahmed, is the reason I was able to to maintain my soul throughout it all. Lil sis, you've been the single stable thread through my life, and I cannot express how much I've grown from being your older brother. Thank you so much for always being on my team and I hope I can continue to live up to the title of Kuya.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

"The Only Constant in Life Is Change." - Heraclitus

Every intelligent system, past, or present, faces the problem of an ever-changing world. And it's not just noticing that something has changed, but understanding the effects of that change that allows one to respond flexibly. This capacity to adapt behavior based on a changing environment—what we call in-context learning—requires keeping track of statistical patterns that evolve at multiple timescales. And despite its ubiquity, we lack a full understanding of how it works, its mechanisms, or limitations. This dissertation tries to tackle a portion of this problem and investigates the computational and behavioral mechanisms that enable in-context learning of multiple temporal statistics.

1.1 Background

1.1.1 In-Context Learning Foundations

Traditional machine learning systems typically require thousands of training examples to learn new patterns, whereas humans and large language models (LLM) quickly adapt their behavior to new observations. This ability, where an agent adapts behavior to new

statistical patterns within a single context, was termed "in-context learning" when it was observed in large language models (Brown et al., 2020). Unlike standard learning-paradigms which requires extensive training of parameters via backpropagation or synaptic plasticity, in-context learning allows an agent to adapt their behavior immediately. LLMs have been shown to do this by being trained to condition their responses on the input prompt, allowing them to "learn" new tasks from samples without fine-tuning their weights. Biological systems also display these abilities when animals adapt their foraging strategies in new environments. Understanding the principles that enable this form of rapid adaptation is critical to explaining intelligent behavior.

This capacity for in-context learning is seen routinely in human behavior, where people quickly extract and apply new patterns from the data. Humans can rapidly infer hidden rules and infer task structures while given minimal observations and still adjust their behavior within a handful of trials (Behrens et al., 2007; Collins and Frank, 2013). These adaptations occurring across multiple cognitive systems ranging from perceptual predictions to abstract reasoning suggest that the mechanisms for extracting and integrating statistical information are conserved across systems. Further, when tracking changing statistics within experimental sessions, task performance often approximates optimal Bayesian inference (Nassar et al., 2010), yet the processes that enable this flexible behavior remain largely unknown. Recent work comparing humans and LLMs on identical tasks shows both similarities and differences in their in-context learning capabilities (Binz and Schulz, 2023). However, these behavioral comparisons cannot directly investigate how either system implements rapid statistical inference. Addressing these gaps requires new paradigms that isolate distinct forms of in-context learning to directly examine internal mechanisms.

This dissertation tackles these challenges by investigating how intelligent systems track multiple statistical patterns, like inferring whether an occluded traffic light has changed based on both elapsed time and whether surrounding cars begin moving. Through

parallel investigation of human participants and recurrent neural networks, combined with mechanistic analysis via targeted neural interventions, we reveal some of the minimal circuitry required to solve the challenge of in-context learning.

1.1.2 Neural Implementation

Recurrent neural networks have have served a key role in modeling cognitive processes that unfold over time (Lipton et al., 2015). Their ability to recognize and learn temporal dependencies and relationships within sequential data makes them well-suited for modeling language processing or behavior (Elsayed et al., 2022; Li et al., 2020). Traditional approaches can often fall short in capturing the context-dependent nature of cognition or behavior, whereas RNNs offer a powerful framework for representing the internal states and transitions that underlie complex cognitive functions (Barak, 2017). Despite their capabilities, standard RNNs struggle with learning long-range dependencies because the gradient either vanishes or explodes as the information moves down the sequence.

LSTMs emerged as a specialized type of RNN designed to address the vanishing gradient problem while improving the ability to learn long range dependencies (Hochreiter and Schmidhuber, 1997; Yang et al., 2019). They accomplish this through their specialized gating mechanisms, which allow them to selectively retain or discard information from previous time steps. These mechanisms comprise the input, forget, and output gates, which control the flow of information into and out of the memory cell, allowing the LSTM to learn long-term dependencies more effectively. Because of these abilities, LSTMs have been increasingly used as tools for capturing the dynamic and sequential nature of human behavior (Karpathy et al., 2015). In this dissertation, we add to this tradition and investigate the representations learned by LSTMs on predictive inference.

1.2 Research Aim and Questions

1.2.1 Research Aim

This research aims to investigate the computational and behavioral mechanisms underlying in-context learning of multiple temporal statistics, examining how humans and recurrent neural networks extract and track statistical regularities from changing and uncertain environments.

1.2.2 Research Questions

Two primary research questions structure the investigation carried out in this dissertation, going from behavioral validation and comparative analysis to mechanistic understanding.

RQ1: How do humans and recurrent neural networks perform in-context learning of multiple statistics compared to optimal inference?

This question investigates whether humans and recurrent neural networks can track time-based probabilities and event-based probabilities in the same trials, along with how they compare to the Bayesian optimal inference. Behavioral signatures of successful learning include: (1) predictions that scale with elapsed time when only temporal information is available, and (2) predictions that respond to discrete events when contingent relationships exist.

RQ2:What neural mechanisms enable successful in-context learning of multiple statistics?

This question investigates what neural representations emerge in recurrent networks that successfully learn both parameters and how these representations causally influence behavioral predictions.

1.3 Structural Outline

1.3.1 Chapter 1: Introduction

Establishes the research questions and frameworks that guide this investigation.

1.3.2 Chapter 2: In-Context Learning of Latent Structure

Presents the Bouncing Ball task, the primary behavioral study of this investigation, and the ideal Bayesian observer (IBO) that establishes the predictions for successful in-context learning. This chapter analyzes behavioral data collected from running the Bouncing Ball task on humans and recurrent neural networks, establishes the Bouncing Ball task as a platform for studying in-context learning, characterizes the performance profiles compared to the IBO, and then proposes the counting bottleneck as an explanation for selective learning.

1.3.3 Chapter 3: Computational Mechanisms of In-Context Learning

Investigates the mechanisms that enable the LSTM to reach near-optimal performance on the Bouncing Ball task. The chapter employs three main approaches to understand neural representations: (1) Identifies neural units that encode the task-relevant parameters. (2) Characterizes their tuning profiles. (3) Perform targeted interventions to establish a functional relationship with behavior.

1.3.4 Chapter 4: Discussion and Future Directions

Synthesizes findings to propose the counting bottleneck as a hypothesis for the behavioral outcomes of both humans and recurrent neural networks. This chapter discusses the limitations of this investigation and future directions.

References

Barak, O. (2017). Recurrent neural networks as versatile tools of neuroscience research. *Current Opinion in Neurobiology*, 46:1–6.

Behrens, T. E. J., Woolrich, M. W., Walton, M. E., and Rushworth, M. F. S. (2007). Learning the value of information in an uncertain world. *Nature Neuroscience*, 10(9):1214–1221. Publisher: Nature Publishing Group.

Binz, M. and Schulz, E. (2023). Using cognitive psychology to understand GPT-3. *Proceedings of the National Academy of Sciences*, 120(6):e2218523120. Publisher: Proceedings of the National Academy of Sciences.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language Models are Few-Shot Learners. arXiv:2005.14165 [cs].

Collins, A. G. E. and Frank, M. J. (2013). Cognitive control over learning: Creating, clustering, and generalizing task-set structure. *Psychological Review*, 120(1):190–229. Place: US Publisher: American Psychological Association.

Elsayed, N., ElSayed, Z., and Maida, A. S. (2022). LiteLSTM Architecture for Deep Recurrent Neural Networks. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1304–1308. ISSN: 2158-1525.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Karpathy, A., Johnson, J., and Fei-Fei, L. (2015). Visualizing and Understanding Recurrent Networks. arXiv:1506.02078 [cs].

Li, Z., Zhang, L., Lei, C., Chen, X., Gao, J., and Gao, J. (2020). Attention with Long-Term Interval-Based Deep Sequential Learning for Recommendation. *Complexity*, 2020(1):6136095. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2020/6136095.

Lipton, Z. C., Berkowitz, J., and Elkan, C. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning. arXiv:1506.00019 [cs].

Nassar, M. R., Wilson, R. C., Heasly, B., and Gold, J. I. (2010). An Approximately Bayesian Delta-Rule Model Explains the Dynamics of Belief Updating in a Changing Environment. *Journal of Neuroscience*, 30(37):12366–12378. Publisher: Society for Neuroscience Section: Articles.

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

CHAPTER 2

In-Context Learning of Latent Structure

2.1 Introduction

To navigate and uncertain world, intelligent systems must continuously infer what is happening in their environment in order to predict what might happen next. Consider trying to cross a busy street - you actively track the hidden state of the traffic light (which may not be visible) and whether the approaching cars are slowing down, before predicting that it is safe to cross. These inferences require adapting your behavior depending on the stream of sensory information, where some portions of it change spontaneously (traffic light cycling) while others are triggered by specific events (cars start moving when lights change).

This challenge of detecting when changes occur and inferring how the latent states changed as well, pervades flexible and intelligent behavior. The brain solves this by tracking at least two types of statistics: the probability of changes that happen randomly (hazard rate) and the probability of changes that happen because of a trigger (contingency). And while the effects of either of may produce identical sensory input, such as a car being

stopped due to a red light or a traffic jam, these statistics need different strategies for inference. Tracking random changes requires continuously monitoring your input stream, so that when a change does occur you can estimate how often they happen. Whereas tracking event-triggered changes requires you only consider your input stream when the trigger is detected, so that you can estimate the relationship between the change and trigger.

Despite the ubiquity of this challenge, we don't have a full understanding of how biological and artificial systems navigate it. This is further compounded when both types of statistics have to be learned simultaneously from limited observations. We introduce the Bouncing Ball task to isolate these two forms of statistical learning within a single paradigm where both humans and neural networks can be directly compared. Observers watch a colored ball bounce around a screen, where its color is occluded as soon as it passes behind a gray vertical strip. The ball's color changes according to two latent parameters that vary trial-by-trial: the hazard rate which controls spontaneous changes and contingency which links color changes to velocity changes. Because these parameters change between trials, observers cannot rely on prior trials to inform their guess of the current statistics and must rely on in-context learning for each video. Then participants report the color of the occluded ball along with their confidence in that choice, allowing us to probe their understanding of the latent variables. And since the underlying statistics for both random changes and contingent changes vary on each trial, we can examine how each of these contextual factors influence the participants' choices, revealing whether and how much they use the patterns of each video to change their behavior.

Optimal Learning

To understand how context should influence the choices made in our task, we developed an Ideal Bayesian observer (IBO) model. The IBO optimally estimates the statistical context of each video by updating probability distributions over both the hazard rate

and contingency as evidence accumulates through observing color changes. Then it uses these estimates to infer the color of the ball whenever its occluded. And by examining its behavior, we can identify key behavioral signatures of successful in-context learning for both of the tracked statistics. Sensitivity to the hazard rate should manifest as changes in the color inferences as the ball spends more time occluded. Additionally, how much it changes with time will depend on the underlying hazard rate. Sensitivity to contingency should manifest as changes to the color inferences depending on how many times the ball changed color on each bounce. Crucially, since the IBO uses the same accumulation machinery for both parameters, differences in performance between these parameters must arise from what gets counted rather how the counts are processed.

Humans and Models

While the Ideal Bayesian Observer describes how context *should* be used for inference, it cannot tell us how context *does* get used in biological and artificial systems. To address this gap, we conducted investigations into both human participants and recurrent neural networks when performing the same task.

For human participants, we first examined whether they exhibited the behavioral signatures predicted by the IBO, specifically the time-dependent sensitivity to hazard rate and the bounce-dependent sensitivity to contingency. Beyond these core predictions, we quantified the ways in which their inferences deviated from optimal, to understand what potential constraints or strategies are being used.

To explore whether our observations reflect human-specific abilities or general computational principles, we also trained recurrent neural networks on the task. We compared two architectures with different memory capacities, a vanilla recurrent neural network (RNN), which uses simple recurrent connections, and a Long Short-Term Memory network (LSTM), which has specialized gating mechanisms that allow for complex memory operations. These models serve two purposes: First, they reveal whether artificial systems

respond similarly to biological ones when faced with similar challenges. Second, they provide a system where we can directly examine the machinery required for successful in-context learning.

Through these two investigations, we identify two hallmarks of contextual inference that emerges in biological and artificial systems while also exploring how their systematic differences point to their fundamental constraints.

We first detail the task's generative model and the IBO's normative solution, then describe our experimental procedures for both humans and neural networks. The results examine each agent's sensitivity to both parameters before revealing the behavioral strategies that emerge. The discussion frames these findings through the lens of counting limitations, explaining why identical computational machinery produces asymmetric learning.

2.2 Methods

2.2.1 The Bouncing Ball Task

The Bouncing Ball task presents a color-changing ball that moves elastically on a screen, with its state characterized by its position, velocity vector, and color (fig. 2.1A). The color changes according to a fixed sequence: Red $\rightarrow$ Green $\rightarrow$ Blue. Participants can fully observe the ball's position and velocity, but the color is occluded when the ball passes behind a vertical strip in the middle of the screen, which we refer to as the gray zone. The ball's color and velocity can change randomly on their own or together. For clarity, velocity changes in the Bouncing Ball task refer only to changes in direction and not magnitude. Put together, the task comprises four possible state change conditions that govern the dynamics of the ball's color and velocity:

1. No color change and no velocity change

2. Color change but no velocity change
3. No color change, but velocity change
4. Color change and velocity change

which are shown visually in Figure 2.1B. These state changes are controlled by two latent parameters:

1. **Hazard Rate:** This parameter controls the probability of a random color change at eligible time steps. It can take one of two values – a low value of ∼0.5% chance of color change per time step or a high value of ∼3.5% chance of color change per time step.
2. **Contingency:** This parameter controls the probability of a color change given that the velocity has changed, regardless of whether it is a wall or a random bounce. It can take one of three values – low (around 5.0% chance of color change on velocity change), medium (50% chance of color change on velocity change), or high (around 95% chance of color change on velocity change).

These specific values for the hazard rate were selected to balance several factors. They yield distributions of color changes for each video that are both perceptually discernible to participants while not presenting too many. Additionally, these values produce color-change probability trajectories that remain fairly linear for the duration of time that the ball passes through the gray zone. For contingency, the values were chosen to span the full range of possible contingent color changes while not being perfectly discernible.

Each trial of the Bouncing Ball task consists of a single video of a ball moving around the screen. In each trial, the values for the hazard rate and contingency parameters are randomly selected and must be learned in-context during the course of a trial. The video length is sampled from an exponential distribution. All experiment trials end with the ball behind the gray zone. Based on the observed statistics from the video, the participant's goal is to choose the correct color of the ball at the end of the video. Participants also

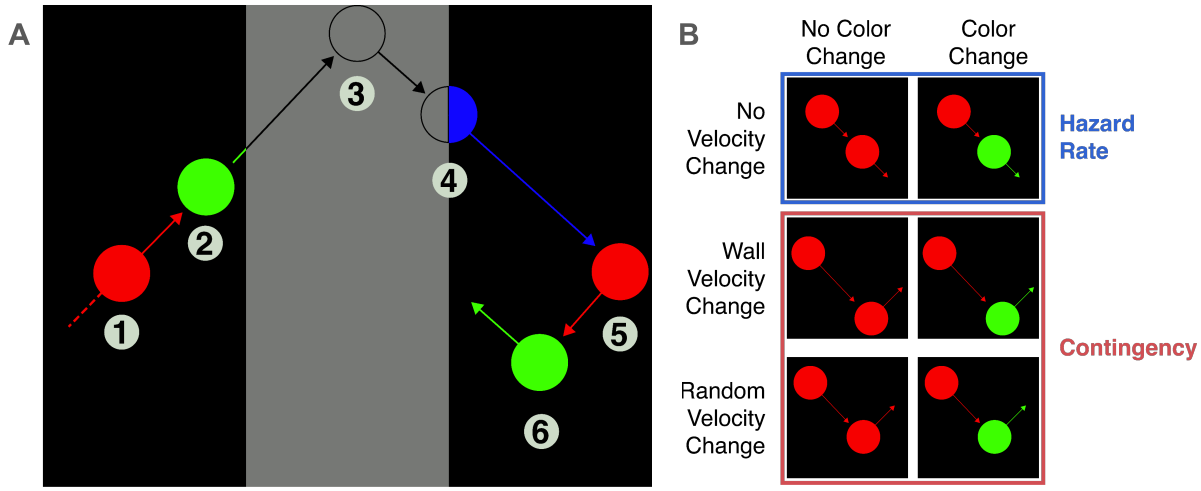


Figure 2.1: Bouncing Ball Sequence and Task Parameters. **(A)** Portion of a Bouncing Ball task sequence with the various change conditions. The ball position is viewable throughout the screen, however, the color is hidden in the gray zone. The ball is initialized with a random sampling of position, velocity, and color (sampled from red, green, or blue). (1) The ball is in a visible segment and red. (2) The ball color undergoes a random color change from red to green with no velocity change. (3) Ball bounces off the wall in the gray zone, potentially triggering a contingent color change. (4) Ball exits the gray zone, revealing it transitioned colors from green to blue. (5) Ball bounces off the wall, triggering a contingent color change from blue to red. (6) Ball randomly bounces, triggering a contingent color change from red to green. **(B)** Color and velocity change matrix, showing which statistic governs which combinations of changes. The hazard rate controls the probability of a color change when there are no velocity changes. The contingency controls the probability of a color change when there is a velocity change, applying equally to wall bounces and random bounces.

select a confidence rating on a slider. A full trial consists of the video, followed by the color choice, followed by the confidence input. In principle, participants who track the hazard rate and contingency parameters associated with each video could infer how the ball's color has likely evolved during its passage through the gray zone. Such inference would allow them to adjust their responses based on the probability that a change occurred while the ball was occluded, potentially improving accuracy beyond chance levels.

There are two main types of experimental trials: straight path trials and bounce path trials, which differ only in how the trials end. In the straight path trials, the ball enters the gray zone and continues along a straight trajectory until the trial ends (fig. 2.2A). The duration within the gray zone varies across three fixed intervals (short, medium, or long); however, the ball's vertical position at trial termination depends on its entry point and trajectory. Whereas in the bounce path trials, on the final entry into the gray zone, the ball bounces off the top or bottom walls before the trial ends (fig. 2.2D).

There is also an additional attention-check trial where the ball ends in a visible portion of the screen; however, these trials were not used in the results. In the main experiment, trials are randomly grouped into 10 blocks, where each block has a variable number of videos and variable video lengths. Feedback is provided after each block, and a 5-minute break follows before the next block begins.

2.2.2 Generative Process

Each Bouncing Ball video is generated through an iterative stochastic process. The complete state of the system evolves over a series of discrete time steps, $t \in 0, 1, ..., T$, where T is the total duration of the trial. The dynamics of the Bouncing Ball are governed by a set of fixed parameters selected for each trial, as well as a set of time-dependent state variables that evolve according to probabilistic rules.

Latent State Representation

The complete state $S^{(t)}$ at timestep t consists of a base set of time-dependent variables $X^{(t)}, V^{(t)}, C^{(t)}, dV^{(t)}, dC^{(t)}$ and time-independent parameters $P_{hz}, P_{cont}, P_{rvc}$. The time-dependent variables are defined as follows:

- $X^{(t)} = (x^{(t)}, y^{(t)})$ represents the ball's position coordinates
- $V^{(t)} = (v_x, t, v_y, t)$ represents the ball's velocity components
- $c^{(t)} \in \{0, 1, 2\}$ represents the ball's color, where 0, 1, and 2 correspond to red, green, and blue, respectively, following the fixed progression sequence red $\rightarrow$ green $\rightarrow$ blue
- $dV^{(t)}$ is a boolean flag indicating whether a velocity change occurs at timestep t
- $dC^{(t)}$ is a boolean flag indicating whether a color change occurs at timestep t

And the time-independent parameters are defined as follows:

- P_{hz} is the hazard rate, defining the probability of a color change at time steps where $dV^{(t)} = 0$
- P_{cont} is the contingency parameter, defining the probability of a color change at time steps where $dV^{(t)} = 1$
- P_{rvc} is the random velocity change probability, controlling spontaneous velocity changes when no wall collision occurs

In addition to these variables, we use several rate-limiting variables to ensure that two random changes in color or a velocity change followed by a random change are perceived by human participants as separate events:

- τ_{dc} sets the minimum number of time steps after a color or velocity change before another random color change can occur

- τ_{dc} sets the minimum number of time steps after a color or velocity change before another random velocity change can occur

This approach is consistent with work demonstrating that the human cognitive system continuously segments ongoing experience into discrete 'events,' with perceived event boundaries influencing how changes are processed and understood (Swallow et al., 2009). Importantly, when these variables are set to zero, color changes and random velocity changes can occur at any eligible timestep.

Sequence Generation

The sequence generation process begins by sampling initial conditions to obtain x_0, v_0, and c_0, with velocity and color change flags initialized as $dV_0 = dC_0 = 0$. Additionally, containers for storing limited time steps, $\mathbf{R}_{dc}^{(t:T)} = \mathbf{0}^{(t:T)}$ and $\mathbf{R}_{dv}^{(t:T)} = \mathbf{0}^{(t:T)}$ are initialized to be zero for all time steps. For each subsequent timestep $t > 0$, the following iterative procedure is applied:

Velocity Change Determination: The velocity change flag $dV^{(t)}$ is computed by first determining the wall collision flag $w^{(t)}$:

$$w^{(t)} = \begin{cases} 1, & \text{if } x^{(t)} > L - r \text{ or } x^{(t)} < r \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

where L represents the dimensions of the Bouncing Ball environment and r is the ball radius, both fixed across all videos. The velocity change flag is then determined as:

$$dV^{(t)} = \begin{cases} w^{(t)}, & \text{if } \max(w^{(t)}) = 1 \\ \text{Bernoulli}(P_{\text{rvc}}), & \text{if } r_{dv}^{(t-1)} = 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

where Bernoulli(P_{rvc}) represents a Bernoulli process that returns 1 for one velocity com-

ponent and 0 for the other (a velocity change in either v_x or v_y, not both), or 0 for both components, and $r_{dv}^{(t-1)}$ is the value of the velocity change rate limiter at time step t.

Color Change Determination: The probability of a color change at timestep t is determined by the velocity change status:

$$P_{dC}^{(t)} = \begin{cases} 0, & \text{if } r_{dc}^{(t-1)} = 1 \\ P_{hz}, & \text{if } dV^{(t)} = 0 \\ P_{cont}, & \text{if } dV^{(t)} = 1 \end{cases} \tag{2.3}$$

Where $r_{dc}^{(t)}$ is the value of the velocity change rate limiter at time step $t-1$. The color change flag is then sampled as:

$$dC^{(t)} \sim \text{Bernoulli}(P_{dC}^{(t)}) \tag{2.4}$$

State Updates: With the boolean change flags defined, the velocity, position, and color are updated according to:

$$v^{(t)} = \begin{cases} v^{(t-1)} & \text{if } dV^{(t)} = 0 \\ -v^{(t-1)} & \text{if } dV^{(t)} = 1 \end{cases} \tag{2.5}$$

$$x^{(t)} = x^{(t-1)} + v^{(t)} \cdot dt \tag{2.6}$$

$$c^{(t)} = \begin{cases} c^{(t-1)} & \text{if } dC^{(t)} = 0 \\ (c^{(t-1)} + 1) \bmod 3 & \text{if } dC^{(t)} = 1 \end{cases} \tag{2.7}$$

where dt is a predefined step size.

Rate Limiting: We now update our rate-limiting vectors according to whether there

were changes at this timestep:

$$\mathbf{R}_{dv}^{(t:\tau_{dv})} = 1 \quad \text{if } dV^{(t)} = 1 \text{ or } dC^{(t)} = 1 \tag{2.8}$$

$$\mathbf{R}_{dc}^{(t:\tau_{dc})} = 1 \quad \text{if } dV^{(t)} = 1 \text{ or } dC^{(t)} = 1 \tag{2.9}$$

This iterative process continues until the desired number of time steps T is reached.

Gray Zone Masking: The final observable Bouncing Ball sequence is defined as $O^{(t:T)} = (o^{(t)}, \ldots, o^{T})$, where each observation $o^{(t)} = (x^{(t)}, \tilde{c}^{(t)})$ consists of the position and the potentially occluded color:

$$\tilde{c}^{(t)} = \begin{cases} -1, & \text{if } G_s < x^{(t)} < G_e \\ c^{(t)}, & \text{otherwise} \end{cases} \tag{2.10}$$

Where -1 denotes the color being occluded behind the gray zone, and G_s and G_e denote the starting and ending positions of the gray zone, respectively.

2.2.3 Generating Participant Datasets

Parameter Control

The process outlined above is used to generate a single Bouncing Ball sequence, where the position, velocity, and color unfold in a non-deterministic manner. However, to use the sequences for investigating human behavior, we needed to be able to create sequences with highly controlled final states. To accomplish this, we set the desired final state as the initial condition for the task, generate a full sequence, and reverse the order of the resulting states. This operation preserves the overall generative structure of the task if followed by the simple indexing shifts below:

$$\hat{C}^{(t:T)} = C[0] + C[0{:}T-1] \tag{2.11}$$

While a single frame shift is not enough time to notice for human participants, this correction is added so that models (which can resolve single frame changes) observe contingent color changes on the same time step as a velocity change.

Building The Trial Types

We use the sequence-reversal protocol to generate batches of sequences, which are then organized into task videos. To this end, all final states are controlled for final color, whereas position and velocity are controlled in a trial-specific manner. The straight path trial type is defined as sequences where the final state ends in the gray zone at a location where it cannot bounce off a wall. Since the gray zone occupies a vertical strip of the stage, we set the trials to end after a fixed duration in the gray zone, which will either be short, medium, or long, where the ball can be at any position vertically. These locations correspond to the ball spending 1, 11, and 22 time steps in the gray zone. We control for the side the ball enters the gray zone (left and right), and its y-velocity (positive and negative). Additionally, it is imposed that there cannot be any random velocity changes in the final gray zone traversal, so the change statistics are strictly related to the hazard rate. For the bounce path trials, we set the final location of the ball to be one of four positions corresponding to the corners of the gray zone. These positions were selected to minimize the amount of time spent in the gray zone (10 time steps), allow the ball to bounce off the wall after the same amount of time in the gray zone (5 time steps), and provide participants with adequate time to perceive the bounce. For the attention check trials, we set the final location of the ball to be in one of the locations outside of the gray zone.

Video Parameters

In addition to the final states, there are several parameters defined for the task videos as a whole, such as the statistics of interest. To ensure the amount of time it takes to traverse the gray zone is consistent across trial types, we set the x-velocity to be fixed

across all trials, and vary the y-velocity between two values, one smaller and one larger in magnitude than the x-velocity, giving unique trajectories for each video. We varied the length of a trial by setting a minimum amount of time for each trial (7.5 seconds) and then sampling from an exponential distribution so that the length of a trial is not predictable. The maximum length is set to be times the length of the minimum (22.5 seconds).

Label Calibration

Setting the rate-limiting variables τ_{dv} or τ_{dc} to be nonzero not only prevents consecutive changes from occurring, but also reduces the total number of changes per video. This has the effect of changing the hazard rate or random velocity probability from a constant to a time-dependent value, where it takes on its specified value when there is no rate limiting but becomes zero when there is, lowering the observed hazard rate. Consequently, participants and models will appear to be underestimating the hazard rate when investigating their behavior, since we are collecting their responses to the local region of the stage where the changes reflect the raw hazard rate. To correct for this, after generating a set of task videos, we first calculate the effective hazard rates , $\tilde{P}_{hz}$, and modify the underlying hazard rate during only the final gray zone traversal to be $\tilde{P}_{hz}$ instead of P_{hz}. Here we define $\tilde{P}_{hz}$ as the ratio of the number of random color changes in the task videos and the number of eligible time steps, leading up to the final gray zone traversal:

$$\tilde{P}_{hz} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\sum_{t=1}^{T_i-G_i} dC_i^{(t)}}{T_i - \sum_{t=1}^{T_i-G_i} dV_i^{(t)}}\right) \tag{2.12}$$

Where N is the number of videos in the task videos, $\mathbf{T}$ is the length of each video in time steps, $\mathbf{G}$ is the number of time steps for the final traverse gray zone traversal per video (which varies for the straight path trials), and $\mathbf{dC}$ and $\mathbf{dV}$ are bounce and velocity flag matrices for each time step in each video. Importantly, since the results of the final color changes are never revealed to the participants or models, this change does not have any

effect on the observed statistics.

2.2.4 The Ideal Bayesian Observer

Having defined the full generative structure of the task, we can now derive the ideal Bayesian observer (IBO) as the normative model for the task. Given some observable state, $o^{(t)}$, at timestep t, the model must optimally infer the latent parameters P_{hz} and P_{cont}, to predict the ball's color, $c^{(t+1)}$, at the next timestep, $t+1$.

Parameter Representation

Given that the latent parameters P_{hz} and P_{cont} dictate binary events (color change or not) that must be represented in an online fashion, we modeled both with conjugate Beta priors:

$$P_{hz} \sim \text{Beta}(\alpha_{hz}, \beta_{hz}) \tag{2.13}$$

$$P_{cont} \sim \text{Beta}(\alpha_{cont}, \beta_{cont}) \tag{2.14}$$

Where each hyperparameter can be initialized individually, but we opted for a uniform prior, $\alpha_{hz} = \alpha_{cont} = \beta_{hz} = \beta_{cont} = \phi$, where ϕ represents the strength of the prior.

Online Inference on Visible Segments

At any timestep t, the IBO computes a belief vector $\mathbf{b}^{(t)} \in \mathbb{R}^3$ where:

$$[\mathbf{b}^{(t)}]_i = P(C^{(t+1)} = i | O^{(1:t)}) \text{for } i \in \{0, 1, 2\} \tag{2.15}$$

At every visible observation $o^{(t)}$, the IBO first updates the beta parameters based on any color or velocity changes from the previous timestep:

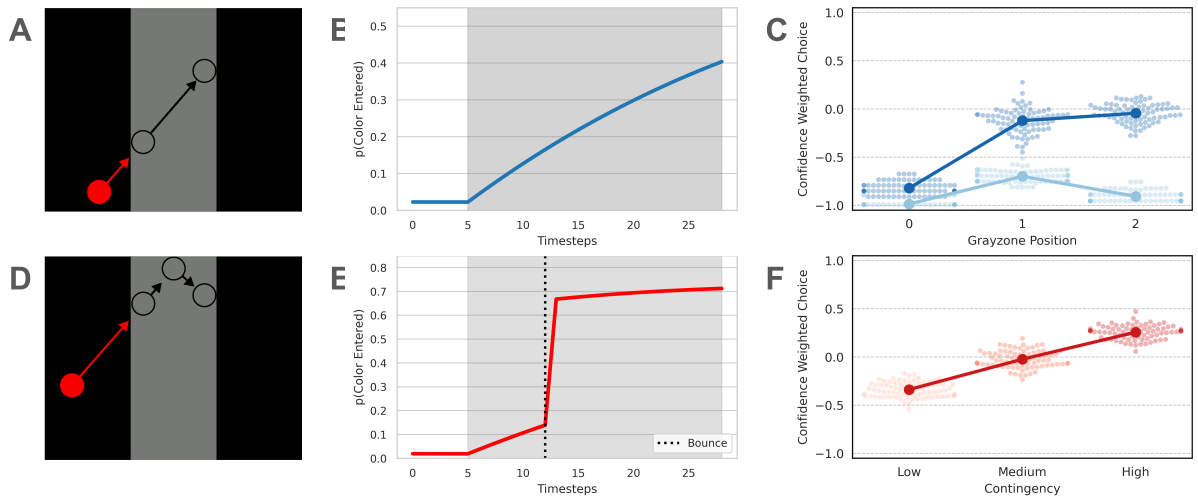


Figure 2.2: Trial Types and Ideal Bayesian Observer Predictions. The two experimental trial types - Straight Path and Wall Bounce, all end with the ball inside the gray zone. **(A)** In the straight path trials, the ball's ending trajectory travels through the gray zone without any velocity change. Trials end in one of three locations along the horizontal dimension, corresponding to spending to varying amounts of time in the gray zone, and can vary in the vertical dimension. **(B)** Ideal Bayesian observer color-change predictions through gray zone as the ball travels without a velocity change for a single trial. **(c)** Confidence weighted choice (CWC) for the straight path trials, split between the hazard rate conditions. Color choices are sampled using the color probabilities at one of three gray zone locations, combined to the generate the CWC, and plotted for each condition at each ending location. **(D)** In the wall bounce trials, the ball enters the gray zone and bounce off a wall before the trial ends. Trials end in one of four locations in the each of the corners of the gray zone which are controlled for the amount of time in the gray zone. **(E)** Ideal Bayesian observer color-change predictions through gray zone as the ball travels, collides with the wall, and changes velocity. **(F)** CWC for the bounce trials, split between the contingency conditions. Color choices at are sampled using the color probabilities at the ending location of the trial, combined to generate the CWC, and plotted for each condition.

$$\text{If } dV^{(t)} = 0: \quad \begin{cases} c^{(t)} \neq c^{(t-1)} \Rightarrow \alpha_{\text{hz}} \leftarrow \alpha_{\text{hz}} + 1 \\ c^{(t)} = c^{(t-1)} \Rightarrow \beta_{\text{hz}} \leftarrow \beta_{\text{hz}} + 1 \end{cases}$$
$$\text{If } dV^{(t)} = 1: \quad \begin{cases} c^{(t)} \neq c^{(t-1)} \Rightarrow \alpha_{\text{cont}} \leftarrow \alpha_{\text{cont}} + 1 \\ c^{(t)} = c^{(t-1)} \Rightarrow \beta_{\text{cont}} \leftarrow \beta_{\text{cont}} + 1 \end{cases} \tag{2.16}$$

With the new updated counts, compute the expectation for both latents:

$$\mathbb{E}[P_{hz}] = \frac{\alpha_{hz}}{\alpha_{hz} + \beta_{hz}}, \quad \mathbb{E}[P_{cont}] = \frac{\alpha_{cont}}{\alpha_{cont} + \beta_{cont}} \tag{2.17}$$

Define the transition probability:

$$P(dC^{(t)} = 1) = \begin{cases} \mathbb{E}[P_{hz}] & \text{if } dV^{(t)} = 0 \\ \mathbb{E}[P_{cont}] & \text{if } dV^{(t)} = 1 \end{cases} \tag{2.18}$$

Which we then use to construct the transition matrix $\mathbf{T}^{(t)}$:

$$\mathbf{T}^{(T-G)} = \begin{bmatrix} 1 - P(dC^{(t)} = 1) & P(dC^{(t)} = 1) & 0 \\ 0 & 1 - P(dC^{(t)} = 1) & P(dC^{(t)} = 1) \\ P(dC^{(t)} = 1) & 0 & 1 - P(dC^{(t)} = 1) \end{bmatrix} \tag{2.19}$$

And then update our beliefs:

$$\mathbf{b}^{(t)} = \mathbf{T}^{(t)\top}\mathbf{b}^{(t-1)} \tag{2.20}$$

Posterior Attribution for gray zone Segments

When ball exits gray zone with color c_{exit} having entered with some color, c_{entry}, we attribute changes to hazard or contingency according to the changes that occurred. First,

define the number velocity changes that occurred,

$$N_{dV} = \sum_{t=t_{\text{entry}}}^{t_{\text{exit}}} dV^{(t)} = 0 \tag{2.21}$$

Where t_{entry} and t_{exit} are the time steps that the ball entered and exited the gray zone respectively, and the number of color changes:

$$N_{dC} = (c_{\text{exit}} - c_{\text{entery}}) \bmod 3 \tag{2.22}$$

No Change Case When $c_{exit} = c_{entry}$, update β_{hz} parameter based on time steps elapsed. If no velocity changes ($N_{dV} = 0$), do a full attribution to hazard rate:

$$\alpha_{hz} \leftarrow \alpha_{hz} + 1 \tag{2.23}$$

If a single velocity change ($N_{dV} = 1$), define the posterior probability weighting:

$$w_{cont} = \frac{\mathbb{E}[P_{cont}]}{\mathbb{E}[P_{cont}] + \mathbb{E}[P_{hz}]} \tag{2.24}$$

Then perform the update:

$$\alpha_{cont} \leftarrow \alpha_{cont} w_{cont}, \quad \alpha_{hz} \leftarrow \alpha_{hz}(1 - w_{cont}) \tag{2.25}$$

The IBO as Theoretical Framework

The IBO serves two purposes in our experimental framework. First, it establishes the ceiling for performance given the inherent stochasticity to any particular set of task videos. Second, it generates testable predictions about how successful in-context learning should manifest behaviorally. These predictions become the organizing framework for our behavioral analyses in our results. To test the IBO's predictions about parameter learning

in human and model behavior, we defined metrics that directly capture sensitivity to hazard rate and contingency.

2.2.5 Task Metrics

Using the behavioral predictions laid out by the ideal Bayesian observer, we can define metrics for sensitivity to hazard rate and contingency using the color choices and confidence of observers.

Confidence Weighted Choice

We first start by observing that color choices made by participants can be described relative to the last visible color that entered the final gray zone traversal, which we define as $\mathbf{c}_{\text{entered}}$. We can then use the next color in the sequence, $\mathbf{c}_{\text{next}}$, and the third color in the sequence, $\mathbf{c}_{\text{third}}$, to transform all of the raw color choices into relative color choices, which we call $\hat{\mathbf{c}}$. To simplify, these relative choices can then be binarized to represent the *no-change*, choice, where $\hat{\mathbf{c}} = \mathbf{c}_{\text{entered}}$, or the *change* choice, where $\hat{\mathbf{c}} = \mathbf{c}_{\text{next}}$ or $\hat{\mathbf{c}} = \mathbf{c}_{\text{third}}$. We then define a new variable, $\hat{\mathbf{a}}$, which encodes the no change and change choices as -1 and 1 respectively:

$$\hat{\mathbf{a}} = \begin{cases} -1, & \text{if } \hat{\mathbf{c}} = \mathbf{c}_{\text{entered}} \\ 1, & \text{if } \hat{\mathbf{c}} = \mathbf{c}_{\text{next}} \text{ or } \hat{\mathbf{c}} = \mathbf{c}_{\text{third}} \end{cases} \tag{2.26}$$

On its own, $\hat{\mathbf{a}}$, can be used to investigate the proportional color change outcomes for any specific task condition. However, we expect that time in the gray zone should lead to changes in confidence, given a specific choice in addition to changes in choice outcomes. To investigate this effect, we designed the Bouncing Ball task to collect confidence measures from participants, and the computational models all return probabilities for each color choice, which can be used as a proxy for confidence. Therefore, given some confidence measure, $\hat{\kappa}$, we can define a combined measure of the color change outcomes and the

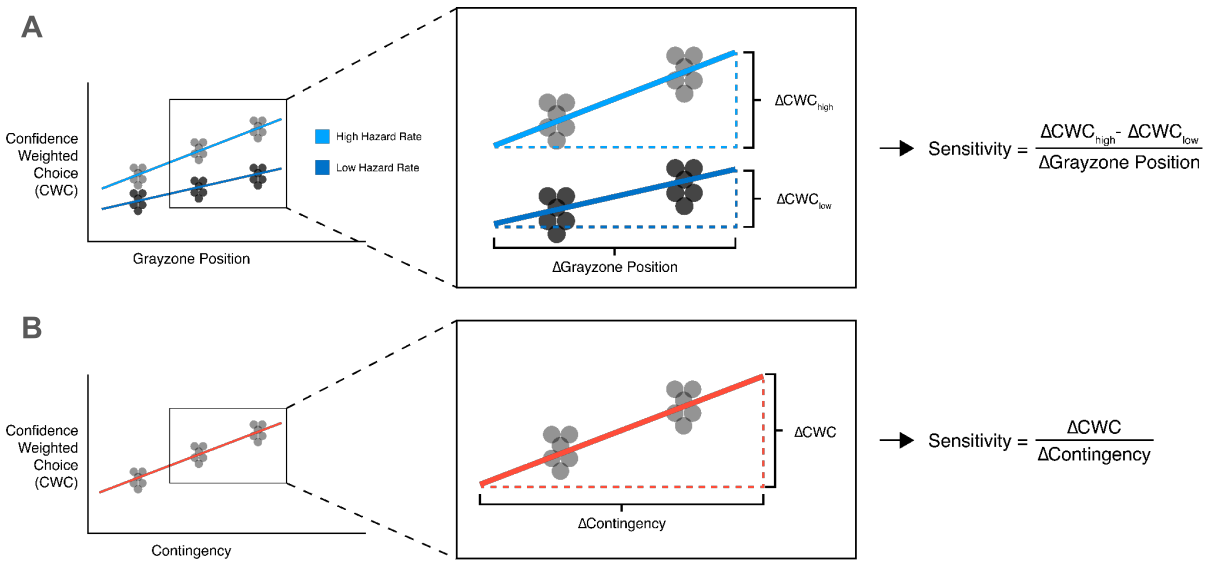


Figure 2.3: Defining Task Sensitivity. We derive two metrics for task sensitivity on each trial type from the confidence weighted choice (CWC). **(A)** Hazard Sensitivity - For each of the high and low hazard rate conditions, we fit a line to the CWC across the positions in the grayzone, and compute the sensitivity to hazard rate as the difference between the slopes. **(B)** Contingency Sensitivity - We fit a line through the CWC across the three contingency conditions and define sensitivity as the slope.

confidence on a specific trial, called the confidence weighted choice:

$$\mathbf{C\hat{W}C} = \hat{\mathbf{a}} \cdot \hat{\kappa} \tag{2.27}$$

Where a value of 1 denotes complete confidence that the color changed, -1 denotes full confidence in no color change, and 0 denotes no confidence in whichever choice that was made.

Hazard Rate Sensitivity

Using the $\mathbf{C\hat{W}C}$, we can better plot the behavioral differences between the low and high hazard rate conditions. Figure 2.2C shows the aggregated $\mathbf{C\hat{W}C}$ outcomes of the IBO on the straight path trials, split by the hazard rate, and plotted along the three end-positions in the gray zone. Similar to what was shown before with the raw color choices, the hazard rate affects the color predictions as a function of time in the gray zone, where a higher hazard rate results in an increased likelihood of a guessed color change. We can quantify this effect of the hazard rate by investigating the slope of the best-fit line through the $\mathbf{C\hat{W}C}$ trajectory, where the slope should be directly proportional to the hazard rate. This is directly shown in Figure 2.2 (top) where the slope of the $\mathbf{C\hat{W}C}$ trajectory is 0.385 for the high hazard condition, and 0.019 for the low hazard condition. Using these slopes, we create a composite sensitivity metric for hazard rate, $\hat{s}_{hz}$, that is the difference in the slopes of the $\mathbf{C\hat{W}C}$ trajectory between the two conditions:

$$\hat{s}_{hz} = \frac{\Delta\mathbf{C\hat{W}C}_{high} - \Delta\mathbf{C\hat{W}C}_{low}}{\Delta t_{gray}} \tag{2.28}$$

As a measure, higher values of $\hat{s}_{hz}$ indicate a higher difference in behavioral sensitivity between the two task conditions as a function of time, and a value closer to zero indicates less of a difference, and we expect $\hat{s}_{hz}$ to be directly proportional to overall task performance.

Contingency Sensitivity

We perform a similar set of operations to define sensitivity for contingency but start by noting that unlike the hazard rate, contingency has no relationship to time and is strictly dependent on whether a velocity change occurred. So we plot the aggregated $\mathbf{C\hat{W}C}$ outcomes on bounce-path trials across contingency conditions to assess color choice outcomes. Similarly to hazard rate, we show in Figure 2.2 (middle), that the expected behavior is for the $\mathbf{C\hat{W}C}$ to trend positively as a function of contingency. Here we define contingency sensitivity, $\hat{s}_{cont}$, as the slope of the best-fit line through the contingency $\mathbf{C\hat{W}C}$ trajectory:

$$\hat{s}_{cont} = \frac{\Delta \mathbf{C\hat{W}C}_{cont}}{\Delta \text{Contingency}} \tag{2.29}$$

As a measure, higher values of $\hat{s}_{cont}$ indicate a higher difference in behavioral choice outcomes between the contingency conditions, and a value closer to zero indicates less of a difference. Similarly to $\hat{s}_{hz}$, we expect $\hat{s}_{cont}$ to be directly proportional to overall task performance.

2.2.6 Participant Data Collection

In this study, we generated two sets of task videos. One represents the main body of work where we test our overall hypotheses surrounding the Bouncing Ball task, and the other serves as a control. While some of the parameters for video generation are different, the overall task schema remains conserved.

Online Task Schema

Both online tasks were built using honeycomb and deployed on a Firebase web server. We recruited participants using prolific from all countries using a standard sample. Participants were filtered for English fluency and no prior participation in the Bouncing Ball task. After agreeing to the consent form for the study, we show the participants several prompts to explain the task, followed by videos that progressively displayed the

task dynamics. Each of the videos is presented with the same choice screens as the main task, and participants are instructed to indicate the color of the ball. Participants are first shown videos without the gray zone where they are explicitly told the sequence of colors, that some trials may have more or less color changes than others, and the ball can randomly change directions. The last two tutorial videos completing the section include the gray zone and inform participants that it hides the color but allows them to see the position of the ball.

Following the tutorial section is a practice section, where we show them five full trials, where a trial includes the main video, the color choice, and the confidence slider screens. They are provided feedback on their choices for each of these videos. Before beginning the main section of the study, they are instructed to answer as accurately and quickly as possible, as they will be compensated extra based on performance. Additionally, they are told to view confidence similarly to a wager, where it has a multiplicative effect on whether they get the answer correct or not. For both studies, the main experiment consisted of ten blocks of trials. Each block consisted of approximately 16 trials, where some would have one more. Unlike in the practice session, participants were given feedback after each block rather than each trial. They would also be given a five minute break before commencing the next block. Trial types were randomly distributed among the blocks and there was no functional difference between them.

Participant Task Videoset

Task Videoset Generation: The participant task videos are comprised of 168 trials: 81 straight path trials, 76 bounce path trials, and 11 attention check trials. For the latent probabilities, the specific values they take on represent a probability of change per timestep. In these task videos, P_{hz} is set to be 0.00875 and 0.15 for the low and high hazard rates respectively. P_{cont}, was set to be 0.05, 0.5, and 0.95 for the low, medium, and high contingencies respectively. We set P_{rvc} to be 0 in this study but explore its effects in

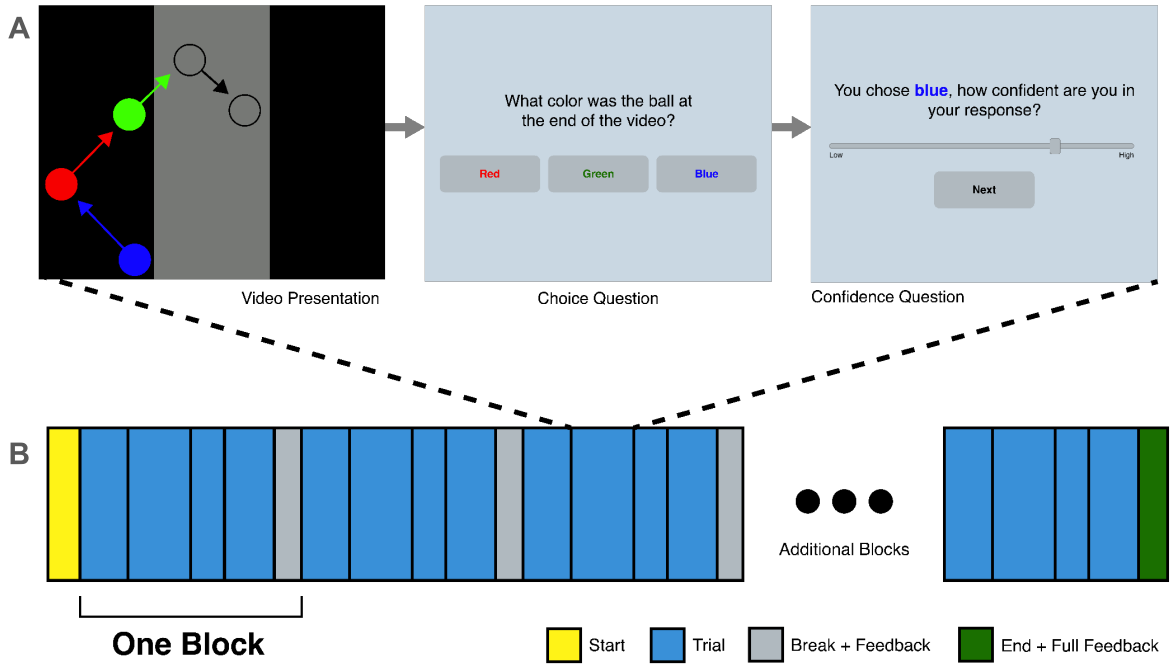


Figure 2.4: Online Task Schema. **(A)** Each trial consists of the video presentation, followed by the color choice, and the confidence entry, before the next trial begins. Confidence is selected between 0 and 100%. **(B)** The full task is organized into the start phase, followed by 10 blocks, and the end phase. Start - Consists of a tutorial which introduces participants to the mechanics of the task, followed by five practice trials with feedback. Block - Consists of 15 trials of randomly selected types, each with varying lengths. After the trials, participants are given a break period of five minutes and are provided feedback on the block. End - Feedback on the full experiment is provided before ending.

the next study. The x velocity was set to 0.11, and the two y-velocities to be selected from are 0.1 and 0.12 for the low and high velocity conditions. To minimize perceptual fusion of two random color changes and causal binding between bounces and random color changes, we set both rate-limiting variables τ_{dv} and τ_{dc} to 15 time steps, corresponding to 600 ms. While prior work indicates that visual temporal fusion for discrete changes only occur at much smaller intervals (Efron, 1967), investigative work on postdictive binding between two events showed that causal impressions were absent by 500 ms (De Freitas and Alvarez, 2018). Setting both variables to 600 ms provides a conservative margin above these thresholds, ensuring that color changes are correctly labeled. These values led to effective hazard rates, $\tilde{P}_{hz}$, of 0.00533 and 0.0376 for the low and high hazard rate conditions, respectively, and were used to perform the label calibration. We found the label sensitivities to be 0.353 for the hazard rate and 1.631 for contingency. A more detailed set of statistics for the task videos is shown in Figure 2.5.

Participant Recruitment: We recruited 100 participants from prolific for our main study. After performing basic filtering to include only participants who scored 80% correct or more on the attention check trials, we were left with 81 participants.

Velocity Change Task Videoset

In our main experimental design, velocity changes from wall collisions were predictable from the ball's trajectory and always coincided with wall contact. This presents an inherent confound about the nature of contingency learning in our task: Participants might learn abstract associations between velocity changes and color changes, or they might learn specific associations between wall contacts and color changes. The predictable nature of wall bounces, where participants can visually track the ball's approach and anticipate the collision, presents a fundamentally different learning signal than unpredictable temporal events. To disambiguate between these possibilities, we conducted another experiment where we set the random velocity change parameter P_{rvs} to be nonzero and introduced an

additional trial type: the random-bounce-path trial.

Random Bounce Path Trial Type: In the random bounce path trial type, the ball's velocity changes randomly while traversing the gray zone, without any wall contact. This change diminishes the predictability of velocity changes and momentarily removes the ability to visually track the ball, potentially during a contingent color change. If contingency learning requires predictable, visually-guided events, performance should significantly differ between wall-based and random velocity changes. Alternatively, if participants learn abstract, velocity-bound color changes, they should show similar sensitivity across both trial types.

Random bounce path trial type is structured similarly to both the straight path trial and the bounce path trial, and differs only in how the trial ends. Similar to the straight path trial, the ball enters the gray zone closer to the center where it will not hit a wall. And like the bounce path trials, the ball randomly bounces 5 time steps into the gray zone so we can control for the effect of the hazard rate across both trial types. To allow for direct control of when the "random" velocity change will occur, we introduced a forced velocity change flag, $\mathbf{F}_{rvc}^{(t:T)} = (f_{rvc}^{(t)}, \ldots, f_{rvc}^{(T)}) = \mathbf{0}^{(t:T)}$, manually set the flag to be 1 at the desired timestep, and modified the velocity change determination to include it:

$$dV^{(t)} = \begin{cases} 1, & \text{if } f_{rvc}^{(t)} = 1 \\ w^{(t)}, & \text{if } \max(w^{(t)}) = 1 \\ \text{Bernoulli}(P_{\text{rvc}}), & \text{if } r_{dv}^{(t-1)} = 0 \\ 0, & \text{otherwise} \end{cases} \tag{2.30}$$

Since contingency can be measured using both the bounce path trials and random bounce path trials, we can calculate separate contingency sensitivities for both trial types.

Experimental Design: The velocity change task videos are comprised of 161 trials: 76

straight path trials, 37 bounce path trials, 37 random bounce path trials, and 11 attention check trials. For the latent probabilities, the specific values represent a probability of change per timestep. To accommodate the increased complexity from the additional trial type, we lowered the high-hazard rate P_{hz} 0.065 but kept the low hazard rate at the same 0.00875. To further strain contingent-learning in the participants and models, P_{cont} was set to be 0.1, 0.5, and 0.9 for the low, medium, and high contingencies respectively, which yielded slightly more overlapping effective color changes. The key manipulation in this trial involved setting P_{rvc} to 0.09825. The x velocity was set to 0.11, and the two y-velocities, selected from low and high velocity conditions, were 0.0975 and 0.1125. Both rate limiting variables, τ_{dv} and τ_{dc}, were set to 15 time steps but were not used for hazard rate calibration, as we are primarily testing contingency, which is not affected by rate limiting. We found the label sensitivity for contingency to be 1.809 for the bounce path trials and 1.832 for the random bounce path trials. A more detailed set of statistics for the task videos is shown in Figure S1.

Analysis Focus: For these task videos, our analyses centered on the primary question of contingency generalization. Specifically, we examined whether participants showed comparable contingency sensitivity between wall and non-wall bounce trials, investigated individual differences in this generalization ability, and tested whether participants could perceptually discriminate between the two trial types. The experimental parameters resulted in a higher overall proportion of color changes compared to the main experiment (56.8% for wall bounces, 56.5% for random bounces). Unlike the participant task videos, we did not perform hazard rate calibration, which, combined with the lower hazard rates used, led to substantial overlap between the low and high hazard rate distributions. We also did not conduct behavioral clustering for these task videos, as our research question focused on whether all participants—regardless of their underlying strategy—would demonstrate similar learning patterns for wall-based versus non-wall velocity changes.

Participant Recruitment: We recruited 50 participants from prolific for the velocity

change study. After performing basic filtering to include only participants who scored 80% correct or more on the attention check trials, we were left with 43 participants.

2.2.7 Task Statistics and Choice Distribution

The generated task videos contained inherent statistical imbalances reflecting the probabilistic nature of the generative process. Understanding these distributions is crucial for interpreting participant behavior and strategy identification.

Overall Task Statistics

In order to keep the number of changes per video within ecologically relevant limits, we set the hazard rate to lower than what was required to generate balanced change outcomes. The resulting task videos were balanced between different final colors, but color changes occurred in only 33% of trials, while no-change outcomes dominated at 67%. Understanding the degree of the asymmetry is critical to utilizing the correct statistical techniques to deal with them.

Straight Path Trial Statistics: Straight path trials exhibited the strongest class imbalance, with only 21% of trials resulting in color changes. The probability of no-change outcomes varied systematically as a function of both gray zone position and hazard rate condition (Table 1). Position 1 trials, resulted in no-change outcomes regardless of hazard rate. As gray zone exposure increased, the rates increased as well. Position 2 trials showed some differentiation (81% vs 64% no-change) between hazard rates, while Position 3 had the largest difference (87% vs 33% no-change).

Position-Specific Distributions The probability of no change varied systematically with gray zone position and hazard rate:

Bounce Path Trial Statistics: Bounce path trials presented a much more balanced color change outcome separation (45% change, 55% no change) while having a wide range

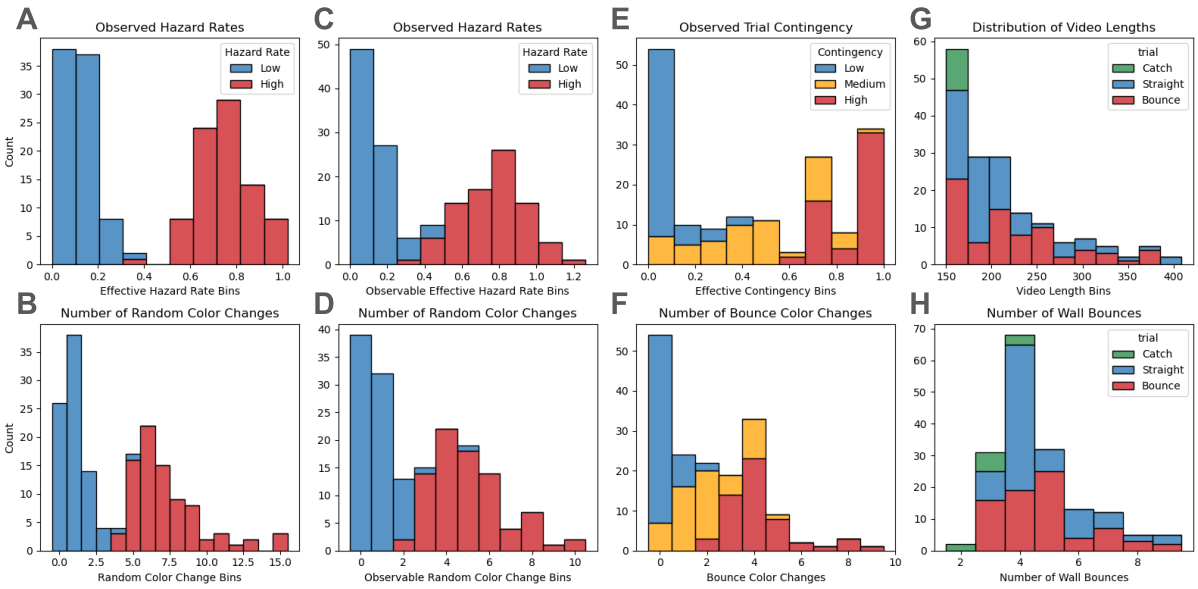


Figure 2.5: Participant Task Videos Overall Statistics. **(A)** Effective hazard rates for each video colored by hazard rate. Computed as the ratio between total random color changes (including those in the grayzone) divided by the number of eligible time steps (total number of non-bounce time steps). **(B)** Histogram of total random color changes per video colored by hazard rate. **(C)** Observable effective hazard rates for each video. Computed similarly as **(A)** but only using changes and time steps outside the grayzone. **(D)** Histogram of total random changes that occur outside the grayzone per video colored by hazard rate. **(E)** Effective contingency for each video colored by contingency. Computed as the ratio of bounce color-changes divided by number wall bounces. **(F)** Histogram of number of bounce color changes per video colored by contingency. **(G)** Histogram of video lengths in frames colored by trial type. **(H)** Histogram of number of bounces per video colored by trial type.

Condition	Position 1	Position 2	Position 3
Low Hz	1.00	0.81	0.87
High Hz	1.00	0.64	0.33

Table 2.1: Proportion of no-change outcomes by hazard rate and ending position

in changes between contingency conditions. Low contingency trials (5%) yielded 92% no-change outcomes, while high contingency trials (95%) inverted this distribution with 92% change outcomes. Medium contingency trials (50%) exhibited intermediate statistics with 64% no-change outcomes. These distributional properties fundamentally shaped the behavioral strategies that emerged in the human participants and models, as they confronted a task where they must learn the statistics in-context for each video.

2.2.8 Statistical Analysis

Performance Metrics

Given the 67% no-change base rate in our task, standard accuracy metrics would mask meaningful performance differences. We therefore employed balanced accuracy and Matthews Correlation Coefficient (MCC) as our primary metrics. Balanced accuracy is computed as $\frac{\text{sensitivity+specificity}}{2}$, which weights both classes equally, and MMC is calculated as:

To test whether observers

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{2.31}$$

Where sensitivity (true positive rate) = $\frac{TP}{TP+FN}$, specificity (true negative rate) = $\frac{TN}{TN+FP}$.

Behavioral Strategy Identification

To identify different behavioral strategies on our task, we applied hierarchical clustering to participants' response patterns across trial types. For each participant, we first constructed separate 2×2 confusion matrices for straight and bounce trials, capturing their change detection performance in each condition. These matrices were normalized by true class counts to account for the inherent class imbalance in our stimuli. We then concatenated these two matrices into a single 4×2 representation for each participant, where rows 1-2 contained straight trial responses, and rows 3-4 contained bounce trial responses. This concatenation preserved the trial-type behavioral differences while allowing us to cluster participants based on their performance across both conditions. After flattening these concatenated matrices into feature vectors and standardizing across participants, we performed hierarchical clustering using Ward's linkage method. This approach identified groups of participants with similar patterns of change detection behavior across both trial types. To determine optimal cluster number, we computed four validation metrics for $k = 2$ through $k = 8$: Within-Cluster Sum of Squares (elbow method), Silhouette Score, Calinski-Harabasz Index, and Davies-Bouldin Index. Three of four metrics converged on $k = 2$, suggesting two distinct strategies to our task (fig. S2). This two-cluster solution aligned with our hypothesis that participants would adopt different strategies in response to the class imbalance inherent in the stimuli.

Statistical Tests

To test whether observers learned task structure above chance, we performed single-sample t-tests against zero for both sensitivity metrics. For group comparisons, we used Welch's ANOVA as an omnibus test across observer types, with effect sizes reported as η^2. Pairwise comparisons employed Welch's t-tests, with all observers compared to the IBO as reference. Given our exploratory aims and specific hypotheses about observer-IBO differences, we report all comparisons without correction for multiple testing. To examine

relationships between learning the hazard rate and contingency, we computed Pearson correlations between the respective sensitivities, both overall and within each observer type.

Velocity Change Analyses: The control experiment required within-subject comparisons using paired t-tests between wall bounce and random bounce sensitivities, as well as comparisons of each bounce type to straight trials. We assessed individual differences through correlations between wall and random bounce performance, and performed median split analyses on wall bounce sensitivity. To quantify generalization, we computed discrimination indices as mean absolute performance differences between trial types, then compared wall-random discrimination against wall-straight discrimination. The ratio of these indices provided a measure of generalization strength.

All analyses were performed at the observer level rather than the trial level, with performance metrics averaged within observer, before statistical testing. This yielded sample sizes of n=100 (IBO), n=100 (LSTM), n=100 (RNN), n=81 participants (C1: n=37, C2: n=44), and n=42 for the control experiment.

2.2.9 Computational Models

To investigate potential computational mechanisms underlying human performance and to establish whether the observed behavioral patterns reflect computational constraints rather than observer-specific limits, we trained recurrent neural networks on the Bouncing Ball task. By comparing model performance to both human participants and the ideal Bayesian observer, we could assess whether artificial systems develop similar strategies when faced with identical statistical learning challenges.

Model Architecture and Learning Framework

We evaluated two recurrent neural network architectures that differ in their abilities to learn long-term dependencies over extended temporal sequences. The first was a

vanilla recurrent neural network (RNN) with a single recurrent layer that maintains a hidden state across time steps. The second was a Long Short-Term Memory network (LSTM) that employs gating mechanisms and a separate cell state to better retain long-range dependencies (Hochreiter and Schmidhuber, 1997). Both models used identical three-module architectures: a feedforward encoding block that processes input features, a recurrent module with 34 or 16 hidden units for the RNN and LSTM respectively (parameter matched), and an output layer that generates color predictions. We discuss the model architecture in more detail in Section 3.2.1.

Rather than framing the task as direct classification, where models predict only the final occluded color, we implemented a semi-supervised learning approach that more closely captures the continuous nature of state inference during sequential observation((Kim, 2011; Ghosh et al., 2025). In this framework, models receive the current observable state $o^{(t)} = (x^{(t)}, \tilde{c}^{(t)})$ at each timestep and generate a prediction for the color at the next timestep $\hat{c}^{(t+1)}$. This approach offers several advantages over end-point classification. First, it encourages the development of representations that capture the temporal dynamics of the task rather than memorizing statistical regularities for final-step predictions (Goroshin et al., 2015; Gamboa, 2017; Wang et al., 2023). Second, it mirrors the continuous updating process that likely underlies human performance, where observers maintain and update beliefs throughout the trial rather than making isolated judgments at the end (Prat-Carrabin et al., 2021; Radvansky et al., 2003). Third, the models can still be directly evaluated using the same behavioral metrics as human participants by examining their predictions at task-relevant time steps.

The semi-supervised nature of the training emerges from our treatment of occluded states. When the ball traverses the gray zone, the true color $c^{(t)}$ is provided to the model during backpropagation, despite being visually occluded in the input. This design choice serves as an inductive bias that encourages models to maintain accurate hidden representations of the latent color state even when direct observation is unavailable,

analogous to how humans maintain object representations during occlusion.

Model Training Task Videoset

While we sought to preserve the core statistical structure and task dynamics from the human experiments, one key adaptation was necessary to achieve model convergence in reasonable time. In the human participant task videos, rate-limiting parameters $\tau_{dc} = \tau_{dv} = 15$ time steps ensured that consecutive color and velocity changes were perceptually distinguishable as discrete events. However, computational models can process information at the frame level without the perceptual limitations of human observers. We therefore set both rate-limiting parameters to 0, allowing color and velocity changes to occur at any eligible timestep.

To maintain statistical equivalence with the participant task videos despite this mechanistic difference, we needed to ensure models would perform in-context learning rather than simply memorizing discrete parameter values. Training on just the two hazard rates (0.00533 and 0.0376) and three contingency values (0.05, 0.5, 0.95) used in human experiments would likely result in models that classify trials into these fixed categories rather than continuously tracking the statistics. To address this limitation, we trained models on a continuous spectrum of parameter values. For hazard rate, values were drawn from a linear spacing of raw hazard rates between 0.001 and 0.05, which encompasses and extends beyond the values used in human experiments. Similarly, contingency values were drawn from a linear spacing between 0 and 1, ensuring exposure to the full range of possible velocity-color change relationships. For both parameters, we used a spacing as large as the batch size used for training, which was 1024 samples. This training scheme forces the models to develop in-context learning mechanisms that track arbitrary parameter values within each trial, rather than pattern-matching to a small set of memorized conditions in the earlier time steps. During evaluation, we tested models on the same task videos used for the participants, which use discrete parameter values, allowing direct comparison

while ensuring the observed network behavior reflect true in-context learning.

The final parameter that was different from the participant task videos was the velocity. For similar reasons to the hazard rate and contingency, we wanted to ensure the models did not overfit to the specific values used for humans. We also used a linear spacing of velocities that encompassed the velocities in the participant task videos for both the x and y components of velocity. The spacing size was also 1024 for the velocity components and spanned 0.08 to 0.13. All other parameters such as the ball radius and gray zone properties remained identical to the human experiments.

Training Procedure

Models were trained using the cross-entropy loss between predicted and true color across all time steps. Training proceeded for 10,000 epochs with batches of 1,024 sequences, where each sequence contained 600 time steps. We employed the AdamW optimizer with a learning rate of 0.001 and weight decay of 0.01 to prevent overfitting. AdamW typically provides better generalization over Adam by decoupling weight decay from gradient updates (Loshchilov and Hutter, 2019).

Due to the computational demands of processing 600-timestep sequences, we used truncated backpropagation through time with a window of 150 time steps. This approach maintains hidden states across the entire sequence while limiting gradient computation to manageable segments, balancing the need to capture long-range dependencies with computational efficiency. The truncation window, extended sequence length, in combination with the semi-supervised objective, encouraged models to develop temporal representations rather than relying on short-term pattern matching. Training progress was monitored using a held-out validation set consisting of the same 168 trials presented to human participants, allowing us to track both convergence and task sensitivity. We go through the training procedure in detail in Section 3.2.1.

Model Evaluation Protocol

To enable direct comparison with human behavior, we evaluated trained models on identical trial sequences using the same performance metrics. Each of the models (10 seeds) processed each of the 168 experimental trials (81 straight path, 76 bounce path, 11 attention check) and generated color probabilities at the same decision points where human participants made their choices. Then for every trial, we obtain the model's color probability distribution $\mathbf{p}^T$ at the final timestep T, and generate synthetic color choices by sampling from these predictions:

$$c_j \sim \text{Categorical}(\mathbf{p}_j^T), \quad j = 1, \ldots, N_p \tag{2.32}$$

where $N_p = 81$, is the number of participants. Then for each sampled color $c_{i,j}$, we compute the confidence-weighted choice (CWC) by multiplying an indicator function by the probability that generated the color choice:

$$m_{i,j} = \mathbb{1}[c_{i,j} = k] \cdot p_{k,i}^{(T)} \tag{2.33}$$

where k is the sampled color and $p_{k,i}^{(T)}$ is its corresponding probability from the model's distribution. This allowed us to compute hazard rate sensitivity ($\hat{s}_{hz}$) and contingency sensitivity ($\hat{s}_{cont}$) using the identical analysis pipeline applied to human data. Beyond the participant task videos, we also evaluated models on an extended videoset of 20,000 videos generated with identical statistical parameters. This larger task videos enabled more robust statistical analyses and would later support mechanistic investigations described in Chapter 3.

2.3 Results

2.3.1 Theoretical Predictions of the Ideal Bayesian Observer

The Bouncing Ball task challenges humans and models to perform in-context learning of the latent parameters, hazard rate and contingency, in order to make color predictions on a video-by-video basis. These two parameters govern how the ball's color changes during each video, where hazard rate is the probability of a color change when the ball does not experience a velocity change, and contingency is the probability of a color change when there is a velocity change. Unlike tasks where parameters remain fixed across trials, here observers must infer these parameters anew for each video based on the observed statistics of color and velocity changes. Before examining how participants and neural networks accomplish in-context learning, we first establish predictions using the Ideal Bayesian Observer (IBO), which performs optimal statistical inference given the generative structure of the task. The IBO's behavior indicates that successful in-context learning requires solving two different problems:

1. Tracking temporal probabilities that accumulate over time (hazard rate)
2. Detecting event-contingent associations that occur instantaneously (contingency)

These computational demands generate testable predictions about how parameter sensitivity should manifest in behavior.

Predictions for Hazard Rate Learning

The hazard rate controls the probability of spontaneous color changes when the ball travels without velocity changes. Optimal in-context learning of this parameter requires the IBO continuously update its estimate of the hazard rate based on observed color changes throughout the video. This leads to the clear behavioral signature of making increasingly higher color-change predictions as a function of time elapsed.

In the Bouncing Ball task, the grayzone allows us to directly examine how these color predictions change in time while the color is occluded. As shown in Figure 2.2B, the IBO exhibits distinct color-change probability trajectories for the low (≈0.5% per timestep) and high (≈3.5% per timestep) hazard rate conditions. If we sample a color choice from this probability and then scale it by the probability that generated it, we obtain the confidence weighted choice (CWC), a metric we can directly compare with the behavioral outcomes of human participants. Setting the sampling positions to be at three locations along the horizontal axis, we obtain a two CWC trajectories, similar to the underlying probability trajectories (fig. 2.2C). The difference in the slope and intercept between these trajectories defines the optimal behavioral outcome between the conditions for the hazard rate.

For human participants and neural networks to demonstrate successful in-context learning of hazard rate, we should observe three related behavioral signatures: (1) systematic increases in color-change predictions with grayzone duration, (2) steeper prediction slopes for videos with higher learned hazard rates, and (3) systematic increases in color-change predictions after bounces on videos with higher contingency values.

Predictions Contingency Learning

Contingency controls the probability of color changes triggered by velocity changes, whether from wall bounces or random velocity changes. Optimal in-context learning of this parameter requires the IBO to continuously update its estimate of contingency based on observed coincidences between velocity and color changes throughout a video. This leads to the clear behavioral signature of making color-change predictions that scale with the learned contingency value whenever a velocity change occurs.

In the Bouncing Ball task, both the bounce and nonwall trials allow us to directly examine how these contingency-dependent predictions manifest when the ball bounces off a wall within the grayzone. As shown in Figure 2.2E, the IBO exhibits distinct color-change

probability levels for the low (5%), medium (50%), and high (95%) contingency conditions at the moment of the bounce. In the same manner as for the hazard rate, we can sample a color choice from this probability and then scale it by the probability that generated it to get the CWC. Plotting the CWC values across the three contingency conditions shows a positively increasing slope, with values ranging from strongly negative (favoring no change) in the low contingency condition to strongly positive (favoring change) in the high contingency condition (fig. 2.2F). Critically, the slope across contingency values differs from zero only when the observer has successfully learned the contingency parameter from the observed statistics and defines our ceiling for contingency sensitivity, representing the optimal differentiation in behavioral outcomes between the conditions.

For human participants and neural networks to demonstrate successful in-context learning of contingency, we should observe three related behavioral signatures: (1) minimal color change predictions following velocity changes in low contingency videos, (2) high color change predictions following velocity changes in high contingency videos, and (3) sensitivity values that approach the optimal value defined by the IBO.

Summary of Predictions

The IBO's optimal in-context learning generates two key predictions that we will test in human participants and neural network models. First, observers that successfully learn the hazard rate should show time-dependent sensitivity with CWC slopes that differ between low and high hazard conditions. Second, observers that successfully learn contingency should show event-dependent sensitivity with CWC values that scale linearly across contingency conditions.

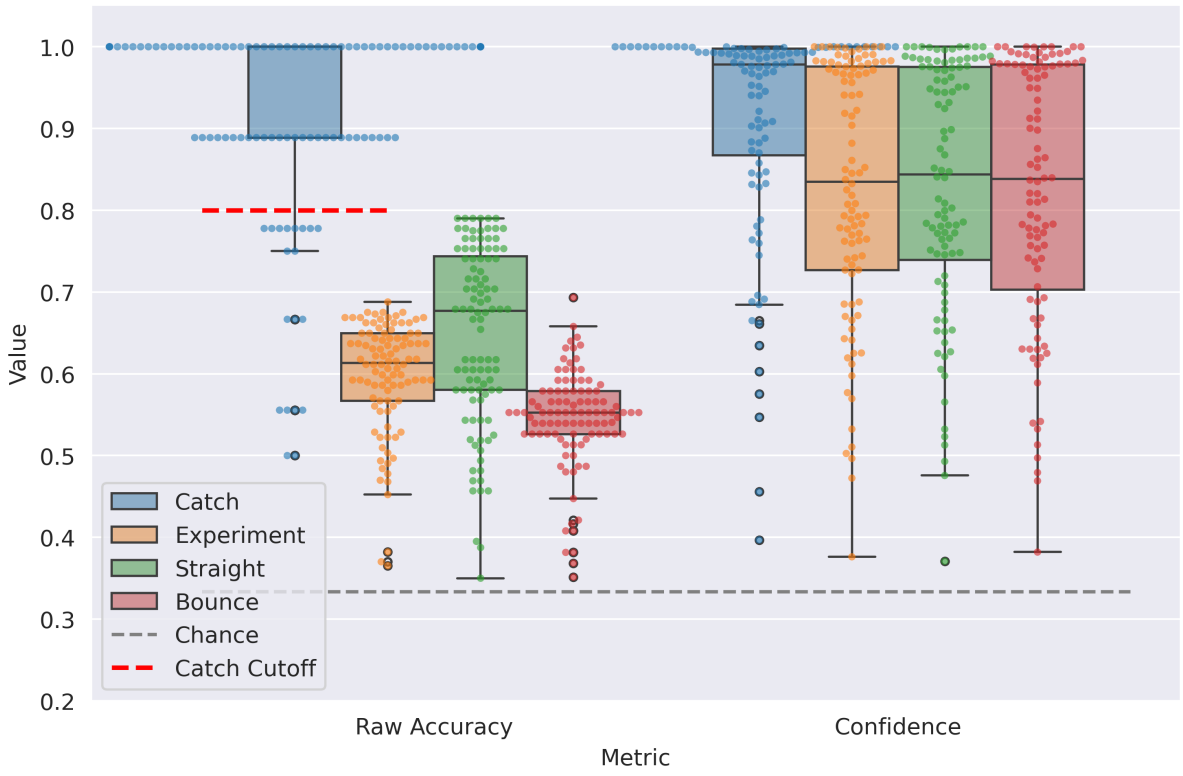


Figure 2.6: Participant Raw Accuracy and Confidence. Unfiltered participant (n=100) performance across trial types in the participant task videos. **Left:** Raw accuracy distributions for catch trials (blue), where the ball remained visible throughout, show near-ceiling performance well above the 80% exclusion criterion (red dashed line). Experimental trials overall (orange) yielded moderate accuracy around 0.63, with straight path trials (green) performing better (≈0.69) than bounce path trials (red, ≈0.55). All trial types exceeded chance performance (gray dashed line, 0.33). **Right:** Raw confidence ratings across all trial types.

2.3.2 Behavioral Signatures of In-Context Learning

Task Comprehension and Inclusion Criteria

Before examining the degree to which participants learned the underlying structure of the task, we first established that they understood the overall task mechanics. While all 11 catch trials ended with the ball in a visible region the screen, for our exclusion criterion we only used the 9 trials where the ball was visible for 400 ms. This ensured that the observed performance reflected task comprehension and not perceptual limitations. Of the 100 participants we recruited from Prolific for our main study, 81 met the attention check requirements of 80% accuracy on the catch trials where median performance was 100% raw accuracy On the clearly-visible catch trials, the included participants achieved near-ceiling accuracy (M=0.962, SD=0.052), where 54 made no errors (fig. 2.6 left). Participants also indicated higher confidence on the attention check trials (M=0.929, SD=0.116) compared to experimental trials (M=0.819, SD=0.168), validating they distinguished between these trials with high certainty from the others (fig. 2.6 right).

Confidence Weighted Choice Validates Behavioral Signatures

To examine whether the participants and models exhibited the predicted behavioral signatures, we used their responses to compute the confidence-weighted choices (CWC) for all trials. The CWC aggregates the color choice and confidence (probability of a class for models) into a single metric ranging from -1 to 1. We then plot the outcomes for straight path trials and wall bounce trials to evaluate hazard rate and contingency learning.

For hazard rate learning, the IBO predicts systematically different trajectories for the high and low hazard rates (fig. 2.2C). Since time-elapsed leads to an increased likelihood of change, the CWC trajectory for the high hazard rate condition should have both more positive values for its trajectory and a larger slope. Comparing the aggregate CWC trajectories for the participants and models shows that this behavioral signature holds (fig. 2.7, left). For all observers, the low hazard rate CWC trajectory stays beneath that

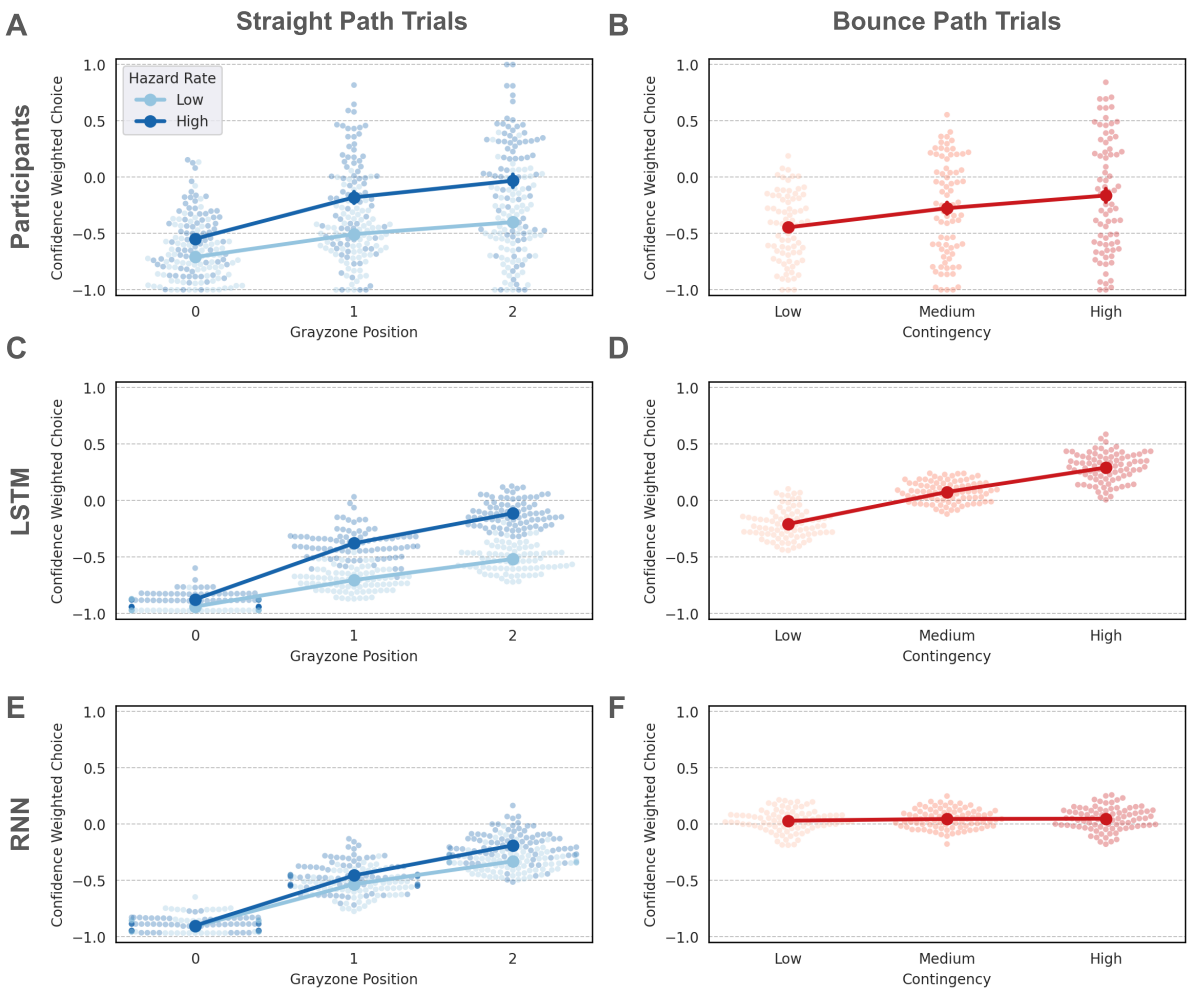


Figure 2.7: Participant, LSTM, RNN Confidence Weighted Choice Trajectories. We calculate the confidence weighted choice (CWC) for participants who scored 80% or more on the attention checks, and the recurrent models (LSTM, RNN). The CWC trajectories are split between outcomes for the straight path trials and the wall bounce trials for each observer. CWC trajectories for straight path trials examine the effects of hazard rate on predictions by showing how they evolve through three positions along the x-dimension (y varies). The resulting trajectories of both high (dark blue) and low (light blue) hazard rate conditions are shown for all observers. CWC trajectories for wall bounce trials examine the effects of contingency on predictions by showing how varying contingency values affects choice outcomes following a bounce **(A)** Participant hazard rate CWC trajectories show increasing trend for each condition, with high hazard rate trajectory at a higher overall value and increasing faster. **(B)** Participant contingency CWC trajectories shows increasing trend across all conditions. **(C)** LSTM hazard rate CWC trajectories show increasing trend for each condition, with high hazard rate trajectory at a higher overall value and increasing faster. **(D)** LSTM contingency CWC trajectories shows increasing trend across all conditions. **(E)** RNN hazard rate CWC trajectories show increasing trend for each condition, with high hazard rate trajectory at a higher overall value and increasing faster. **(F)** RNN contingency CWC trajectories flat across conditions. Indicates no change in predictions across contingency conditions.

of the high hazard rate trajectory, indicating that fewer color-change predictions are being made and with less confidence. Additionally, examining the trajectories in time shows that the high hazard rate trajectories rise at a faster rate along the three grayzone positions. Although the degree to which these outcomes are present differs between the observers, the clear separation between conditions across the participant and models confirms our task paradigm successfully captures temporal tracking of color-change probability.

For contingency learning, the IBO predicts a rising trajectory through the various contingency conditions (fig. 2.2F). As the contingency increases, the probability of a color-change when the ball bounces increases, leading to the predicted outcome of increased number of color-change choices, and at a higher confidence. Comparing the aggregate CWC trajectories for the participants and models shows that this behavioral signature also holds (fig. 2.7, right). Unlike for the hazard rate, we see that participants and the LSTM CWC trajectory displays a clear increasing slope across the three contingency conditions but not for the RNN (fig. 2.7F). The lack of this ordering across conditions for the RNN indicates that the model did not change its color predictions as contingency changed. The presence of this pattern in the participants and LSTM, indicates that the absence of this pattern in the RNN most likely reflects the lack of computational machinery to learn contingent color changes, rather than contingency not being learnable, a topic that will be explored in later sections. Despite this partial presence, the presence of an increasing CWC trajectory across contingency conditions for the participants and LSTM confirms our task paradigm successfully captures event-based learning of color-change probability.

Put together, the CWC trajectories for the observers across the trial types confirms that our experimental paradigm captures the two forms of in-context learning we investigate in this study. We next sought to to quantify the magnitude of these effects by computing and comparing sensitivity metrics for each task parameter.

2.3.3 In-Context Learning Performance

Having validated our behavioral paradigm, we examined the degree to which participants and models learned each task parameter compared to the IBO. All observers demonstrated statistically significant sensitivity to both hazard rate and contingency (all $p < 0.05$), however performance varied between the participants and models.

	Hazard Rate Sensitivity		**Contingency Sensitivity**	
Observer Type	**Mean (SD)**	**% of Optimal**	**Mean (SD)**	% of **Optimal**
IBO	0.377 (0.063)	100.0%	0.660 (0.121)	100.0%
LSTM	0.190 (0.107)	50.3%	0.533 (0.250)	80.7%
Participant	0.058 (0.151)	15.5%	0.315 (0.313)	47.7%
RNN	0.081 (0.106)	21.5%	0.068 (0.153)	10.2%

Table 2.2: Hazard Rate and Contingency Sensitivity Across Observers. Hazard Rate Sensitivity is defined as the difference in the slope of the mean confidence weighted choice (CWC) through time on the straight path trials. Contingency Sensitivity is defined as the slope of the mean CWC across contingency conditions. IBO represents optimal performance on the task. All observers showed significant learning ($p < 0.05$) for both parameters.

Parameter Sensitivity

Hazard Rate: We first examine the degree to which observers demonstrate the predicted sensitivity to hazard rate. The LSTM networks achieved the highest hazard rate sensitivity among non-IBO observers (0.190 ± 0.012; 50.3% of optimal performance, $p < 0.001$), outperforming the vanilla RNN architecture (0.081 ± 0.012; 21.5% of optimal, $p < 0.001$) (fig. 2.8 green, red). Human participants demonstrated minor hazard rate learning, achieving only 0.058 ± 0.017 or 15.5% of optimal performance ($p = 0.001$) (fig. 2.8 blue). Performance on straight path trials was dependent on both time spent in the grayzone and hazard rate conditions, where performance deteriorated predictably across all observers as time through the grayzone increased.

Contingency: We next examine whether observers show the predicted sensitivity to contingency (fig. 2.8, right) and found a different profile of learning. Participants achieved 0.315 ± 0.313 contingency sensitivity, representing 47.7% of optimal ($p < 0.001$),

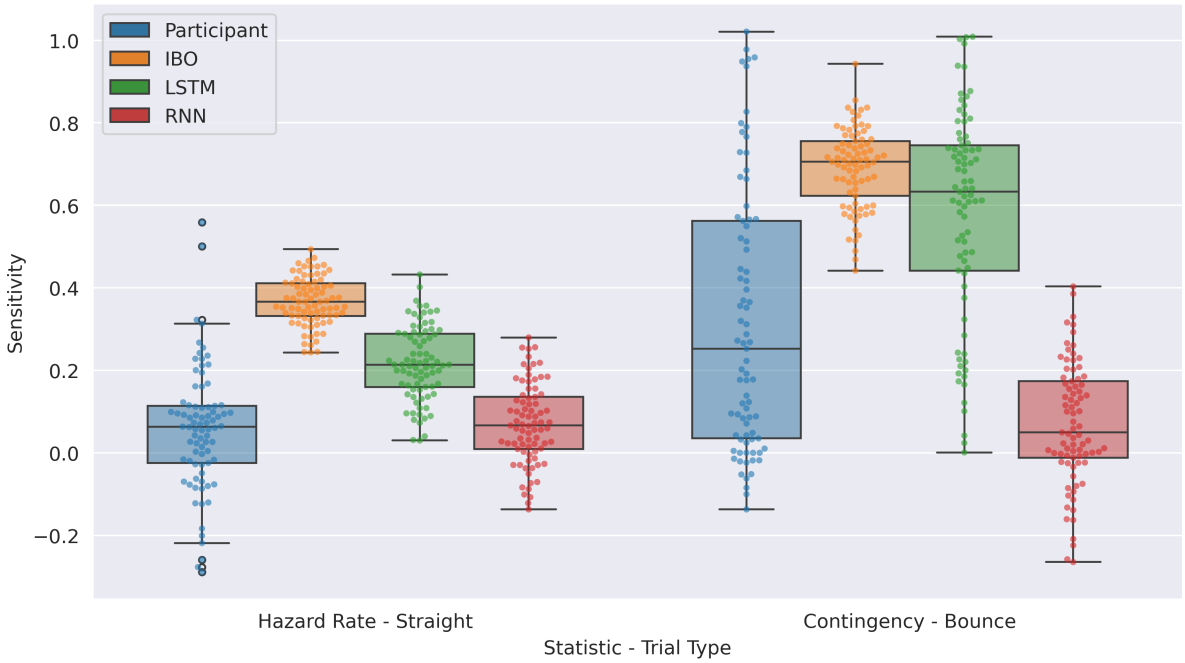


Figure 2.8: Observer Hazard Rate and Contingency Sensitivity Differential learning of task parameters reveals asymmetric learning capabilities. Hazard rate sensitivity is calculated as the the slope of confidence-weighted choice trajectories across the time steps in the grayzone. Contingency sensitivity defined as the slope of the confidence weighted choice across the three contingency values. Using the IBO, we can define the optimal sensitivities of 0.377 for hazard rate and 0.660 for contingency. Left: Hazard rate sensitivity on straight path trials shows LSTM networks reach 50.3% of optimal (0.190 $\pm$ 0.012), and human participants demonstrate minimal learning at 15.5% of optimal (overall: 0.058 $\pm$ 0.017). Right: Contingency sensitivity across conditions shows the LSTM achieves 80.7% of optimal performance (0.533 $\pm$ 0.028)

showing much more varied and substantial performance than for hazard rate. The LSTM architecture approached near-optimal performance (0.533 ± 0.250; 80.7% of optimal, $p < 0.001$) whereas the RNN showed minimal contingency sensitivity (0.068 ± 0.153; 10.2% of optimal, $p < 0.001$). Performance varied inversely with contingency across all observers, with the low contingency condition yielding the highest accuracy.

Classification Performance

Having examined the choice profiles of the participants and models, we evaluated their choices relative to the true labels. Using the raw color choices made by participants (raw accuracy in Figure 2.6) and models, we computed their color-change and non-change choices, where we compared their final color choice to the color that entered the grayzone for the last time. We found that despite the above chance accuracy for all observers, computing the Matthews Correlation Coefficient (MCC) revealed the task to be intrinsically difficult. Using the IBO as the benchmark, it achieved MCC = 0.018 while having 71.7% accuracy, indicating that even optimal inference struggles to discriminate between change and non-change. This performance is translated to the humans and models, where it is clear that task accuracy is largely driven by specificity.

Observer Type	Accuracy	Bal. Acc.	MCC	Sensitivity	Specificity
IBO	0.717 (0.038)	0.648 (0.041)	0.018 (0.068)	0.413 (0.071)	0.716 (0.058)
LSTM	0.681 (0.038)	0.622 (0.041)	−0.013 (0.074)	0.386 (0.080)	0.690 (0.055)
Participant	0.664 (0.050)	0.622 (0.044)	0.011 (0.059)	0.344 (0.256)	0.732 (0.191)
RNN	0.639 (0.032)	0.608 (0.040)	−0.019 (0.073)	0.405 (0.102)	0.662 (0.045)

Table 2.3: Performance Metrics on the Bouncing Ball Task. Values shown as mean (standard deviation) for each metric. Accuracy: proportion of correct change predictions; Balanced Accuracy: average of sensitivity and specificity, accounting for class imbalance; MCC: Matthews Correlation Coefficient ranging from -1 to 1, where 0 indicates random performance; Sensitivity: true positive rate (hit rate); Specificity: true negative rate (correct rejection rate).

Participant Within-Group Differences

The results from the sensitivity metrics show that participants have a clear asymmetry in learning capability between hazard rate and contingency. On the one hand, they have low relatively low sensitivity to hazard rate, only reaching 15.5% of optimal behavior. On the other hand, they excel at detecting the differing contingency conditions (47.7% of optimal), and vary greatly (SD = 0.313) between participants, with some matching optimal sensitivity. Further, when placed side-by-side, the range of sensitivity values for participants spans that of all other three models, from the RNN to the IBO. This variance suggested there could be key differences in the approaches taken by sub-populations of participants.

Investigating the performance outcomes more directly, we performed a median split on the final accuracy of participants. We found the performance of the top split was largely driven by their accuracy in the straight path trials, whereas their performance on the wall bounce trials were at or near the base rate for the trial type (55%). Looking directly at their confusion matrices revealed the responses of this subgroup of participants reflected a no-change strategy. These patterns motivated a data-driven approach to identify potential behavioral strategies.

2.3.4 Multiple Strategies

Identifying Behavioral Strategies

To better understand the potential differences in human performance, we performed hierarchical clustering on participants' change confusion matrices (fig. 2.9). Three of four validation metrics converged on an optimal two-cluster solution ($k = 2$, silhouette coefficient = 0.585), with consensus from the Silhouette Score, Calinski-Harabasz Index, and Davies-Bouldin Index, and only the elbow method suggesting $k = 4$ (fig. S2). This consensus for a two-cluster solution suggests a dichotomy in the strategies employed by the participants when faced with the inherent class imbalance (67% no-change trials) of

the task.

The first group exhibited a highly conservative change base rate (C1, n=37), as evidenced by their high specificity (0.914, SD = 0.199) and minimal sensitivity (0.100, SD = 0.245), effectively defaulting to "no change" responses across trials (fig. 2.9B). This is further reflected in their normalized confusion matrix, with all values in the right-most column (change) being zero for both trial types (fig. 2.9C). These participants achieved the highest accuracy on the task (69.5% accuracy) by exploiting the high base rate for no-change, further evidenced by their near-zero MCC (-0.001).

The second group of responders distributed their responses more evenly (C2, n=44), with comparable sensitivity (0.554, SD = 0.430) and specificity (0.577, SD = 0.365)(fig. 2.9B). This is also confirmed by examining their change confusion matrices, which have higher values along the diagonal than the off-diagonal elements, indicating active task-discrimination. However, this more balanced approach yielded a lower accuracy (63.7%) than the C1 group, but a positive MCC (0.021), further indicating engagement with the inference problem of the task.

Strategy-Specific Performance

Having identified the two subgroups of participants, we next investigated the behavioral outcomes resulting from the strategies, how it compared to the rest of the models. Hazard rate sensitivity remained similarly low for both the C1 (0.051 $\pm$ 0.136, 13.6% of optimal) and C2 (0.065 $\pm$ 0.165, 17.2% of optimal) groups and was not significantly different (p = 0.67). This similarity in insensitivity to hazard rate suggests limitations of human performance on the task is more likely than strategy differences. However, contingency sensitivity showed a different pattern. C1 responders showed minimal contingency sensitivity at 0.075 $\pm$ 0.134 (11.4% of optimal) whereas C2 responders achieved near-optimal sensitivity at 0.521 $\pm$ 0.274 (78.9% of optimal). This difference in sensitivity is a direct result of their approaches to the task.

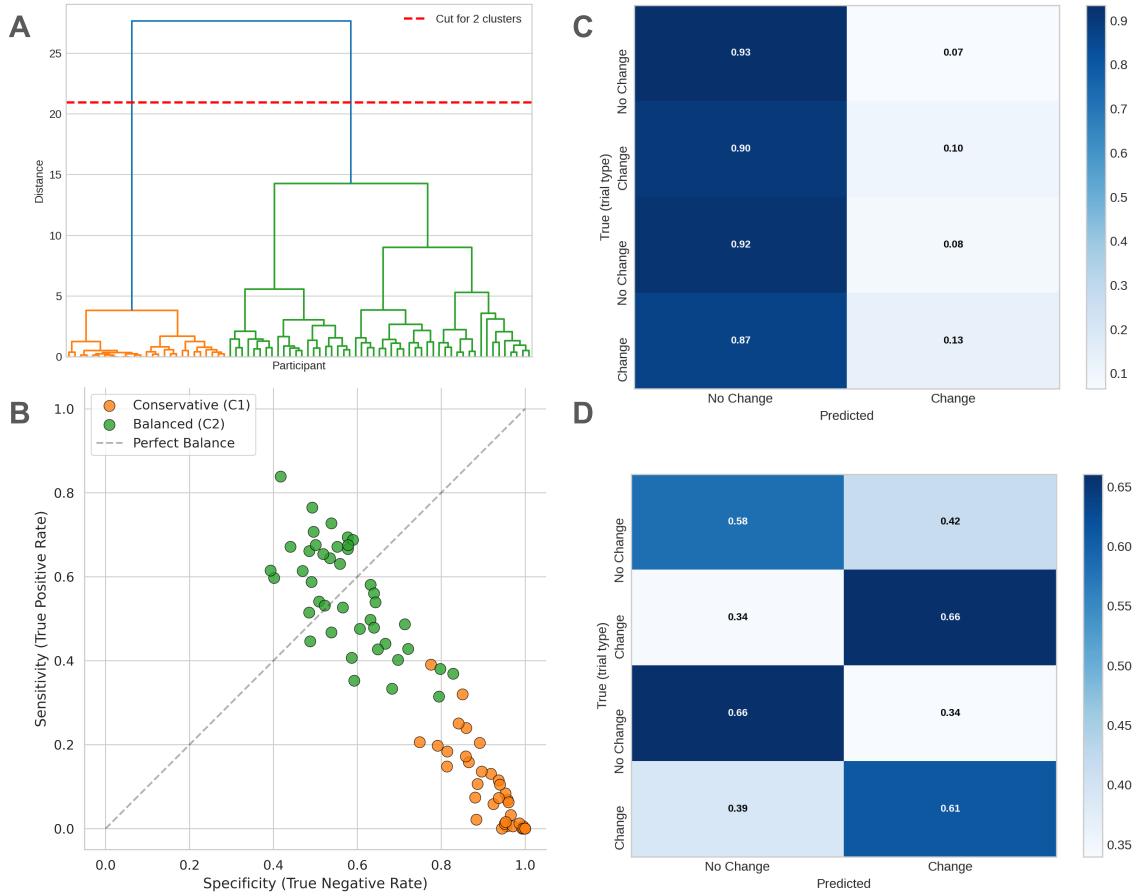


Figure 2.9: Identifying Participant Strategies. Identification of distinct behavioral strategies through hierarchical clustering of participant confusion matrices. For each participant, we concatenated their 2×2 confusion matrices from straight and bounce path trials into a 4×2 representation, where rows 1-2 correspond to straight trial responses and rows 3-4 to bounce trial responses. Hierarchical clustering with Ward's linkage was applied to these concatenated matrices, clustering participants based on their probability of correctly identifying change and no-change events in each trial type. The optimal cluster number ($k = 2$) was determined by multiple validation metrics. **(A)** Dendrogram showing hierarchical clustering of 81 participants with two-cluster solution indicated by red dashed line, separating C1 responders (orange, n=37) and C2 responders (green, n=44). **(B)** Scatter plot of sensitivity (recall) versus specificity revealing C1 responders (orange) clustering in high-specificity/low-sensitivity region (specificity: 0.914, sensitivity: 0.100) while C2 responders (green) show more balanced performance (specificity: 0.577, sensitivity: 0.554). **(C)** Mean confusion matrix for C1 responders showing strong no-change bias with 91.4% no-change responses (rows 1-2: straight trials; rows 3-4: bounce trials). **(D)** Mean confusion matrix for C2 responders showing more uniform response distribution with 44.6% no-change responses (rows 1-2: straight trials; rows 3-4: bounce trials).

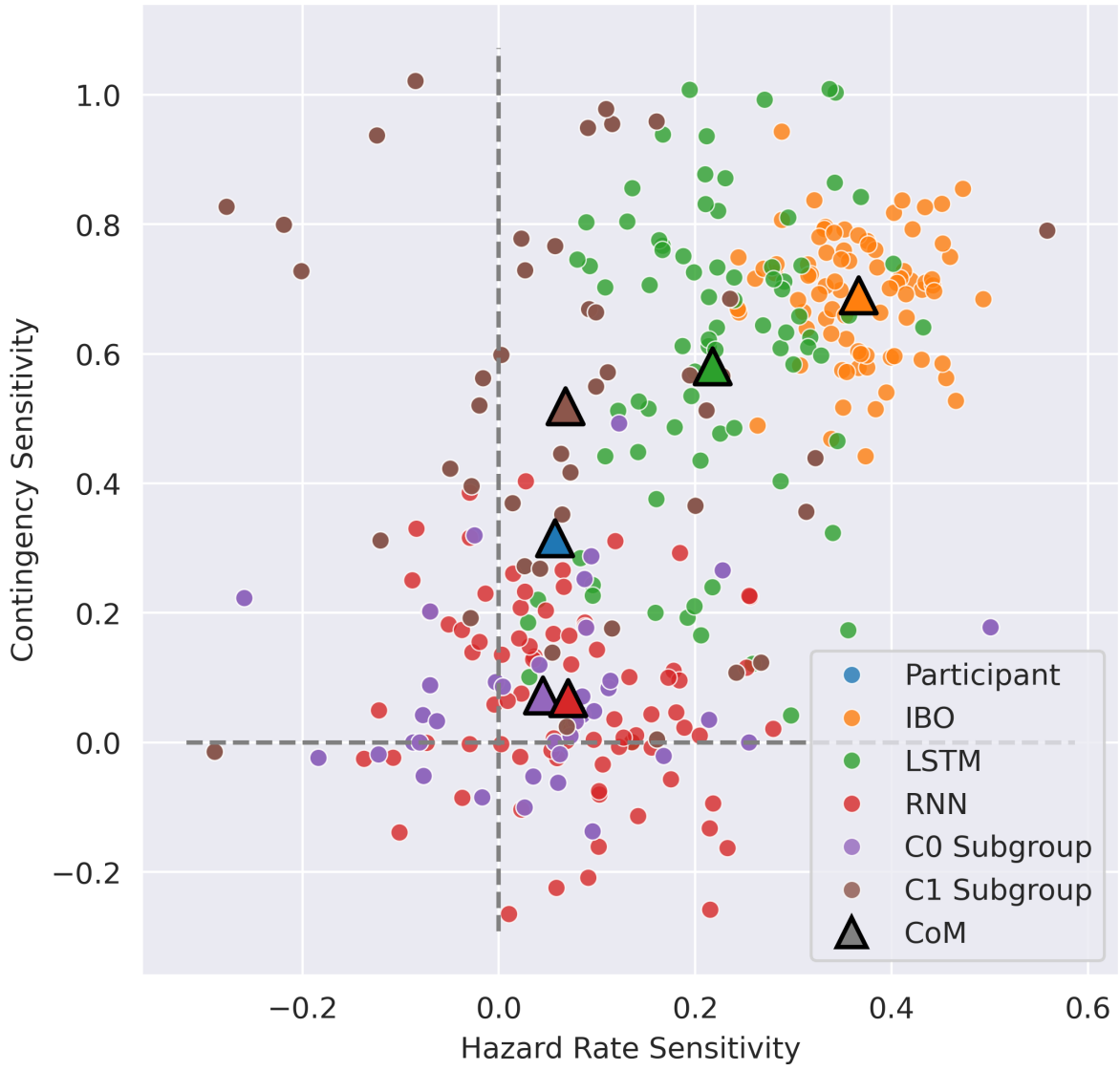


Figure 2.10: Hazard Rate vs Contingency Across Observers. The scatter plot displays observer performance on hazard rate and contingency sensitivity with center of mass (CoM) markers indicate average performance for each observer type. The Ideal Bayesian Observer (orange) defines the ceiling for performance and clusters at the highest end of both metrics. LSTM networks (green) show moderate hazard sensitivity (≈0.19) with high contingency sensitivity (≈0.53). C2 responders (purple) cluster at low hazard sensitivity (≈0.065) but high contingency sensitivity (≈0.52). C1 responders (brown) and vanilla RNN (red) both cluster near the origin with minimal sensitivity on both dimensions (≈0.05 hazard, ≈0.07 contingency). All participant (blue) CoM included for reference.

When comparing the subgroup sensitivities to that of the models, a clearer picture of correspondence emerges. The C1 group's hazard rate and contingency sensitivity entirely overlaps with the RNNs, statistically matching them in hazard rate and contingency. Whereas the C2 group statistically matched the RNNs in their hazard rate sensitivity, while statistically matching the LSTMs in their contingency sensitivity. Viewing all the observers' sensitivities on a single plot best illustrates these correspondences (fig. 2.10). The IBO naturally clusters in the top right due to its high sensitivity to both parameters. The RNNs and C1 responders occupy the space near the origin due to their low sensitivity to both task conditions. The C2 group clusters directly above, entirely along the axis of contingency sensitivity, and the LSTMs cluster directly to the right them, along the hazard the axis of hazard rate sensitivity. See fig. S3 to view the CWC curves for each subgroup.

Parameter Independence

Having shown that participants have asymmetrical learning capacities for hazard rate and contingency, we wondered if there was any correlation between them. The overall correlation between hazard rate and contingency sensitivities was moderate and significant ($r = 0.416$, $p < 0.001$), performing the correlation split by observer type revealed this was driven by between-group effects rather than within-agent covariation. We found that within-group analyses consistently yielded negligible correlations across all observer types:

Observer Type	**Correlation (r)**	**p-value**
Overall (all agents)	0.416	< 0.001
Participants (All)	0.025	0.831
Participants (C1)	0.129	0.454
Participants (C2)	-0.059	0.713
Ideal Bayesian Observer	0.177	0.120
LSTM networks	0.214	0.060
RNN networks	0.017	0.880

Table 2.4: Correlations between hazard rate and contingency sensitivities by observer type

2.3.5 Contingency Learning Generalization

Having established that C2 responders and LSTM networks achieve near-optimal contingency learning with wall bounces, we tested whether this learning reflects abstract rule acquisition or specific perceptual associations. The Random Velocity Change experiment tested whether participants' contingency learning reflected abstract rule acquisition or specific perceptual associations. By introducing random bounce trials where velocity changes occurred without wall contact, we could assess whether learning generalized beyond the specific features present during training.

Task Difficulty and Hazard Rate Learning: The addition of random bounce trials and parameter adjustments substantially increased task difficulty, particularly for hazard rate learning. While the Ideal Bayesian Observer maintained a significant, albeit lower, sensitivity to hazard rate changes ($M = 0.088 \pm 0.014$, $t(41) = 6.29$, $p < 0.001$), participants did not show significant sensitivity ($M = -0.022 \pm 0.016$, $t(41) = -1.38$, $p = 0.170$). This failure of hazard rate learning, and diminished sensitivity in the IBO, likely resulted from two factors: (1) the increased cognitive load from tracking three trial types, and (2) the reduced separation between low and high hazard rate conditions due to our parameter adjustments.

Contingency Learning

The control experiment tested whether participants learned abstract velocity-color contingencies or merely wall-specific associations. Despite the increased difficulty of random velocity changes, participants demonstrated significant contingency learning for both trial types. Wall bounce trials yielded a contingency sensitivity of 0.231 ± 0.059 ($t(41) = 3.918$, $p < 0.001$), while random bounce trials showed comparable sensitivity at 0.191 ± 0.055 ($t(41) = 3.501$, $p = 0.001$). Both values significantly exceeded chance, representing approximately 25% of optimal performance (24.5% for bounce, 21.0% for nonwall) as defined by the IBO.

Critically, the contingency sensitivity difference between trial types was not significant (mean difference = 0.040 ± 0.333, t(41) = 0.779, p = 0.441, d = 0.120), providing initial evidence for abstract contingent color change learning. Both trial types showed significantly greater sensitivity than straight path trials (wall vs. straight: p = 0.002, d = 0.568; random vs. straight: p = 0.003, d = 0.519), confirming that velocity changes of both types led to contingency learning.

Individual Differences

Individual difference analyses strengthened the case for abstract learning. Wall and random bounce sensitivities were strongly correlated (r = 0.592, p < 0.001), with wall bounce performance explaining 35% of the variance in random bounce learning (fig. 2.11). This substantial shared variance suggests a common underlying mechanism rather than separate learning systems for different velocity change types.

A median split analysis further confirmed this pattern. Participants in the top 50% split showed correspondingly high random bounce sensitivity (M = 0.356), while those in the bottom 50% split showed minimal random bounce sensitivity (M = 0.026; difference: t(40) = 3.30, p = 0.002). This consistency across individuals indicates that contingency learning ability generalizes across velocity change types.

To quantify the degree of generalization, we computed discrimination indices measuring how differently participants performed across trial types. Participants showed significantly less discrimination between wall and random bounce trials (M = 0.230) than between wall bounce and straight trials (M = 0.370; t(41) = -2.54, p = 0.015). The ratio of these discrimination indices (0.622) indicates good generalization, with participants treating wall and random bounces more similarly than they treat wall bounces and straight trials.

Although formal clustering was not performed on the smaller velocity change task videos, we examined participants showing response patterns consistent with the conservative and balanced strategies identified in the main experiment. C1-like responders (n=20)

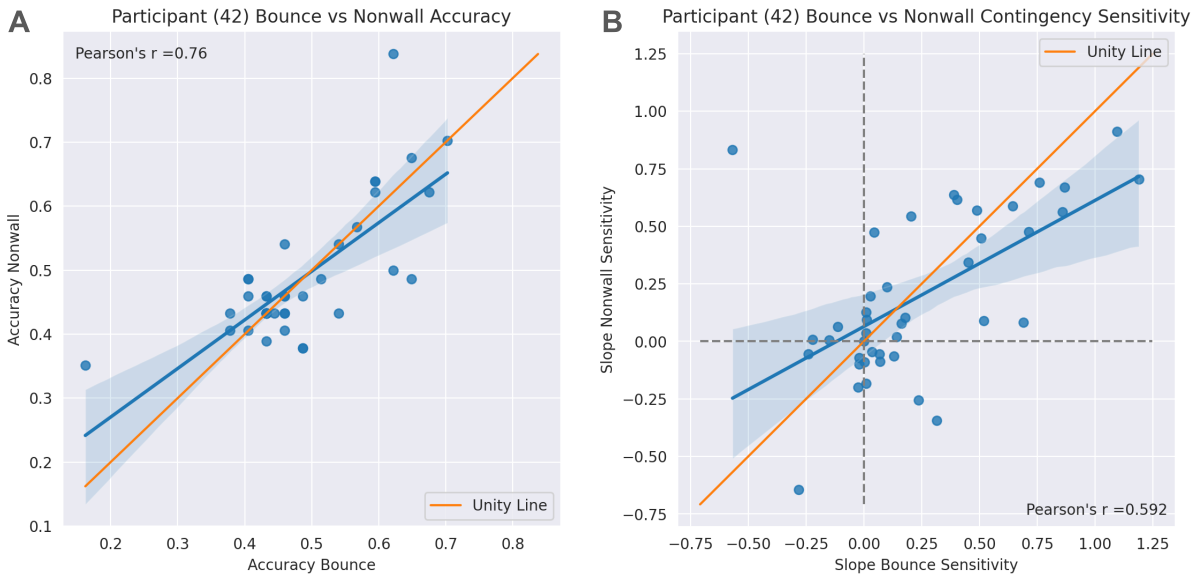


Figure 2.11: Nonwall Accuracy and Sensitivity Correlations. **(A)** Scatter plot of accuracy on bounce trials versus nonwall trials shows strong positive correlation (Pearson's $r = 0.76$), with individual performance clustered along the unity line (orange) indicating similar accuracy levels across both trial types. **(B)** Contingency sensitivity comparison between wall bounce and nonwall sensitivity shows strong correlation (Pearson's $r = 0.592$, $p < 0.001$), with wall bounce performance explaining 35% of variance in nonwall learning. Participants achieved sensitivity of 0.231 ± 0.059 for wall bounces and 0.191 ± 0.055 for random bounces, both significantly above chance (dashed line at 0).

showed minimal contingency sensitivity (M = 0.024, p = 0.340), while C2-like responders (n=22) demonstrated robust sensitivity (M = 0.420, $p < 0.001$). Importantly, both groups maintained consistent performance across wall and random trials, suggesting that strategy differences reflect stable individual traits rather than trial-specific adaptations.

Abstract In-Context Learning

The random velocity experiment demonstrates that participants learned abstract contingency rules rather than specific perceptual associations. Three key findings support this conclusion: (1) statistically equivalent sensitivity across trial types despite different velocity change sources, (2) strong correlation between individual performance on both trial types, and (3) consistent strategy patterns across conditions.

These results reveal that human contingency learning is not predicated on predictability and is observed even under randomness. Further, the poor hazard rate learning in this experiment further demonstrates the selective learning results in humans, with event-contingent relationships proving more robust to increased task complexity than event-free changes.

2.3.6 Summary of Key Findings

Five main findings emerged from our analyses:

1. **In-Context Learning Validation:** Bayesian inference makes specific and testable predictions for how color changes depend on context, for which we define sensitivity metrics. All observers showed statistically significant sensitivity to both hazard rate and contingency.

2. **Multiple Strategies**: Two human strategies were identified: C1 participants had a strong "no change" bias (91.4%), while C2 participants attempted inference despite only achieving modest classification performance. These strategies reflect fundamentally different task approaches where C1 responders achieved zero discrimination

ability (MCC = -0.001) despite 69.5% accuracy through base rate exploitation and C2 responders showed modest but positive discrimination (MCC = 0.021).

3. **Selective Parameter Learning**: Humans showed low sensitivity to probability tracking in the grayzone (15.5% of optimal performance) with no meaningful differences between strategies (C1: 13.6%, C2: 17.2%). However, C2 responders achieved near-optimal event-based contingency learning (78.9% of optimal), while C1 responders showed minimal learning (11.4%), comparable to the vanilla RNN.

4. **Abstract Contingency learning**: In the random velocity change experiment, participants demonstrated robust contingency learning that generalized across contexts. Despite increased task difficulty that diminished human hazard rate learning ($M = -0.022$, $p = 0.170$), participants showed statistically equivalent sensitivity to wall bounces (0.231 ± 0.059) and random velocity changes (0.191 ± 0.055), and strong individual correlation ($r = 0.592$, $p < 0.001$).

5. **Non-Correlative Sensitivity**: Correlation analyses revealed non-correlated mechanisms for temporal versus event-based learning across observers. While overall correlation between hazard and contingency sensitivities was significant ($r = 0.416$, $p < 0.001$), within-group correlations were uniformly non-significant for all observer types. This systematic absence of within-observer correlations suggests proficiency at the Bouncing Ball task may depend on independent systems for inference.

2.4 Discussion

Our investigation of in-context learning reveals selective learning capabilities in how humans and neural networks track environmental statistics. While all the observers were able to learn both task parameters ($p < 0.05$ for all observers), they demonstrate clear differences in their capacities for each (fig. 2.8). For hazard rate, the LSTM is the only observer that approached the performance of the IBO with 50.3% of optimality, whereas

all human groups and the RNN fell within the 13-22% of optimality range. However, for contingency sensitivity, we find that the C2 group of participants showed 78.9% of optimality, which is directly comparable to that of LSTM's 80.7%. Contrasting these two groups, the C1 responders reached 11.4% of optimality along with the RNN, which reached 10.2%. When examining the human participants as a whole, their optimality falls into a region directly in between C1 and C2.

The systematic nature of these behavioral outcomes becomes clearer when examined through the lens of the ideal Bayesian observer's (IBO). As defined, the IBO estimates hazard rate and contingency using conjugate Beta priors, as a way to perform evidence accumulation for both parameters independently. Because this approach is conserved between the two parameters, performance differences in estimation are most likely to come from what information successfully enters the accumulation system, rather than implementations of the system itself. Thus, assuming that an observer can become sensitive to one of the parameters, the selective sensitivity for one or the other is most likely not in the ability to accumulate evidence, but in the ability to properly count the relevant events. For humans, this indicates that a lower sensitivity to hazard rate is not due to difficulties with leveraging evidence accumulation as a computational tool, but that properly counting the relevant events is the difficulty.

2.4.1 Evidence Accumulation

Shared Mechanisms

The Ideal Bayesian Observer implements both hazard rate and contingency learning through Beta prior updating based on the observed counts. For hazard rate, the Beta distribution tracks the ratio of spontaneous color changes to eligible time steps (time steps with no velocity changes) For contingency, it tracks the ratio of color changes to velocity changes. This shared mechanism means that successful learning of either parameter demonstrates the capacity for evidence accumulation exists in that observer.

The performance differences between parameters must therefore originate not from the accumulation process itself, but from what an how events gets counted. This framework gives a different lens to understand the selective learning observed in the observers. It is not that they have different learning systems for different types of statistics, but rather that they have different abilities to extract countable events from a sequence.

Diagnosing Computational Capacity

Each observer's sensitivity pattern provides diagnostic information about their computational capabilities. Starting with the RNN, it's low sensitivity to both parameters (10.2% contingency, 21.5% hazard rate) leaves ambiguous whether it lacks the accumulation machinery entirely or simply cannot properly count, or both. In contrast, the LSTM's sensitivity to both parameters suggests it can both properly accumulate evidence and count the relevant events for both parameter. Indeed, examining the differences between the trained seeds shows that hazard rate estimation is a much easier task for the model, reaching it's final sensitivity within 5% of training epochs (fig. 2.12A). However, a different pattern emerges for contingency. The models all reach the same levels of contingency sensitivity as the humans or RNN within the first 5% of total training epochs, and they initially remain there. It is only through continued training that some of the models undergo some representational change where their sensitivity rapidly reaches the observed final values. Further, some of the models never experienced this transition and exhibit contingency sensitivity comparable to the participants and RNN (n=3/10) (fig. 2.12B). This variable and delayed sensitivity to contingency relative to hazard rate suggests that even with the architectural machinery to perform evidence accumulation, counting certain types of events remains challenging.

In the participants, a similar but reversed story can be formed. C2 responders' near-optimal contingency learning (78.9%) indicates that they posses the ability to properly perform evidence accumulation and properly count contingent color changes. This is best

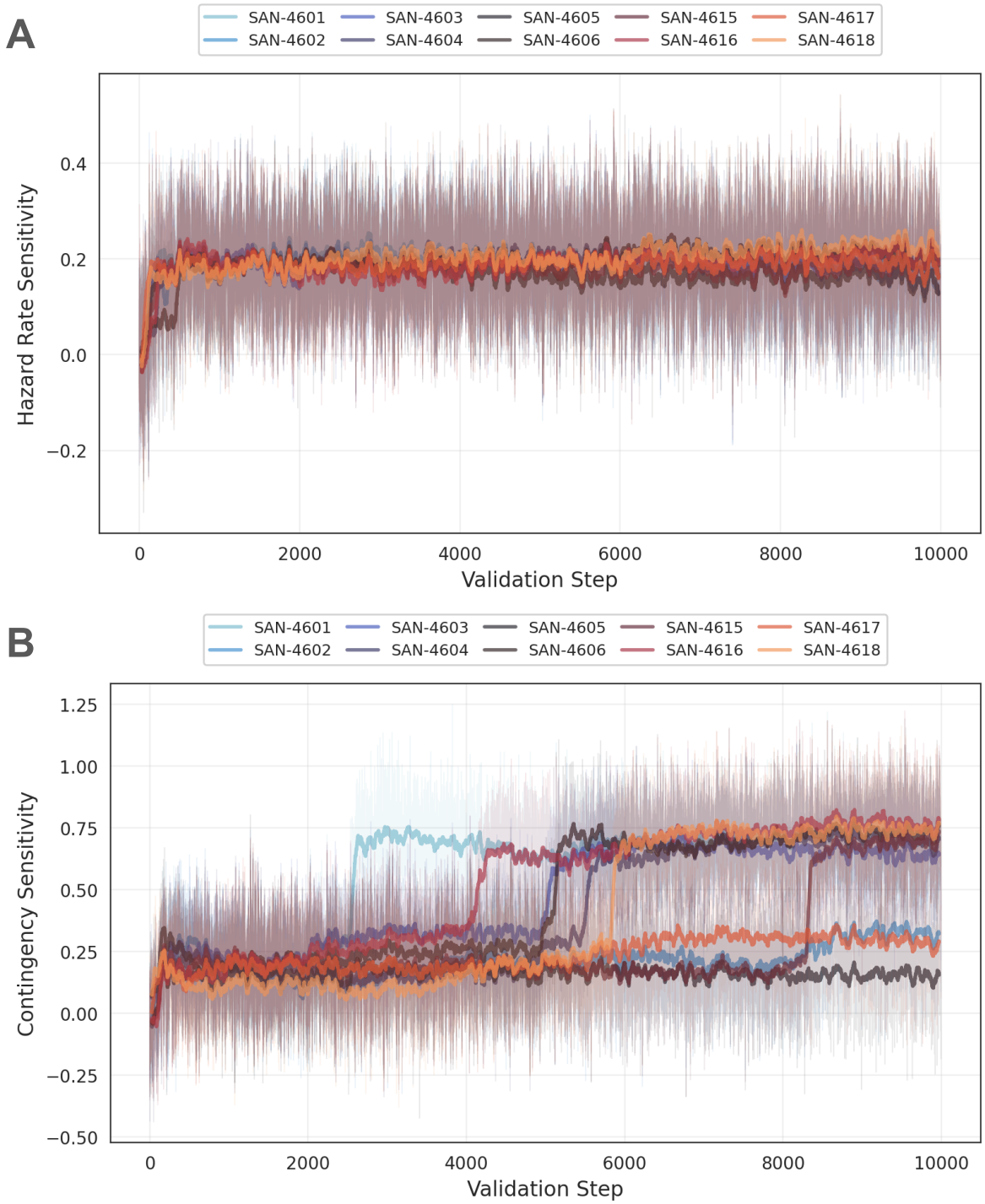


Figure 2.12: LSTM Sensitivity During Training. Participant task videos were used as a validation set to compute hazard rate and contingency sensitivity as a function of training for all 10 model seeds. **(A)** Hazard rate sensitivity for all models shows robust sensitivity learning. All models reach peak sensitivity within 10% of training. **(B)** Contingency sensitivity for all models shows more difficulty to reach peak sensitivity. All models initially reach RNN contingency baseline within first 10% of training, but 3/10 do not improve. Models that reach peak sensitivity do so at variable points during training.

shown in their near-optimal accuracy on the bounce trials. This indicates that both C2's poor hazard rate performance (17.2%) and C1's poor performance on both parameters likely stem from counting difficulties rather than computational incapacity.

2.4.2 The Counting Problem

Counting for Contingency Sensitivity

Understanding why observers succeed or fail at each parameter requires examining what must be counted for accurate learning. For contingency learning, the quantities that must be tracked are bounces with color changes and bounces without color changes, both of which, typically number only 1-10 per trial (fig. 2.5F, H). For the LSTM, this creates a low-sample scenario that explains the LSTM's variable performance. With so few observations relative to the number of time steps within a trial, the evidence accumulation process has limited data to work with. Yet for humans, each bounce represents a discrete, salient event that naturally segments experience into countable units. The data sparsity that challenges LSTMs plays to human strengths in event-based attention and memory.

Counting for Hazard Rate Sensitivity

Hazard rate presents the opposite challenge, where the two quantities that need to tracked are the number of eligible time steps where no color change occurs and the number of random color changes, which is typically 1-10 per trial (fig. 2.5B). This creates a high signal-to-noise ratio since one of the two values to count rapidly increases, which the LSTM can leverage to quickly converge on accurate estimates. However, for humans, tracking hundreds of "non-event" time steps presents a fundamental challenge. These eligible time steps accumulate as a continuous, undifferentiated stream rather than discrete, countable units.

The low sensitivity to rate learning in participants cannot be attributed to difficulty detecting color changes themselves. Spontaneous color changes are perceptually identical

to contingent color changes, and participants clearly detect them (as evidenced by their successful contingency learning when these changes follow bounces). Similarly, the velocity changes that trigger contingent color changes involve minimal visual displacement at the moment of change, yet participants track them successfully. Therefore, since counting color changes, the core difficulty for participants must be in tracking eligible time steps. These non-events accumulate continuously, requiring participants to maintain their attention to count the passage of time. Even the C2 responders, who demonstrate motivation and capability through their contingency learning, achieve only 17.2% optimal hazard rate performance, suggesting a cognitive limitation rather than a lack of effort.

2.4.3 Differing Counting Abilities

Model Correspondence

By framing the task difficulty around counting estimation, the similar performance of C1 and RNNs masks fundamentally different underlying causes. For hazard rate, C1 would exhibit the same issues faced as C2 and for the same reasons. Whereas for contingency, C1's behavioral performance reflects their no-change strategy rather than a fundamental inability to perform evidence accumulation. The RNN's poor performance, however, likely indicates either inability to implement evidence accumulation or inability to count properly, or both. This indicates that from a computational standpoint, there may be fundamentally different processes (or lack of them) that lead to the same sensitivity. However, the correspondence between C2 and LSTM performance on contingency (78.9% vs 80.7%) likely reflects a shared computational process. Both demonstrate evidence accumulation circuitry, both can count discrete events, and both achieve near-optimal performance when these capabilities align with task demands.

Non-Correlative Sensitivities

Our finding that hazard rate and contingency sensitivities show no within-observer correlation can be understood through the same framework as above. While both parameters use the same evidence accumulation machinery, they require different counting abilities. And although a suboptimal evidence accumulation strategy could theoretically create correlated deficits across parameters, for the Bouncing Ball task, the counting requirements for each parameter are distinct enough that it behavioral differences will most reflect differences in counting abilities. This independence suggests that in-context learning abilities are likely not unitary but composed of separable components that can vary independently across individuals and systems.

Abstract Contingency Learning

The success of participants in learning contingencies from random velocity changes provides strong support for the counting-based framework. Participants showed statistically equivalent sensitivity to the bounce trials and nonwall trials with both significantly exceeding chance. Viewing it through the limitations of counting, since random velocity changes remain discrete, they are similarly countable events.

2.4.4 Strategy Heterogeneity

Both C1 and C2 responders faced the same challenges in counting where they are equally unable to track continuous non-events for hazard rate learning. The C2 group's active engagement with the task shows that despite clear motivation to learn, they achieve only 17.2% optimal hazard rate performance. Faced with this limitation in combination with task videos where 67% of trials involve no color change, the C1 participants adopt a "no-change" strategy. Under the current framing of the task to the participants, such a strategy is not only viable, but leads to a higher accuracy overall (0.695 bal. acc.) on the task relative to the C1 group (0.637 acc).

2.4.5 Limitations and Future Directions

In our work, the behavioral outcome distinctions between the hazard rate and contingency emerge out of the different counting needs. Our instantiation of the Bouncing Ball task only tests one such combination of counting demands, where we found an asymmetry in well participants can perform each of them. However, several potential limitations to this understanding within and outside the framework of the Bouncing Ball task.

As designed, the task's crucial mechanic, the grayzone occluding color in the middle third of the arena, may not be the most ecologically relevant timescale to probe human temporal counting. Even under more conservative video generation parameters, the ball only remains occluded for 1-2 seconds. Thus, the source of the asymmetry may be that this brief period, isn't long enough to investigate temporal counting as most real-world decisions require integration across minutes, hours, days, and seasons. This would also explain why event-based counting remains intact as discrete events which require immediate behavioral adaptation aligns much more with human experience. From visually salient changes like the the traffic light changing to green, to noticing a sudden turn in a conversation, we encounter this form of in-context learning constantly and across abstraction levels. An experimental paradigm testing temporal integration might reveal that humans do optimality integrate time, but over more naturalistic timescales such as minutes to hours. However, there is likely a minimum window of integration, and so a potential future direction would be to investigate the effects of using a rectangular stage and a longer grayzone. The experimental paradigm would likely need to be different, but under slightly longer timescales, humans might exhibit more optimal sensitivity.

Another limitation that goes beyond the current framework is that while we sought to investigate in-context learning of multiple latent factors, our design has only explored this for two, whereas humans have to perform in-context learning many features concurrently, with varying levels of interdependence. We explored the degree to which people are sensitive to when a variable changes randomly, and when it changes in relation to another

discrete feature. From here there are several axes of further exploration. Along the scaling axis, questions surrounding the capacity limits for concurrent changes can be raised. Using the same experimental paradigm, we could introduce a second ball with independently sampled statistics, and using the same video generation process (generate sequence in reverse to set final position), participants would need to learn both concurrently.

An orthogonal axis of exploration would be along the chain of causal changes. In our current framework, color changes are caused by velocity changes with a fixed probability for each sequence that is chosen randomly for each video. However, naturalistic sequence have multiple layers of change dependence, that are interconnected in multiple ways. The introduction of an additional parameter that controls the statistics of the causal relationship between features would explore the degree to which humans discover hierarchies of interdependence and causation. A simple way to test this within the current framework would to introduce another feature, such as the sequence direction (R→G→B vs R→B→G), which then controls the contingency.

2.4.6 From Behaviors to Mechanisms

Having identified counting as the critical bottleneck for in-context learning on the Bouncing Ball task, we next turn to understanding how these counting operations are implemented in neural networks. The evidence accumulation process works effectively when appropriate counts can be extracted, but how do neural circuits actually perform this extraction? In the following chapter, we examine the neural circuits that successfully learn these parameters, seeking to understand how sparse networks solve the counting problem. Understanding these mechanisms may ultimately explain why counting discrete events differs from tracking continuous streams, and why some networks succeed where others fail.

References

De Freitas, J. and Alvarez, G. A. (2018). Your visual system provides all the information you need to make moral judgments about generic visual events. *Cognition*, 178:133–146.

Efron, R. (1967). The Duration of the Present. *Annals of the New York Academy of Sciences*, 138(2):713–729. _eprint: https://nyaspubs.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1749-6632.1967.tb55017.x.

Gamboa, J. C. B. (2017). Deep Learning for Time-Series Analysis. arXiv:1701.01887 [cs].

Ghosh, A., Eldar, Y. C., and Chatterjee, S. (2025). Semi-Supervised Model-Free Bayesian State Estimation from Compressed Measurements. arXiv:2407.07368 [eess].

Goroshin, R., Bruna, J., Tompson, J., Eigen, D., and LeCun, Y. (2015). Unsupervised Learning of Spatiotemporally Coherent Metrics. arXiv:1412.6056 [cs].

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Kim, M. (2011). Discriminative semi-supervised learning of dynamical systems for motion estimation. *Pattern Recognition*, 44(10):2325–2333.

Loshchilov, I. and Hutter, F. (2019). Decoupled Weight Decay Regularization. arXiv:1711.05101 [cs].

Prat-Carrabin, A., Wilson, R. C., Cohen, J. D., and Azeredo da Silveira, R. (2021). Human inference in changing environments with temporal structure. *Psychological Review*, 128(5):879–912. Place: US Publisher: American Psychological Association.

Radvansky, G. A., Copeland, D. E., Berish, D. E., and Dijkstra, K. (2003). Aging and Situation Model Updating. *Aging, Neuropsychology, and Cognition*, 10(2):158–166. Publisher: Routledge _eprint: https://doi.org/10.1076/anec.10.2.158.14459.

Swallow, K. M., Zacks, J. M., and Abrams, R. A. (2009). Event Boundaries in Perception Affect Memory Encoding and Updating. *Journal of experimental psychology. General*, 138(2):236.

Wang, Y., Yin, Y., Suryanarayana, K. S. N., Drgona, J., Schram, M., Halappanavar, M., Liu, F., and Li, P. (2023). Semi-Supervised Learning of Dynamical Systems with Neural Ordinary Differential Equations: A Teacher-Student Model Approach. arXiv:2310.13110 [cs].

CHAPTER 3

Computational Mechanisms of In-Context Learning

3.1 Introduction

In the previous chapter, we found that LSTM networks achieve the highest hazard rate sensitivity (50.3% of optimal) and near-optimal contingency sensitivity (80.7% of optimal), matching or exceeding human performance on both statistics. In contrast, vanilla RNNs showed minimal learning on both dimensions (21.5% hazard, 10.2% contingency), performing comparably to humans only on hazard rate where they struggle potentially due to counting limitations. This performance difference makes the LSTM the natural focus of our next investigation, since our goal is to understand which mechanisms enable successful in-context learning of both statistics.

Of interest in this study is what these networks actually represent in their states to perform the task. The LSTM architecture processes information through a 32-dimensional state space comprising hidden and cell states. These states could encode task parameters through distributed activity (Tinsley, 2008), similar to cortical populations in the brain, or they might develop specialized sub-circuits in which specific units track specific statistics

(Liu et al., 2020). Previous work examining trained networks found that task-relevant information often concentrates into sparse subsets of units (Bakaraju and Konda, 2019; Liu et al., 2020; Arpit et al., 2016; Gupta et al., 2023), though this phenomenon has not been explored for problems requiring simultaneous tracking of multiple task-relevant variables with different tracking requirements.

3.1.1 Neural Integration as a Computational Primitive

If such specialized units do emerge, they would need mechanisms for evidence accumulation across time, similarly to that of the IBO. Neural integrators provide an established solution for this problem that has been studied in biological and artificial settings (Goldman et al., 2009; Cain et al., 2013; Sanchez and Rowe, 2018). Mechanistically, these circuits implement a "leaky integrator" by accumulating inputs over time while discounting older information via a static decay. Additionally, this prevents the circuit response from saturating, keeping it sensitive to new inputs.

The mathematics of leaky integration fit naturally within the Bayesian framework of our task. The beta distributions of the ideal Bayesian observer must be updated as new observations are encountered, effectively counting the relevant events as they occur. A leaky integrator can directly approximate this process by incrementing with each observable while maintaining a decay. The result of this process would be a running weighted-average of whichever variable is being tracked. This can be directly applied to the hazard rate via incrementing on a random color change while maintaining a steady decay. For contingency, integration would need to be triggered by velocity changes with a smaller leak term.

3.1.2 Architectural Considerations

The LSTM forms the primary model of investigation for this study and carries out specific gating processes on its hidden and cell states. The details of how these gates

are formed and which state they act upon creates specific constraints in the way the integration could be implemented. The forget gate directly controls the flow of information out of the cell state, potentially implementing the "leak" for the integrator. Whereas the input gate controls the flow of information into the cell state, potentially implementing when the count needs to be incremented. The cell state itself serves as the memory that would be needed to keep track of the integration. Finally, the output gate constructs the hidden state from cell state, before being passed forward to make a prediction. These architectural considerations suggest cell states could contain the causal representations driving a leaky-integrator process, where the hidden state functions as a readout.

3.1.3 Approaches to Mechanistic Understanding

Three complementary approaches are used to probe how these computations are implemented in recurrent networks. An elastic net regularization pipeline systematically identifies which units are necessary for decoding each parameter by progressively enforcing sparsity constraints. Examining the temporal dynamics of these units during color changes and velocity changes reveals the computational strategy they implement. And finally, targeted interventions that manipulate unit activities establish whether identified correlations reflect causal control over behavior.

Several specific questions guide this investigation. First, whether the networks encode hazard rate and contingency through overlapping or independent neural populations. Second, if these representations concentrate in sparse subsets or require the full network capacity. Third, what computational operations these units implement to track their respective statistics. Fourth, whether manipulating these units produces behavioral changes consistent with their hypothesized function.

This chapter systematically addresses these questions by first identifying critical units through regularization analysis. We then characterize how these units respond to task-relevant events across different parameter conditions. Finally, we apply targeted state

injections during gray zone traversals to establish causal relationships between neural activity and color predictions. This progression from correlation through mechanism to causation reveals the minimal neural circuits sufficient for multi-timescale in-context learning, while also establishing predictions for future neurophysiological experiments seeking to identify analogous mechanisms in biological systems.

3.2 Methods

To investigate the potential computational mechanisms used by participants on the Bouncing Ball task, we used recurrent neural network models trained on a modified version of the human participant dataset. We detail the several changes that were made to effectively train the networks in Section 2.2.9, but briefly:

- We set the rate-limiting parameters τ_{dc} and τ_{dv} to 0.
- Trained on a linear spacing of hazard rates, contingencies, and velocities instead of using the discrete values for the participants.

In this section, we will first go through the models used and their training procedure in detail before going through the analysis techniques.

3.2.1 Sequence Modeling

The sequence model we used for the Bouncing Ball task was a three module network comprising a feedforward block, a recurrent module, and an output layer. The feedforward block and output layer were conserved across all experiments, whereas the recurrent block comprised either an RNN or an LSTM blocks. We trained 10 identical model seeds for each of the RNN and LSTM sequence models to ensure our results reflect architectural differences rather than the stochasticity of any specific seed. All model sizes were constrained to have 1.8K parameters.

Feedforward Block: The feedforward block maps the input features (dimensionality

D_{in}) to a higher-dimensional representation (dimensionality D_{ff}). For all experiments, the input features are $D_{in} = 5$, corresponding to (x, y, c_1, c_2, c_3) where (x, y) are the coordinates of the ball and c_i are each color channel, and the feedforward block size was set to $D_{ff} = 8$. In its simplest form, this block consists of a linear transformation followed by ReLU activation:

$$\mathbf{y}_{ff}^{(t)} = \text{ReLU}(\mathbf{W}_{ff}\mathbf{x}_{in}^{(t)} + \mathbf{b}_{ff}) \tag{3.1}$$

where $\mathbf{W}_{ff} \in \mathbb{R}^{D_{ff} \times D_{inp}}$ and $\mathbf{b}_{ff} \in \mathbb{R}^{D_{ff}}$ are the feedforward weight matrix and bias vector, respectively.

Recurrent Module: The recurrent module processes the sequence using one of the recurrent modules described below with a hidden dimensionality of D_{rec} units. The exact values of D_{rec} differ between the LSTM and RNN to match the total number of parameters between the models. The contents of the outputs are specific to each module, but in general, for sequences of length T, they take inputs and compute outputs in the following form:

$$\mathbf{y}_{rec}^{(t)}, \mathbf{H}_{rec}^{(t)} = \text{Recurrent}(\mathbf{y}_{ff}^{(t)}, \mathbf{H}_{rec}^{(t-1)}) \tag{3.2}$$

where $\mathbf{H}_{rec}$ is either a single vector in $\mathbb{R}^{D_{rec}}$, or a tuple of L vectors, each in $\mathbb{R}^{D_{rec}}$:

$$\mathbf{H}_{\text{rec}} \in \mathbb{R}^{D_{rec}} \cup \left(\mathbb{R}^{D_{rec}}\right)^L \tag{3.3}$$

Output Layer: The output layer transforms the recurrent outputs (dimensionality D_{rec}) into final predictions (dimensionality D_{out}) through a single linear layer, followed by a softmax activation:

$$\mathbf{y}_{out}^{(t)} = \text{Softmax}(\mathbf{W}_{out}\mathbf{y}_{rec}^{(t)} + \mathbf{b}_{out}) \tag{3.4}$$

where $\mathbf{W}_{out} \in \mathbb{R}^{D_{out} \times D_{rec}}$ and $\mathbf{b}_{out} \in \mathbb{R}^{D_{out}}$ are the output weight matrix and bias vector, respectively. For all experiments, the output dimensionality was set to $D_{out} = 3$, corresponding to the prediction probability for each color.

Recurrent Modules

Vanilla Recurrent Neural Network (RNN): We used an Elman recurrent neural network (RNN) as the simplest recurrent module for the sequence model (Elman, 1990). The RNN architecture consists of a single recurrent layer that processes the output sequence of the feedforward block, $\mathbf{y}_{ff} = \{\mathbf{y}_{ff}^{(1)}, \ldots, \mathbf{y}_{ff}^{(T)}\}$, iteratively, maintaining an internal hidden state that captures temporal information across time steps. At each timestep t, the hidden state $\mathbf{H}^{(t)}$ is computed as:

$$\mathbf{H}_{rec}^{(t)} = \tanh(\mathbf{W}_{yH}\mathbf{y}_{ff}^{(t)} + \mathbf{W}_{HH}\mathbf{H}_{rec}^{(t-1)} + \mathbf{b}_h) \tag{3.5}$$

where $\mathbf{W}_{yh} \in \mathbb{R}^{D_{ff} \times D_{rec}}$ is the input-to-hidden weight matrix, $\mathbf{W}_{HH} \in \mathbb{R}^{D_{rec} \times D_{rec}}$ is the recurrent weight matrix connecting the previous hidden state to the current hidden state, $\mathbf{b}_h \in \mathbb{R}^H$ is the bias vector, D_{rec} is the hidden state dimensionality, and D_{ff} is the feedforward dimensionality. For all experiments using the RNN, D_{rec} was set to 34 units, leading to 1.8K parameters in the full sequences model.

Substituting the RNN into the recurrent module in Equation (3.2), the activities passed to the output layer are the hidden state at timestep t:

$$\mathbf{H}_{rec}^{(t)}, \mathbf{H}_{rec}^{(t)} = \text{RNN}(\mathbf{y}_{ff}^{(t)}, \mathbf{H}_{rec}^{(t-1)}) \tag{3.6}$$

Long Short-Term Memory Network (LSTM): The primary recurrent module we used on the Bouncing Ball sequence data was the Long Short-Term Memory (LSTM) network. The LSTM addresses the vanishing gradient problem by implementing gating mechanisms and a separate cell state, allowing it to retain dependencies across extended temporal sequences (Hochreiter and Schmidhuber, 1997).

The LSTM cell computes three gates to generate its outputs: forget, input, and output gates ($\mathbf{f}$, $\mathbf{i}$, $\mathbf{o}$). At each timestep t, they are generated as follows:

$$\mathbf{f}^{(t)} = \sigma(\mathbf{W}_f \cdot [\mathbf{H}^{(t-1)}, \mathbf{y}_{ff}^{(t)}] + \mathbf{b}_f) \tag{3.7}$$

$$\mathbf{i}^{(t)} = \sigma(\mathbf{W}_i \cdot [\mathbf{H}^{(t-1)}, \mathbf{y}_{ff}^{(t)}] + \mathbf{b}_i) \tag{3.8}$$

$$\tilde{\mathbf{C}}^{(t)} = \tanh(\mathbf{W}_C \cdot [\mathbf{H}^{(t-1)}, \mathbf{y}_{ff}^{(t)}] + \mathbf{b}_C) \tag{3.9}$$

$$\mathbf{C}^{(t)} = \mathbf{f}^{(t)} \odot \mathbf{C}^{(t-1)} + \mathbf{i}^{(t)} \odot \tilde{\mathbf{C}}^{(t)} \tag{3.10}$$

$$\mathbf{o}^{(t)} = \sigma(\mathbf{W}_o \cdot [\mathbf{H}^{(t-1)}, \mathbf{y}_{ff}^{(t)}] + \mathbf{b}_o) \tag{3.11}$$

$$\mathbf{H}^{(t)} = \mathbf{o}^{(t)} \odot \tanh(\mathbf{C}^{(t)}) \tag{3.12}$$

where σ denotes the sigmoid function, $\odot$ represents element-wise multiplication, $\mathbf{W}_f, \mathbf{W}_i, \mathbf{W}_C, \mathbf{W}_o \in \mathbb{R}^{D_{rec} \times (D_{ff} + D_{rec})}$ are the weight matrices for their respective gates, and $\mathbf{b}_f, \mathbf{b}_i, \mathbf{b}_C, \mathbf{b}_o \in \mathbb{R}^{D_{rec}}$ correspond to their bias vectors. The notation $[\mathbf{H}^{(t-1)}, \mathbf{y}_{ff}^{(t)}]$ indicates concatenation of the previous hidden state and current input. For all experiments using the LSTM, D_{rec} was set to 15 units, leading to 1.8K parameters in the full sequences model.

Substituting the LSTM into the recurrent module in Equation (3.2), the activities passed to the output layer is the hidden state at timestep t, and the outputted states are a tuple of the hidden and cell states:

$$\mathbf{H}_{rec}^{(t)}, (\mathbf{H}_{rec}^{(t)}, \mathbf{C}_{rec}^{(t)}) = \text{LSTM}(\mathbf{y}_{ff}^{(t)}, \mathbf{H}_{rec}^{(t-1)}, \mathbf{C}_{rec}^{(t-1)}) \tag{3.13}$$

Training Procedure

Models were trained using the semi-supervised learning framework where, at each timestep t, the network receives observable state $\mathbf{o}^{(t)} = (\mathbf{x}^{(t)}, \tilde{\mathbf{c}}^{(t)})$, and predicted the color at the next timestep $\hat{\mathbf{y}}^{(t+1)}$, where $\mathbf{x}^{(t)}$ is the $(x^{(t)}, y^{(t)})$ coordinates and $\tilde{\mathbf{c}}^{(t)}$ is the potentially masked color of the ball. The loss function was computed as the cross-entropy

between predicted and true color distributions at all time steps:

$$\mathcal{L} = -\frac{1}{NT} \sum_{n=1}^{N} \sum_{t=1}^{T-1} \sum_{k=0}^{2} y_{n,k}^{(t+1)} \log \hat{y}_{n,k}^{(t+1)} \tag{3.14}$$

where N is the batch size, T is the sequence length, and $y_{n,k}^{(t+1)}$ is the scalar entry of the one-hot encoded label vector $\mathbf{y}_n^{(t+1)} \in \{0,1\}^3$ for a video n and timestep $t+1$, which is defined element-wise by:

$$\left[\mathbf{y}_n^{(t+1)}\right]_k = \mathbb{I}(\mathbf{c}_n^{(t+1)} = k), \quad \text{for } k \in \{0,1,2\} \tag{3.15}$$

so that $y_{n,k}^{(t+1)} = \left[\mathbf{y}_n^{(t+1)}\right]_k$, and $y_{n,k}^{(t+1)}$ is the predicted probability for color $k \in \{0,1,2\}$ (red, green, blue). For all experiments, the batch size was set to $N = 1024$, and the sequence length was set to $T = 600$.

Truncated Backpropagation Through Time

Given the extended sequence length of 600 time steps, we employed truncated backpropagation through time (TBPTT) to manage computational resources and gradient stability (Werbos, 1990). In TBPTT, sequences are divided into non-overlapping segments for gradient computation while maintaining hidden states across segment boundaries. During forward propagation, the hidden state flows continuously through the entire sequence. However, gradients are only backpropagated through a predefined truncated window, which we set to 150 time steps for our experiments, dividing each 600-timestep sequence into four segments. This approach balances the need to capture long-range dependencies with computational efficiency and helps mitigate the vanishing gradient problem inherent in training RNNs on long sequences. When combined with the number of epochs (10,000) and batches per epoch (50), this gives a total of 2,000,000 gradient updates for each training run. In total, we trained 10 independent model seeds with different random initializations.

Learning Rate Scheduling

We implemented a Cosine Annealing with Warm Restarts (CAWR) learning rate scheduler, which was found to significantly accelerate training convergence (Smith, 2017). The learning rate at iteration t is computed as:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min}) \left(1 + \cos\left(\frac{T_{cur}}{T_i}\pi\right)\right) \tag{3.16}$$

where $\eta_{min} = 0$ and $\eta_{max} = 0.001$ are the minimum and maximum learning rates, T_{cur} is the number of iterations since the last restart, and T_i is the current cycle length. Our implementation used $T_0 = 20000$ iterations for the initial cycle length and $T_{mult} = 1$, meaning each cycle maintained the same duration. This configuration results in learning rate restarts every 20000 gradient updates (100 epochs), with gradual cosine annealing between restarts. The warm restarts help escape local minima by periodically increasing the learning rate while enabling rapid exploration of the loss landscape after each restart.

Evaluation Datasets

For model evaluation, we employed two sets of task videos, each serving complementary roles in assessing model performance and behavior.

Participant Dataset: The original human participant task videos served as our primary validation set throughout training and enabled direct comparison between model and human behavior. This task videos comprised 168 trials following the exact structure described in the human experiments: 81 straight path trials, 76 bounce path trials, and 11 attention check trials. By evaluating models on the identical sequences shown to human participants, we could assess whether models developed similar sensitivities to hazard rate and contingency parameters. During training, model performance on this validation set was monitored every 50 epochs to track learning progress and prevent overfitting. The validation metrics included overall color prediction accuracy across the trial types,

hazard rate sensitivity ($\hat{s}_{hz}$), and contingency sensitivity ($\hat{s}_{cont}$) computed using the same confidence-weighted choice framework applied to human data.

Extended Dataset: While the human participant task videos provided ecological validity, its limited size constrained our ability to probe model behavior across the full space of possible state configurations and transition sequences. To address this limitation, we generated an extended evaluation task videos comprising 20,000 videos with identical statistical parameters to the participant task videos (including the rate-limiting variables). One change we made for the bounce trials was to set the number of time steps before the bounce to be 1, whereas in the original participant task videos it was set to 5. By making this change, we can better isolate the effects of hazard rate on the bounce trials by minimizing the amount of time before the event of interest (wall bounces). This was set for participants to mitigate potential perceptual limitations that are not present in the models. By maintaining identical statistical properties between the validation and extended task videos, we ensure that insights gained from the extended task videos' analyses would generalize to the ecological conditions under which humans performed the task.

3.2.2 Identifying Critical Units

To understand how recurrent networks encode task-relevant information, we employed a systematic approach to identify units whose activity patterns reliably track latent task variables. This analysis relied primarily on elastic net regularization to reveal sparse representations within the recurrent hidden state space(s) (Zou and Hastie, 2005).

State Extraction and Preparation

After training models using the procedure described above, we evaluated each model on the extended evaluation task videos (20,000 videos) and extracted complete state trajectories. For each timestep t in every video, we recorded both the hidden states $\mathbf{H}^{(t)} \in \mathbb{R}^{D_{rec}}$ and cell states $\mathbf{C}^{(t)} \in \mathbb{R}^{D_{rec}}$ from the LSTM architecture. These states were

concatenated to form a combined representation $\mathbf{S}^{(t)} = [\mathbf{H}^{(t)}; \mathbf{C}^{(t)}] \in \mathbb{R}^{2D_{rec}}$ and each unit was standardized using z-score normalization across all videos and time steps:

$$\mathbf{Z}^{(t)} = \frac{\mathbf{S}^{(t)} - \boldsymbol{\mu}_u}{\boldsymbol{\sigma}_u} \tag{3.17}$$

where $\boldsymbol{\mu}_u$ and $\boldsymbol{\sigma}_u$ are the unit activity mean and standard deviation across the task videos.

Decoding Analysis with Elastic Net Regularization

To identify units encoding specific task variables, we trained separate linear decoders to predict each latent parameter from the neural states. For each task variable of interest (hazard rate and contingency), we employed elastic net regularization, which combines $L1$ and $L2$ penalties:

$$\mathcal{L}_{elastic} = \mathcal{L}_{task} + \lambda \left[\alpha ||\beta||_1 + \frac{1-\alpha}{2} ||\beta||_2^2 \right] \tag{3.18}$$

where $\mathcal{L}_{task}$ is the task-specific loss (logistic loss for classification, squared error for regression), β represents the decoder weights, λ controls overall regularization strength, and α balances between $L1$ and $L2$ penalties. For hazard rate, we performed binary classification (high vs. low) with $\alpha = 0.64$. For contingency, we used regression with $\alpha = 0.4$ to predict continuous contingency values, then discretized the predictions into classes by assigning each prediction to the nearest ground-truth value (0.05 for low, 0.5 for medium, 0.95 for high). The key insight of this approach is that, as regularization strength λ increases, the $L1$ component increasingly drives coefficients to zero, revealing sparser subsets of units that encode each task variable. We systematically varied λ across seven orders of magnitude (10^0 to 10^{-6}) using logarithmic spacing, training separate decoders at each regularization level.

Identifying Critical Units

For each regularization strength, we recorded both the decoding performance (accuracy for classification tasks, R^2 for regression) and the resulting coefficient vectors $\beta(\lambda)$. This

produced performance curves showing how prediction quality degraded with increasing sparsity, and coefficient paths showing which units retained non-zero weights under strong regularization. Units were identified as critical units if they maintained non-zero coefficients at regularization strengths just before performance dropped to chance. This approach identifies units that not only correlate with task variables but are necessary for maintaining predictive accuracy when representational capacity is constrained. The identified critical units could then be further probed to understand their role in the overall network computation. Additionally, by applying this analysis to hidden and cell states together, we could additionally examine whether task-relevant information was differentially encoded in the LSTM's memory versus output pathways.

3.2.3 Neural Tuning Profiles

Having identified task-relevant units through regularization analysis, we next characterized their tuning profiles through systematic probing experiments.

Whole Dataset and Single-Trial Activity

To understand how task-relevant units organize information across the full range of experimental conditions, we first analyzed their activity patterns across the entire extended evaluation task videos. For each critical unit, we extracted activation values at every timestep and visualized these as activity matrices where rows represented individual trials and columns represented time. Trials were sorted according to their latent parameters (hazard rate and contingency), revealing how unit activities systematically varied with task statistics. We then examined their responses to controlled stimuli using the controlled color change task videos. For each identified critical unit, we extracted activation traces across all variants of each base trajectory, for each color. We aligned the traces across the trajectory variants to investigate how the neural activity evolved as a function of color changes.

Temporal Dynamics of Change Detection

To understand how units respond dynamically to state changes, we performed event-triggered analysis of neural activity surrounding color changes. This analysis was conducted separately for random changes (to characterize hazard rate encoding) and contingent changes (to characterize contingency encoding). For hazard rate, we identified all random color changes across the extended task videos and extracted activity windows surrounding each event. Windows spanned 16 time steps: 5 time steps before the change, the change timestep itself, and 10 time steps following the change. Changes were ordered by their occurrence within each video (first change, second change, etc.) to examine how representations evolved with experience. For contingency analysis, we performed similar event-triggered averaging around velocity changes, comparing trials where velocity changes were accompanied by color changes versus those without color changes across the contingency conditions.

3.2.4 Functional Identification by Neural Interventions

While the linear regularization analysis and neural tuning profiles showed correlations with task variables, neither of these approaches causally relate the units' activities to the model's color predictions. To bridge this gap, we developed a pipeline to perform targeted interventions on the task-specific units and assess their impact on color predictions.

Generating Intervention States

To create our "injection" states, we start by sub-selecting the critical units' activities from the last timestep where the color was visible for all trials in the extended task videos. From our neural tuning experiments, we found that activities from the final portions of a trial reflect the accumulated evidence for the latent parameters. To build the hazard rate interventions, we split the task videos by condition, and computed the mean unit activity for each, resulting in high-hazard and low-hazard injection values. We perform a

similar procedure for contingency and start by splitting the trials according to the high and low contingency conditions. We opt to exclude the medium contingency to maximize the intervention effect, and go with high-contingency and low-contingency injections.

Intervention Protocol

Our intervention protocol uses a targeted, single-timestep, state injection designed to test whether altering unit activities during a grayzone traversal causally affects color predictions. For each video, the models process the frames in the standard manner up to the final grayzone traversal. After the transition point, we performed a single-timestep intervention where we replaced the activity of a target unit i at the first grayzone timestep with a weighted combination of its original activity and a reference value. This intervention was applied to the hidden state $\mathbf{H}_i^{(t)}$ or the cell state $\mathbf{C}_i^{(t)}$, resulting in:

$$\tilde{\mathbf{X}}_i^{(t)} = (1 - \alpha) \cdot \mathbf{X}_i^{(t)} + \alpha \cdot \mathbf{X}_i^*, \tag{3.19}$$

where $\mathbf{X} \in \{\mathbf{H}, \mathbf{C}\}$, $\mathbf{X}_i^*$ denotes the condition-specific injection value, and $\alpha \in [0, 1]$ controls the intervention strength. This intervention protocol was applied separately to examine both paired units together and individual hidden or cell units in isolation. After the manipulation, the state activities evolved normally for the remainder of the grayzone traversal, and we recorded the resulting color predictions. We varied the intervention strength α across 10 values from 0 (no intervention) to 1 (complete substitution) to characterize the relationship between unit manipulation and behavioral change. Functional units were identified by whether the interventions produced changes in color predictions consistent with the unit's correlated statistic. Conversely, non-functional units were identified by whether the interventions produced minimal changes in color predictions.

3.3 Results

3.3.1 Neural Mechanisms for In-Context Learning in Recurrent Networks

Having established that LSTM networks achieve human-level performance on contingency learning while exceeding them at hazard rate detection, we next investigated the computational mechanisms underlying these capabilities. Reusing the same networks from Chapter 2, we analyzed the mechanisms underlying their behavior using three complementary approaches. We first identified task-relevant units through graded regularization, then characterized their temporal dynamics when faced with the different color change conditions, and then validated their causal role for behavior using targeted interventions. As we will show, both hazard rate and contingency units implement variations of a leaky integrator strategy triggered by different task conditions: continuous temporal updating versus discrete event-driven accumulation.

3.3.2 Sparse Neural Representations Encode Task-Relevant Information

Elastic Net Regularization Reveals Minimal Sufficient Representations

To identify which units encoded task-relevant behavior, we applied the elasticnet regularization pipeline to decode the hazard rate and contingency across varying degrees of regularization. For hazard rate decoding, the analysis revealed pairs of units, one in each of the hidden and cell states (fig. 3.1B), which were responsible for maintaining decoding performance through the majority of the alpha parameters (fig. 3.1A). This pattern was also found when we applied the pipeline to decode contingency, where classification performance remains high until the coefficients for a specific pair of units are brought to zero (fig. 3.1D). Additionally, these pairs always had corresponding indices, almost certainly due to the LSTM output gate mechanism Equation (3.12). See Figure S4 and

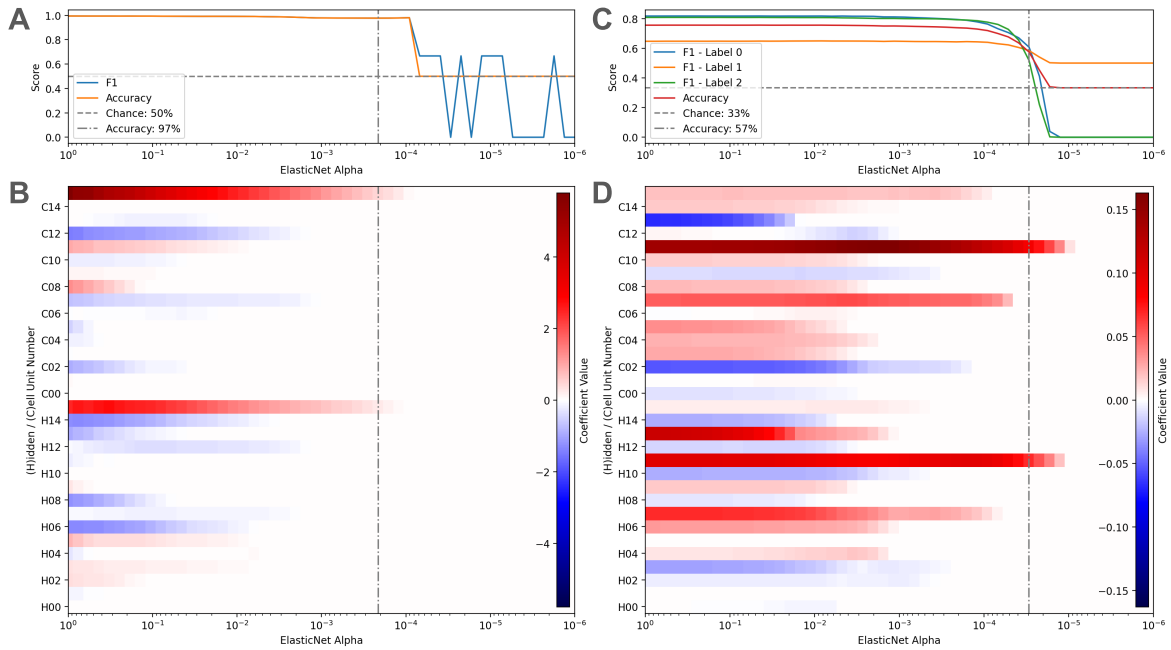


Figure 3.1: ElasticNet Regularization Identifies Critical Units. ElasticNet identifies critical units that encode task-relevant variables by systematically increasing regularization strength along seven orders of magnitude (10^0 to 10^{-6}). **(A)** Decoding accuracy for hazard rate (binary classification: high vs low) as elastic net alpha (regularization strength) increases, showing maintained performance of 97% until $\alpha \approx 10^{-4}$ (vertical dashed line) before dropping to chance (50%, horizontal dashed line). **(B)** Coefficient values for individual hidden (H) and cell (C) state units under increasing regularization strength for hazard rate decoding, revealing units H15, C15, maintain non-zero coefficients at high regularization strengths. **(C)** Decoding accuracy for contingency (binary classification: high vs low) maintains 57% accuracy until $\alpha \approx 10^{-4}$ before degrading to chance performance (33%). **(D)**Coefficient values for individual hidden (H) and cell (C) state units under increasing regularization strength for contingency decoding, revealing units H11, C11, maintain non-zero coefficients at high regularization strengths. Critical units were identified as those maintaining non-zero coefficients at regularization strengths just before performance dropped to chance ($\alpha = 10^{-4}$), revealing sparse encoding of task variables within the 32-dimensional LSTM state space.

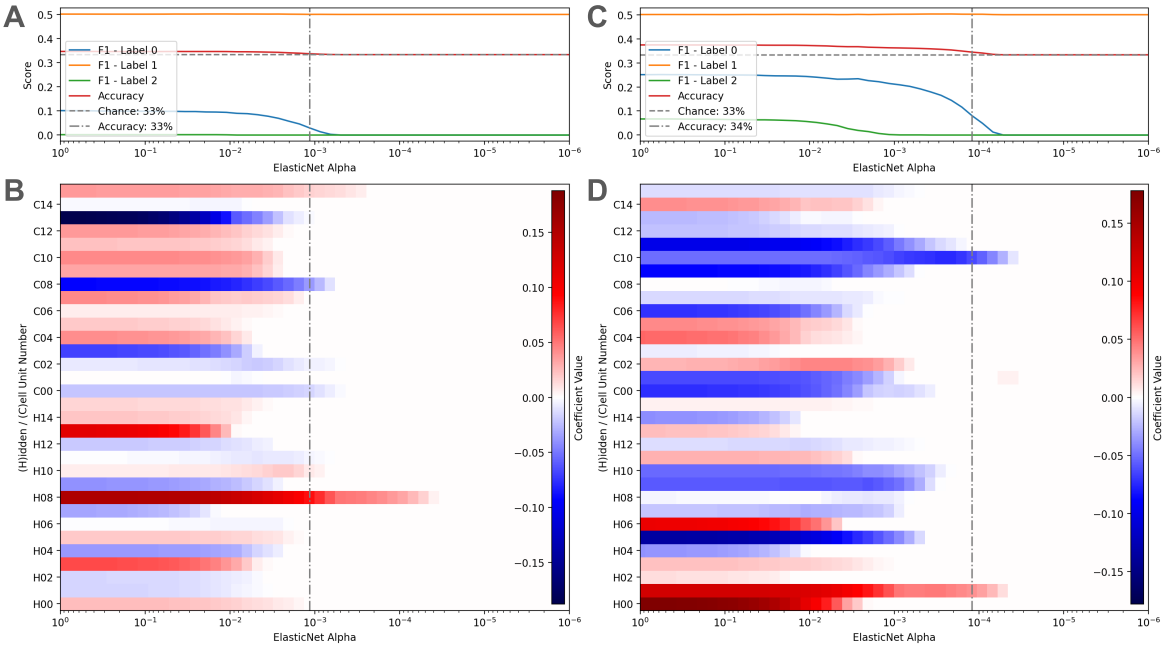


Figure 3.2: Contingency-Insensitive Models Do Not Represent Contingency. Elastic net regularization analysis reveals that models failing to learn contingency lack sparse representations., with decoding performance remaining at chance levels across all regularization strengths. **(A)** Decoding accuracy for contingency (3-class classification: low, medium, high) in model seed 1 starts close to chance performance (35% accuracy at $\alpha = 0$) and only decreases. **(B)** Coefficient heatmap for model seed 1 shows distributed activation across multiple units (H2, H4, H8, H12, C0, C2, C4, C8, C10, C12, C14) that persist even under strong regularization ($\alpha = 10^{-3}$. **(C)** Model seed 2 similarly shows chance-level contingency decoding (37% accuracy) that fails to improve regardless of regularization strength. **(D)** Coefficient paths for model seed 2 reveal a different but equally distributed pattern with strong coefficients across units (H0, H4, H6, H10, C0, C6, C8, C10, C12) that cannot be reduced to a sparse subset while maintaining performance. The inability reliably decode contingency (even with $\alpha = 0$ demonstrates contingency-insensitive models never developed specialized representations for velocity-contingent color changes.

Figure S5 to see identified units of other model seeds. These results reveal that sparse and distinct subsets of the 32-dimensional LSTM state space encode the different task parameters, which we call critical units.

Absence of Sparse Representations Reveals Learning Failures

While the regularization pipeline revealed, for all trained models, pairs of critical units for representing hazard rate, this was not the case for contingency. For a small subset of models (n=3), the decoder showed near-chance level decoding regardless of regularization strength (3.2, 3.2). The inability to decode contingency, even for regularization ($\alpha = 0$) demonstrates that these models never developed representations selective for velocity-change color changes. This was also seen during training, where these models' sensitivity to contingency never reached the final values reached by the other models.

3.3.3 Neural Dynamics Reveal Computational Strategies

Hazard Rate Units Implement Evidence Accumulation

Having identified the critical units for hazard rate and contingency, we set out to characterize their tuning profiles when presented with unit-relevant stimuli. We applied the event-triggered analysis and found that the critical units for hazard rate displayed discrete stepping upon observing random color changes (fig. 3.3). This effect was present for both the low and high hazard rate trials, but most visibly for the high hazard rate, where multiple successive color changes continued to raise the base activity (fig. 3.3, right). In the absence of random color changes, critical unit activity decayed over time, an effect that continually pushes unit activity opposite to the stepping on color changes. This effect is best shown in the pre-color change activity in the low hazard rate condition, where the unit's activity returned to the same starting level before each of the first two color changes. See Figure S6 to see identified units of other model seeds.

Together, this accumulation during color changes and decay during unchanged periods

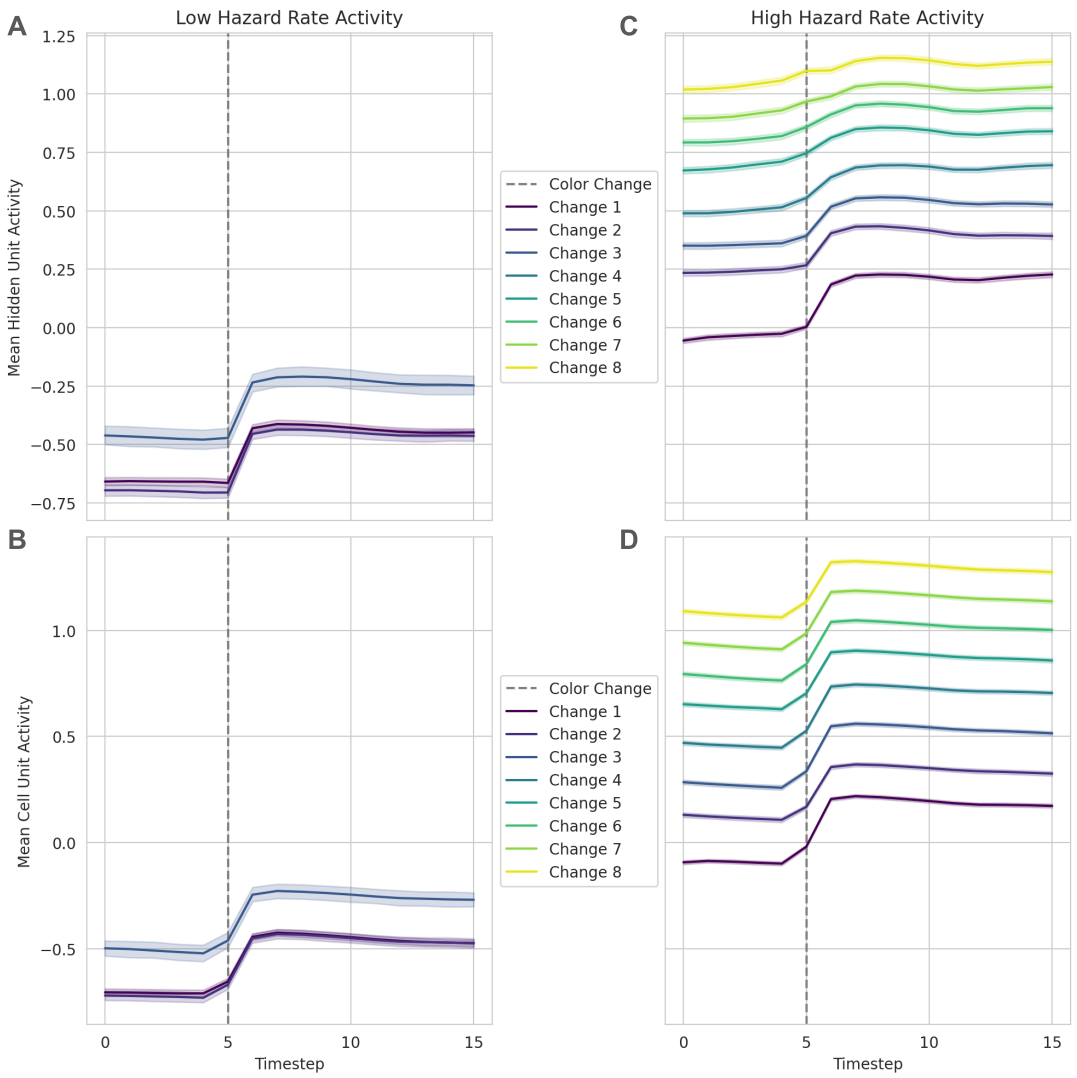


Figure 3.3: Hazard Rate Units Accumulate Color Changes. Event-triggered analysis of hazard rate encoding units reveals accumulating responses to random color changes that are preserved between the low and high hazard rate condition. Hidden unit activity is aligned to random color changes, showing activity profile 5 time steps before the change and 10 time steps after, with the change all occurring at timestep 5 (dashed line). Changes are then sorted according to which occurrence they are in the sequence. **(A)** Mean hidden unit activity across low hazard trials shows activity stepping with minimal accumulation. First and second changes overlap, indicating increase in activity after the first change likely decayed before the second. **(B)** Mean cell state activity under low hazard rate conditions shows the same pattern as the hidden state where there is clear stepping, but color changes are not frequent enough to produce noticeable accumulation effects. **(C)** Mean hidden unit activity under high hazard rate conditions shows clear accumulation effects after each random color change. Activity after each change settles into the same baseline activity going into the next. **(D)** Mean cell state activity under high hazard rate conditions shows the same accumulation pattern as the hidden state, with more uniform stepping between changes.

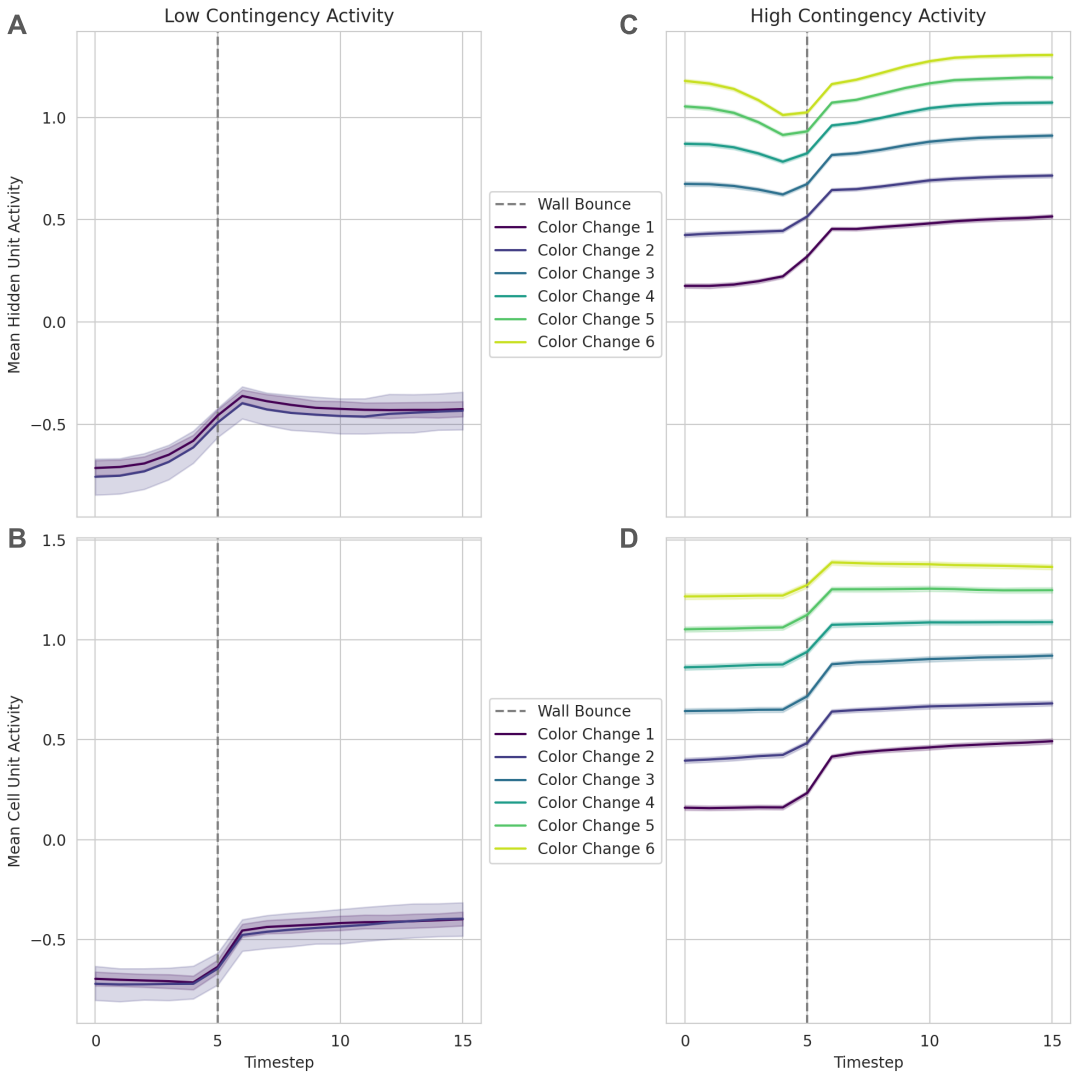


Figure 3.4: Contingency Units Accumulate Color Changes on Bounces. Event-triggered analysis of contingency encoding units reveals accumulating responses to velocity-contingent color changes that are preserved between the low and high contingencies. Hidden and cell unit activity are aligned to wall bounces where color changes occur, showing activity profiles 5 time steps before the bounce and 10 time steps after, with bounces occurring at timestep 5 (dashed line). Color change bounces are sorted according to occurrence in the sequence of contingent changes. **(A)** Mean hidden unit activity across low contingency trials (5% change probability) shows activity stepping with minimal accumulation. First and second changes largely overlap indicating any accumulation effects had diminished between the events. **(B)** Mean cell state activity under low contingency conditions shows the same pattern as the hidden state where there is clear stepping but no accumulation. **(C)** Mean hidden unit activity under high contingency conditions (95% change probability) shows clear positive accumulation effects after each velocity-contingent color change. Activity after each change settles into progressively higher baseline levels before the next event. **(D)** Mean cell state activity under high contingency conditions demonstrates similar accumulation patterns, with more uniform stepping between changes across six contingent color changes.

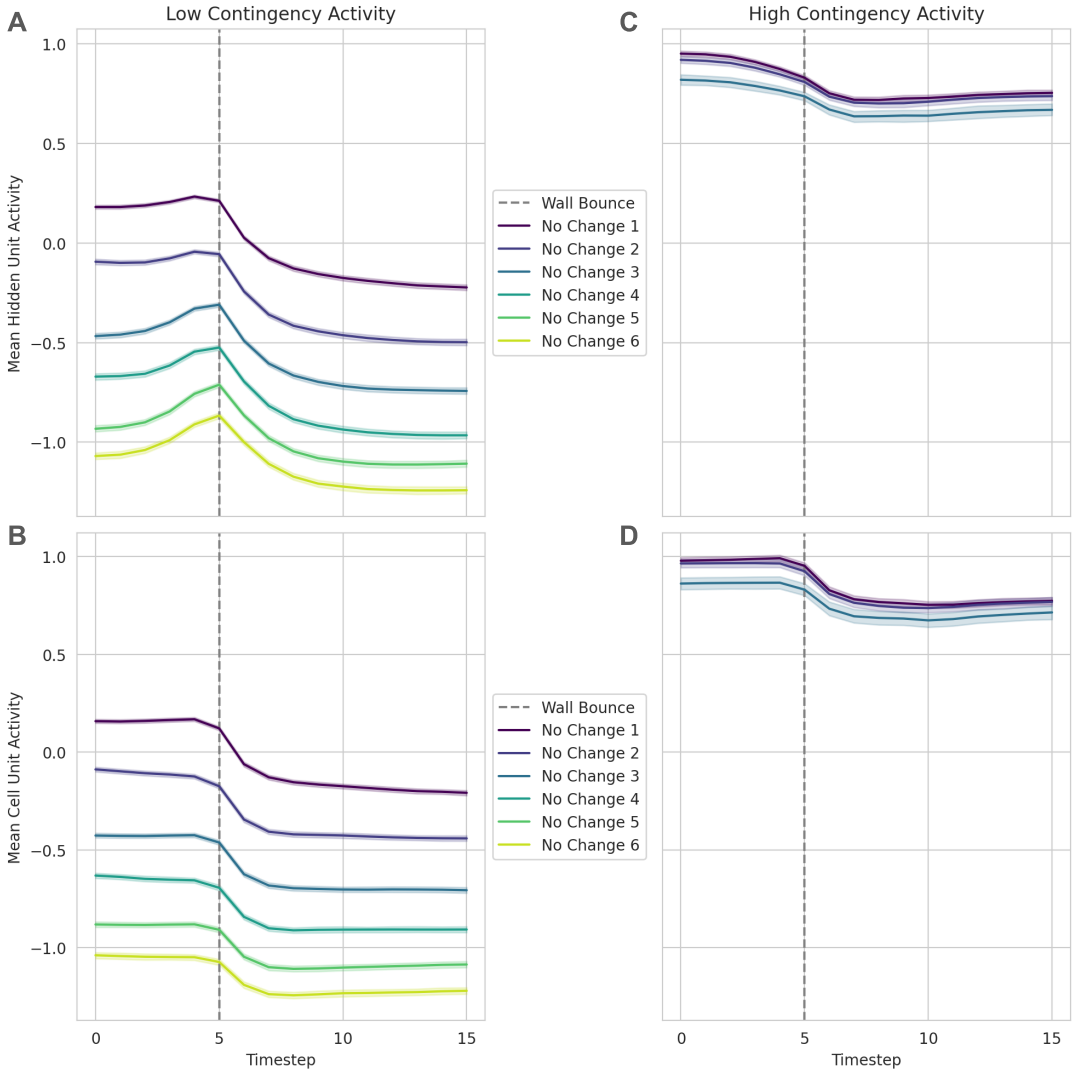


Figure 3.5: Contingency Units Accumulate Non-Changes on Bounces. Event-triggered analysis of contingency encoding units reveals negative activity deflections to velocity changes with no color change. Hidden and cell unit activity are aligned to wall bounces where color changes did not occur, showing activity profiles 5 time steps before the bounce and 10 time steps after, with bounces occurring at timestep 5 (dashed line). Bounces without color changes are sorted according to occurrence in the sequence of contingent changes. **(A)** Mean hidden unit activity across low contingency trials (5% change probability) shows activity down-stepping on each bounce with no color changes. Activity settles at the end of each sequence to the baseline level before the next change, showing negative accumulation. **(B)** Mean cell state activity under low contingency conditions shows the same pattern as the hidden state where there is clear negative stepping and with more conserved profiles. **(C)** Mean hidden unit activity under high contingency conditions (95% change probability) shows minimal accumulation effects between the first three occurrences. Indicates that unit activity rises back to prior state between each of the bounces. **(D)** Mean cell state activity under high contingency conditions shows the same pattern as the hidden unit.

implements a leaky integrator that maintains a running estimate of the hazard rate, mirroring the temporal evidence integration performed by the ideal Bayesian observer. We next tested whether contingency units showed the complementary pattern, affecting only event-based predictions while leaving temporal predictions intact.

Contingency Units Show Event-Triggered Responses

Similarly to the hazard rate units, critical units for contingency also displayed discrete stepping behavior when observing contingent color changes (fig. 3.4). Like the hazard rate units, this was clearest to see in the high contingency trials where each color change raised the base activity of the unit (fig. 3.4, right). However, since contingency tracking depends on discrete events, we also analyzed its response to bounces with no color change and found it had the opposite effect, stepping downwards (fig. 3.5). This was most apparent for the low contingency trials, where each successive bounce with color change drove the activity of the unit further downward (fig. 3.5, left). This bidirectional response pattern reveals an evidence integration strategy in which the unit accumulates support both for and against the contingency relationship, with positive accumulation signaling high contingency and negative accumulation signaling low contingency. Like the hazard rate units, contingency units also implement a leaky integrator, but one that is selectively triggered by velocity changes rather than continuously updating over time. This event-specific accumulation mechanism allows the network to maintain separate estimates for temporal and event-based probabilities, with both hazard rate and contingency units implementing the same leaky integrator computation but with different triggers. See Figure S7 to see identified units of other model seeds.

3.3.4 Causal Validation Through Targeted Neural Interventions

Having shown that critical unit activity is both highly correlated with task relevant variables and demonstrates a potential mechanism for their representation, we sought to

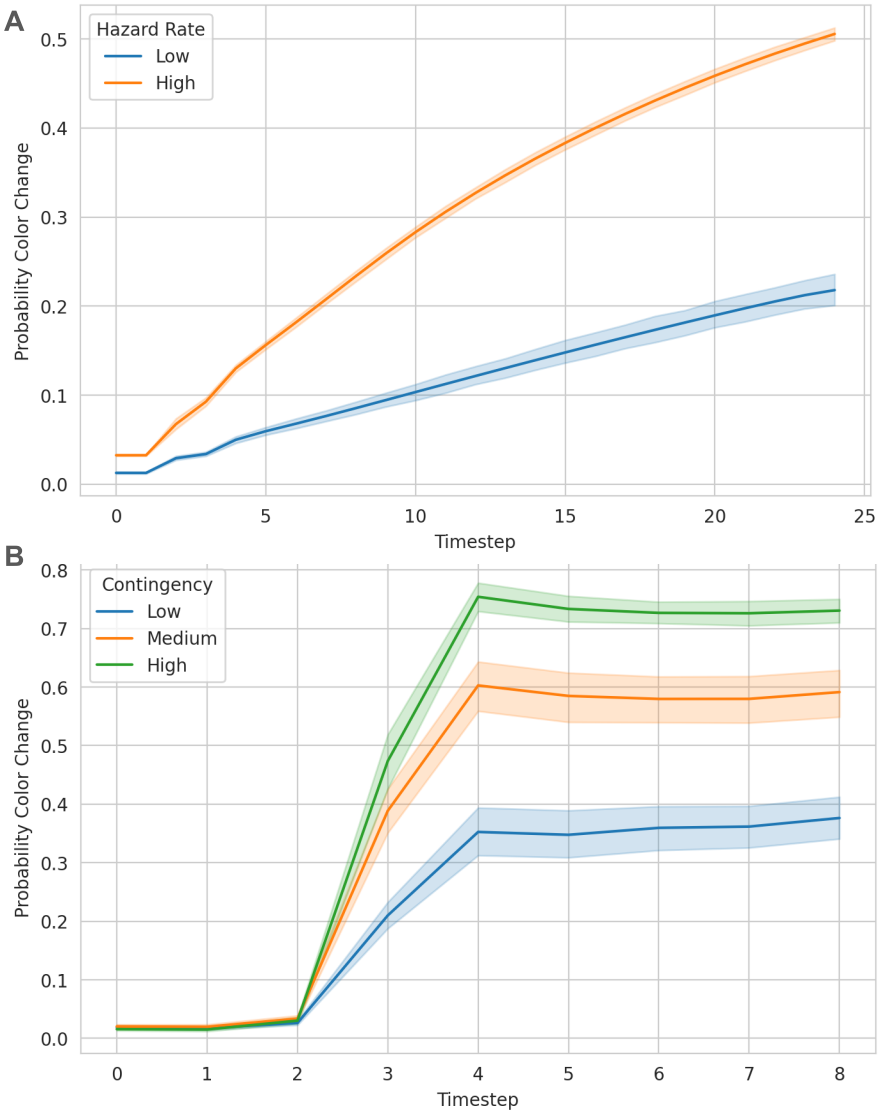


Figure 3.6: Baseline LSTM Grayzone Color Predictions. Baseline LSTM mean color change predictions (computed as 1 minus the probability of the entering color) on the straight and bounce path trials for each task condition. These predictions establish the target behavioral changes for intervention experiments, where successful manipulations of critical units should shift predictions from one parameter condition toward another. **(A)** Mean probability of color change during straight path trials increase monotonically with time in the gray zone, with low hazard rate conditions (blue) rising more slowly than high hazard rate conditions (orange) demonstrating temporal integration of change probability. **(B)** Mean probability of color change for bounce path trials shows rapid transitions at the bounce event (timestep 18).

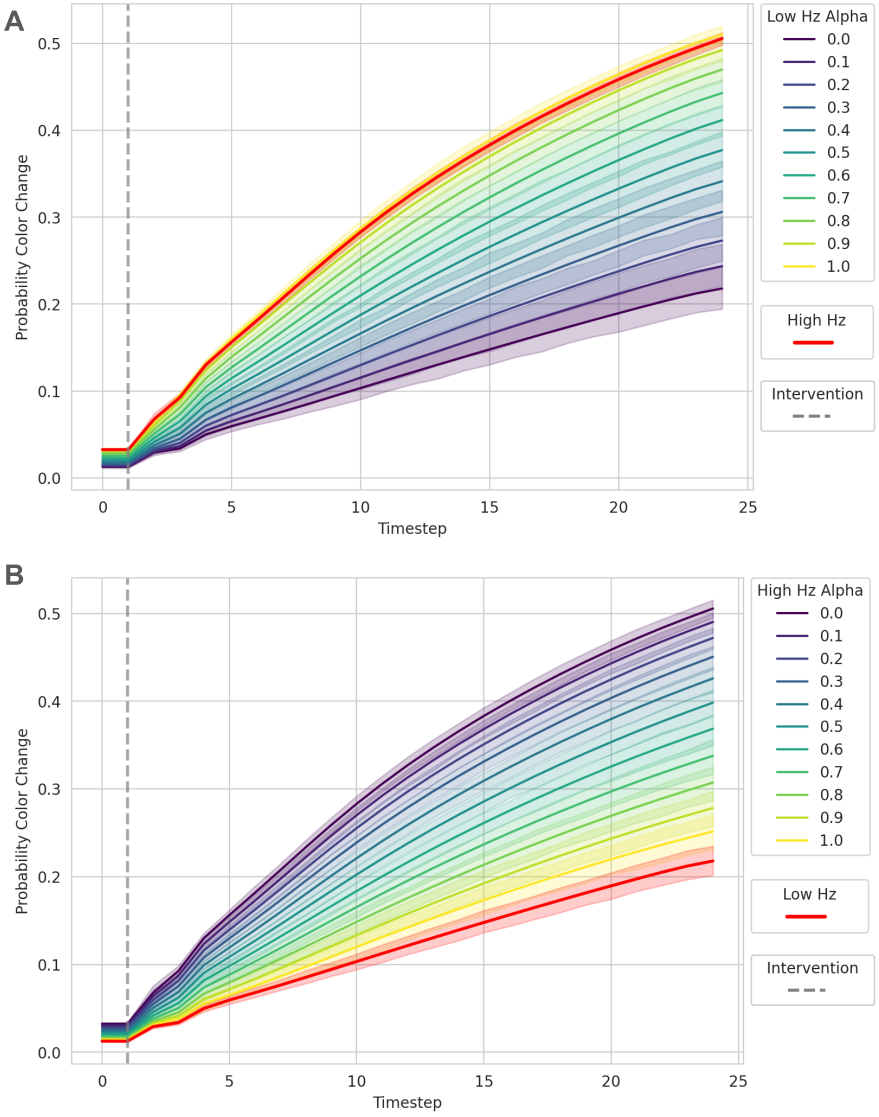


Figure 3.7: Hazard Rate Interventions Causally Shift Time-Dependent Color Predictions. Targeted state injections at gray zone entry (timestep 1, vertical dashed line) demonstrate that identified critical units causally control hazard rate encoding in the LSTM. Intervention strength α is varied across 11 steps from 0 (no intervention, purple) to 1 (full state replacement, yellow), with the target condition prediction shown in red. **(A)** Interventions on low hazard rate trials progressively shift predictions toward high hazard rate behavior as intervention strength α increases. With no intervention ($\alpha = 0$), predictions follow the baseline low hazard rate trajectory reaching 0.23 by timestep 24, while complete intervention ($\alpha = 1.0$) pushes predictions to 0.49, approaching the high hazard rate target (red line, 0.51). Intermediate intervention strengths produce graded effects, with $\alpha = 0.5$ (cyan) yielding predictions of 0.36 at timestep 24, demonstrating dose-dependent control. **(B)** Complementary interventions on high hazard rate trials inject low hazard rate states, progressively reducing color change predictions from the baseline high hazard rate trajectory.

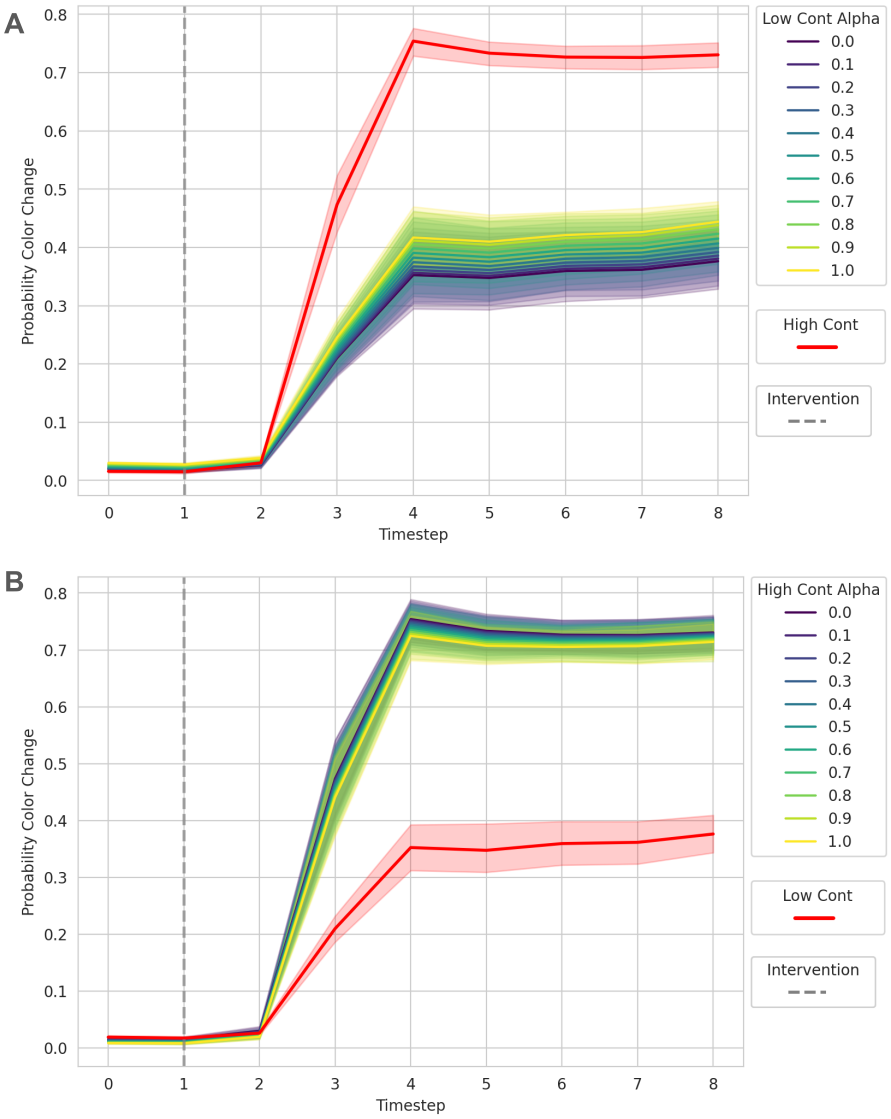


Figure 3.8: Hazard Rate Interventions Do Not Shift Velocity-Change Dependent Color Predictions. Hazard rate unit interventions applied to bounce path trials at gray zone entry (timestep 1, vertical dashed line) test whether these units influence velocity-contingent color predictions. Intervention strength α is varied across 11 steps from 0 (no intervention, purple) to 1 (full state replacement, yellow), with the target condition prediction shown in red. **(A)** Interventions on low contingency trials using high hazard rate states show minimal effect on contingency-driven predictions at the bounce event (timestep 5). Despite complete intervention ($\alpha = 1.0$, yellow), predictions plateau at 0.45 after the bounce, below the high contingency target (red line, 0.74), with the sharp transition at bounce timing preserved across all intervention strengths. **(B)** Complementary interventions on high contingency trials (95% baseline) using low hazard rate states similarly fail to alter contingency-dependent predictions.

identify whether it plays a causal role with behavior. To establish causal relationships between these identified units and behavior, we performed targeted state injections at gray zone entry, varying intervention strength α from 0 (no intervention) to 1 (complete replacement) while measuring effects on color change predictions (baseline predictions in Figure 3.6).

Hazard Rate Interventions Systematically Shift Color Predictions

Using the states endogenous to the model at the final time steps of either high or low hazard rate trials, we performed graded state injections on the critical units for hazard rate. When looking at the outcomes of straight path trials, we found that injecting hazard rate unit states into low hazard trials produced dose-dependent increases in color change predictions, reaching 0.49 at full intervention ($\alpha = 1.0$) compared to 0.23 baseline, approaching the high hazard target of 0.51 (fig. 3.12). Complementary injections of low hazard states into high hazard trials symmetrically reduced predictions to those expected of the low hazard rate condition(fig. 3.12). We considered that these interventions may simply affect color predictions made due to wall bounces. To test this, we applied interventions in the same manner before the bounce trials to see if they had an effect on the outcomes of velocity-change dependent color changes. Crucially, these interventions had no effect on color changes due to bounce events, with predictions remaining at baseline levels despite full intervention (fig. 3.8). This dissociation, where hazard rate interventions affect only temporal predictions while leaving contingency-driven predictions intact, demonstrates that the network maintains functionally independent circuits for time-based versus event-based in-context learning. When repeating this process for all model seeds, we found that these dose-dependent intervention effects were consistent across all trained models, demonstrating that temporal probability tracking through these critical units represents a robust and reproducible computational solution for hazard rate tracking.

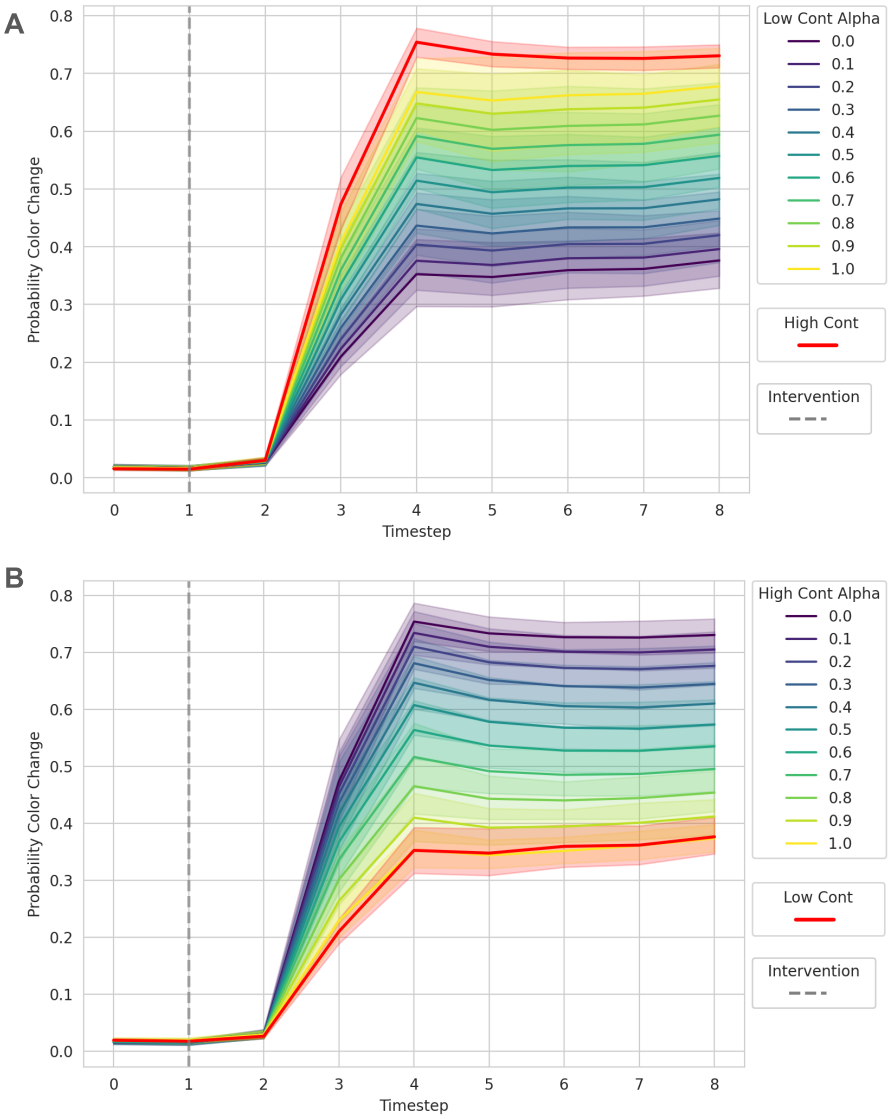


Figure 3.9: Contingency Interventions Causally Shift Velocity-Change Dependent Color Predictions. Targeted state injections at gray zone entry (timestep 1, vertical dashed line) demonstrate that identified critical units causally control contingency encoding in the LSTM. Intervention strength α is varied across 11 steps from 0 (no intervention, purple) to 1 (full state replacement, yellow), with the target condition prediction shown in red. **(A)** Interventions on low contingency trials (5% baseline) progressively shift predictions toward high contingency behavior at the bounce event (timestep 5) as intervention strength α increases. With no intervention ($\alpha = 0$), predictions follow the baseline low contingency trajectory plateauing at 0.38 after the bounce, while complete intervention ($\alpha = 1.0$) pushes predictions to 0.72, approaching the high contingency target (red line, 0.74). Intermediate intervention strengths produce graded effects, with $\alpha = 0.5$ (cyan) yielding predictions of 0.55 after the bounce, demonstrating dose-dependent control specifically triggered by the velocity change event. **(B)** Complementary interventions on high contingency trials (95% baseline) inject low contingency states, progressively reducing color change predictions after the bounce from the baseline high contingency trajectory.

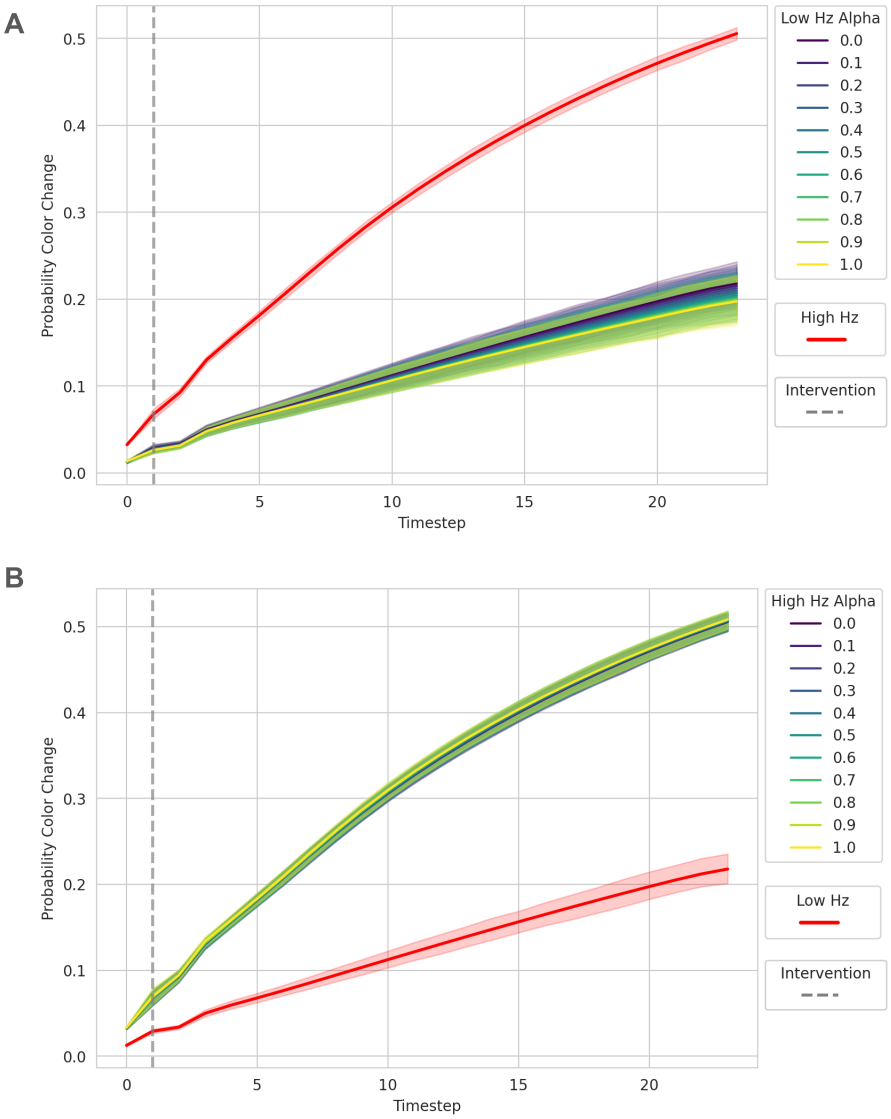


Figure 3.10: Contingency Interventions Causally Shift Velocity-Change Dependent Color Predictions. Contingency unit interventions applied to straight path trials at gray zone entry (timestep 1, vertical dashed line) test whether these units influence time-dependent color predictions. Intervention strength α is varied across 11 steps from 0 (no intervention, purple) to 1 (full state replacement, yellow), with the target condition prediction shown in red. **(A)** Interventions on low hazard rate trials using high contingency states show no effect on temporal color change predictions throughout the gray zone traversal. Despite complete intervention ($\alpha = 1.0$, yellow), predictions remain at 0.23 by timestep 24, identical to the baseline trajectory (purple, $\alpha = 0$) and far below the high hazard rate target (red line, 0.51), with all intervention strengths producing overlapping trajectories. **(B)** Complementary interventions on high hazard rate trials using low contingency states similarly fail to alter time-dependent predictions.

Contingency Interventions Produce Bounce-Specific Effects

Having demonstrated the causal role between the critical units for hazard rate, we turned to critical units for contingency. We repeated the same procedure used for the hazard rate interventions and used the observed states of the model at the end of low and high contingency trials as injection states. When looking at the outcomes of bounce path trials, we found that injecting high contingency unit states into low contingency trials shifted predictions at the bounce event from 0.38 to 0.72 at full intervention (fig. 3.14). Complementary injections of low contingency states into high contingency trials symmetrically reduced predictions to those expected of the low contingency rate condition (fig. 3.14). We then performed the same intervention on the straight path trials and found that interventions on contingency critical units had no effect on the prediction outcomes for straight path trials (fig. 3.10). This selective effect on bounce-triggered but not time-dependent predictions confirms functional specialization of contingency units. When repeating this process for all model seeds, we found these intervention effects were consistent across every model that successfully developed sparse representations, indicating that this circuit architecture represents a convergent solution rather than the idiosyncrasies of neural networks.

Dissociating Correlation from Causation

Thus far, we had performed interventions on each pair of critical units for both hazard rate and contingency. These pairs were identified as necessary for maintaining high decoding accuracy for their respective task variables. Each pair of units included one the hidden state and one from the cell state, each serving a different function within the LSTM block. The cell state remains within the block and serves as the core storage site for memory within the LSTM, whereas the hidden state is fed back into the LSTM and then outward to the next module. Considering the differing roles of these units, we suspected they may not have the same causal relationship to behavior.

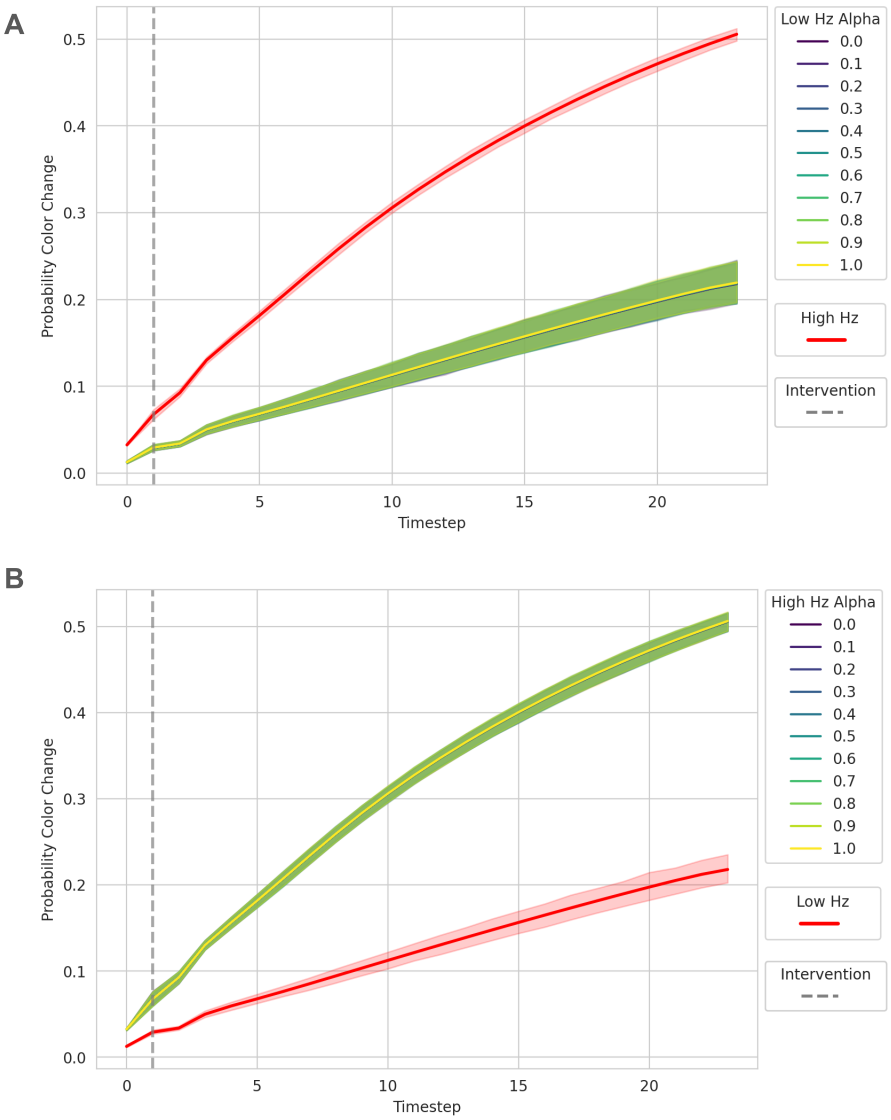


Figure 3.11: Isolated Hidden Unit Interventions Do Not Shift Time-Dependent Color Predictions. To test whether individual components of the critical unit pairs are sufficient for behavioral control, interventions were performed on hidden units while leaving the cell units unmodified. Hidden unit interventions applied at grayzone entry (timestep 1, vertical dashed line) test whether the hidden component independently encodes hazard rate information. Intervention strength α is varied across 11 steps from 0 (no intervention, purple) to 1 (full state replacement, yellow), with the target condition prediction shown in red. **(A)** Interventions on low hazard rate trials using high hazard rate hidden states show minimal effect on temporal color change predictions. Despite complete intervention ($\alpha = 1.0$, yellow), predictions reach only 0.24 by timestep 24, nearly identical to the baseline trajectory (purple, $\alpha = 0$, 0.23) and far below the high hazard rate target (red line, 0.51). **(B)** Complementary interventions on high hazard rate trials using low hazard rate hidden states similarly fail to alter predictions.

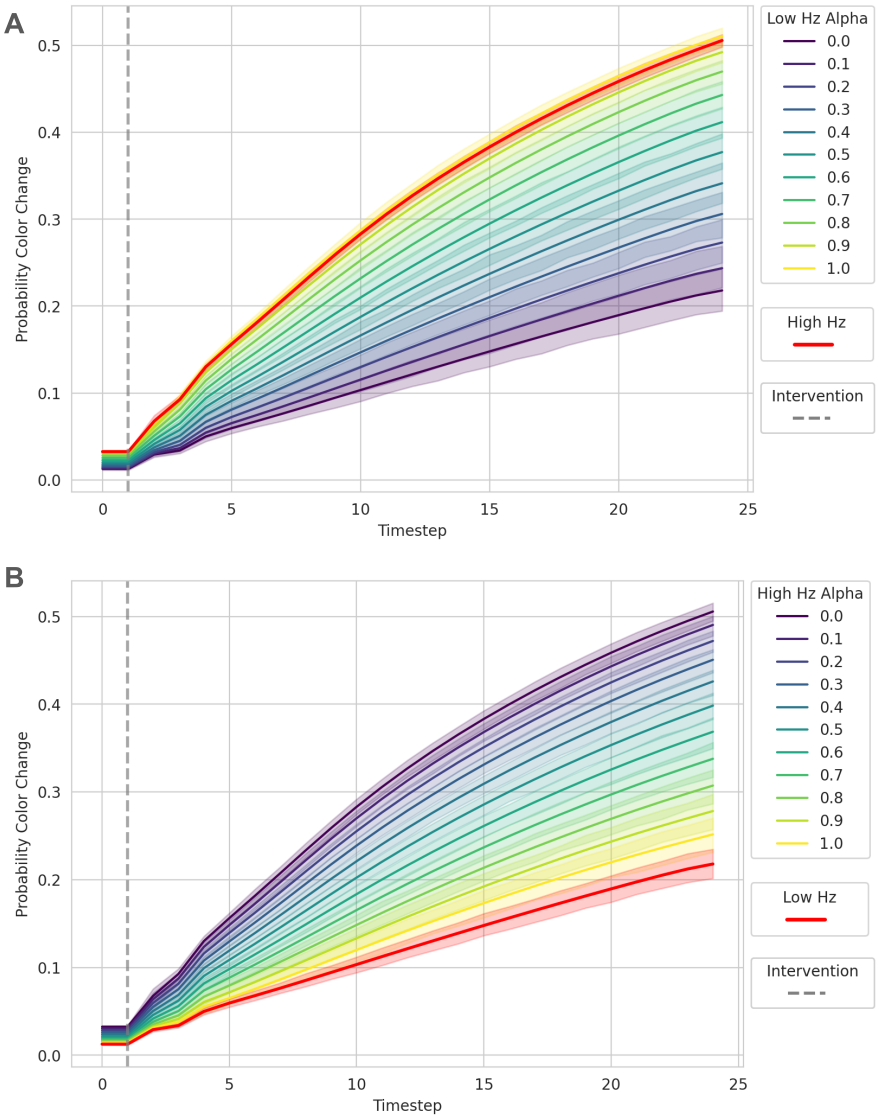


Figure 3.12: Isolated Cell Unit Interventions Causally Shift Time-Dependent Color Predictions. To test whether individual components of the critical unit pairs are sufficient for behavioral control, interventions were performed on cell states while leaving hidden units unmodified. Cell state interventions applied at gray zone entry (timestep 1, vertical dashed line) test whether the cell component independently controls hazard rate encoding. Intervention strength α is varied across 11 steps from 0 (no intervention, purple) to 1 (full state replacement, yellow), with the target condition prediction shown in red. **(A)** Interventions on low hazard rate trials using high hazard rate cell states progressively shift predictions toward high hazard rate behavior as intervention strength α increases. With no intervention ($\alpha = 0$, purple), predictions follow the baseline trajectory reaching 0.23 by timestep 24, while complete intervention ($\alpha = 1.0$, yellow) pushes predictions to 0.49, closely approaching the high hazard rate target (red line, 0.51). Intermediate intervention strengths produce dose-dependent effects, with $\alpha = 0.5$ (cyan) yielding predictions of 0.36 at timestep 24, demonstrating systematic control through cell state manipulation alone. **(B)** Complementary interventions on high hazard rate trials using low hazard rate cell states progressively reduce color change predictions.

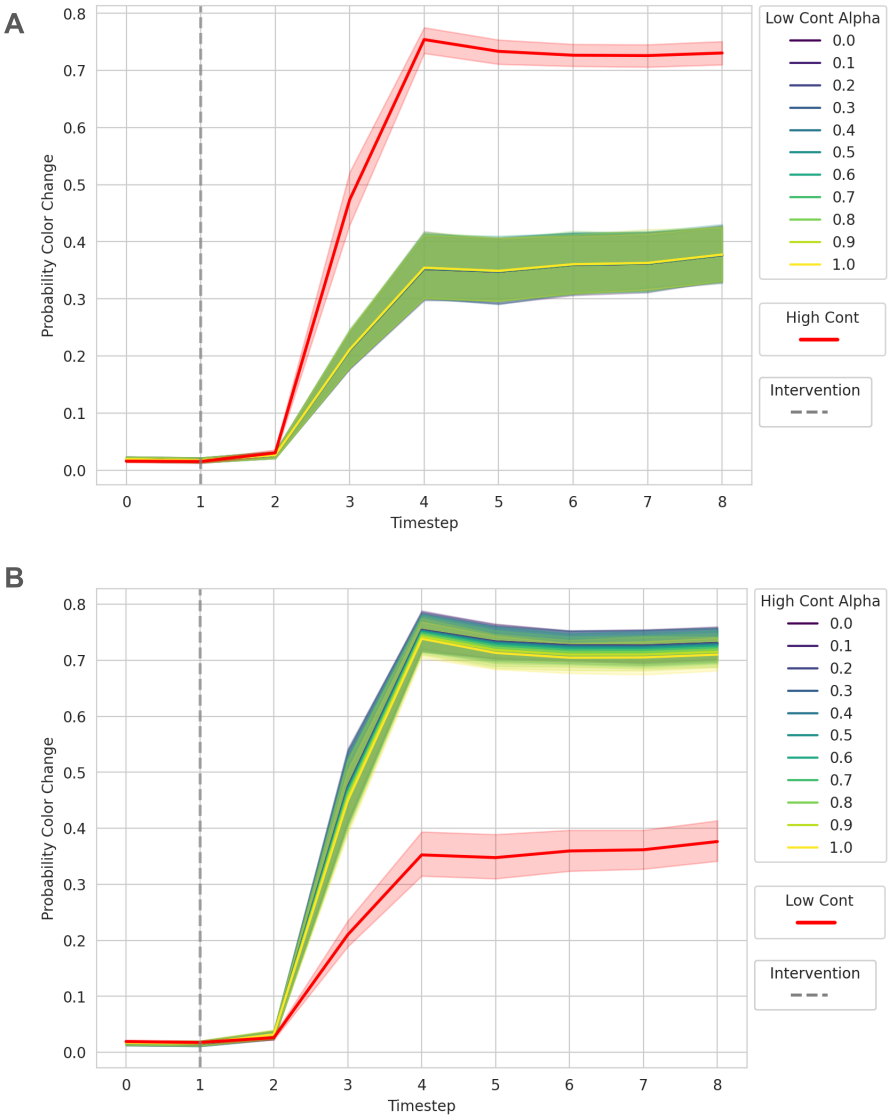


Figure 3.13: Isolated Hidden Unit Interventions Do Not Shift Velocity-Change Dependent Color Predictions. To test whether individual components of the critical unit pairs are sufficient for velocity-change dependent control, interventions were performed on hidden units alone while leaving the cell units unmodified. Hidden unit interventions applied at gray zone entry (timestep 1, vertical dashed line) test whether the hidden component independently encodes contingency information. Intervention strength α is varied across 11 steps from 0 (no intervention, purple) to 1 (full state replacement, yellow), with the target condition prediction shown in red. **(A)** Interventions on low contingency trials using high contingency hidden states show minimal effect on velocity-contingent color predictions at the bounce event (timestep 5). Despite complete intervention ($\alpha = 1.0$, yellow), predictions plateau at only 0.41 after the bounce, far below the high contingency target (red line, 0.74), with minimal separation from the baseline trajectory (purple, $\alpha = 0$, 0.38). All intervention strengths produce highly overlapping trajectories with negligible dose-dependent effects, demonstrating that hidden unit manipulation alone cannot shift contingency encoding. **(B)** Complementary interventions on high contingency trials using low contingency hidden states similarly fail to alter velocity-contingent predictions.

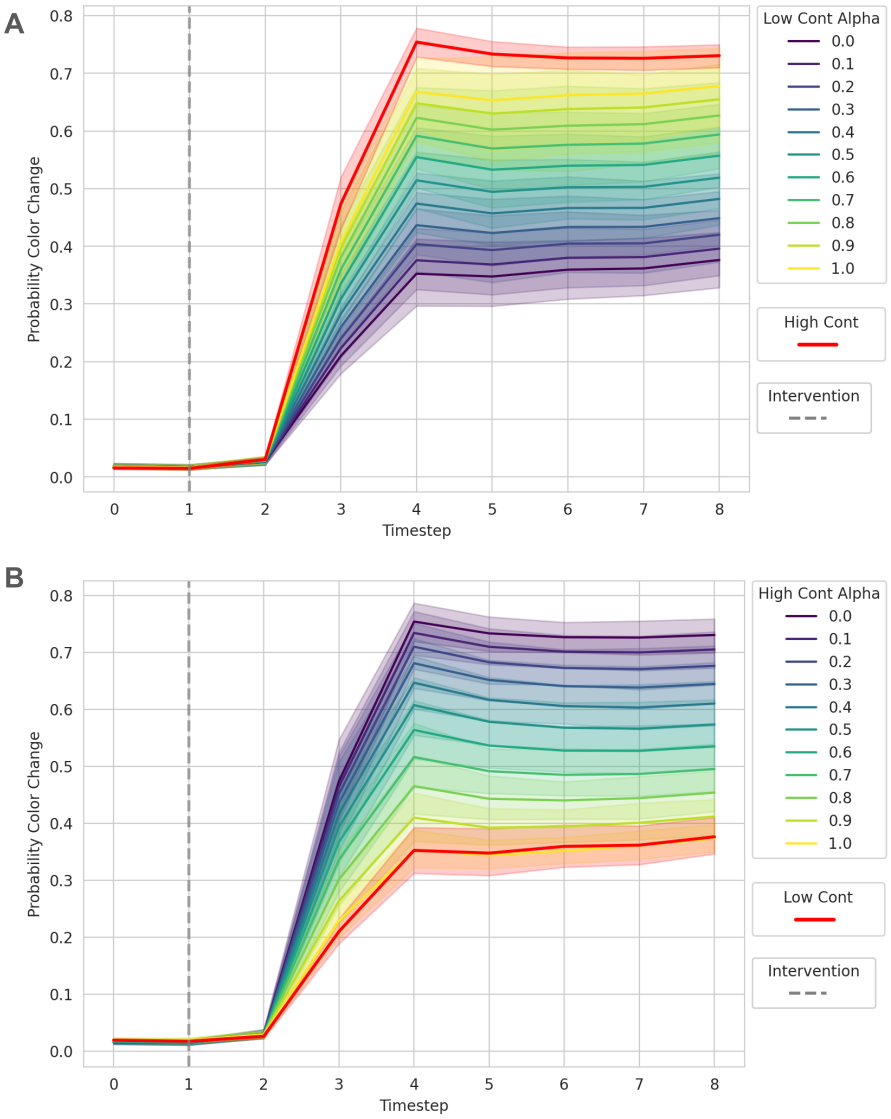


Figure 3.14: Isolated Cell Unit Interventions Causally Shift Velocity-Change Dependent Color Predictions. To test whether individual components of the critical unit pairs are sufficient for behavioral control, interventions were performed on cell states while leaving hidden units unmodified. Cell state interventions applied at grayzone entry (timestep 1, vertical dashed line) test whether the cell component independently controls contingency encoding. Intervention strength α is varied across 11 steps from 0 (no intervention, purple) to 1 (full state replacement, yellow), with the target condition prediction shown in red. **(A)** Interventions on low contingency trials using high contingency cell states progressively shift predictions toward high contingency behavior after the bounce event as intervention strength α increases. With no intervention ($\alpha = 0$, purple), predictions follow the baseline trajectory plateauing at 0.38 after the bounce, while complete intervention ($\alpha = 1.0$, yellow) pushes predictions to 0.68, approaching the high contingency target (red line, 0.74). Intermediate intervention strengths produce dose-dependent effects, with $\alpha = 0.5$ (cyan) yielding predictions of 0.53 after the bounce, demonstrating systematic control through cell state manipulation alone. **(B)** Complementary interventions on high contingency trials using low contingency cell states progressively reduce velocity-contingent predictions.

Having shown that these units together played a causal role on the color-change prediction outcomes of the models, we next sought to understand the degree to which each unit contributed to this causal relationship. To test this relationship, we reran our intervention experiments for hazard rate and contingency on each individual unit in the critical unit pairs. Our results show that for both time-dependent and velocity-change dependent color-change predictions, cell state interventions were necessary and sufficient to reproduce all outcomes seen earlier (fig. 3.12, fig. 3.14). In contrast, hidden state interventions produced minimal behavioral changes despite these units showing strong correlations with task variables (fig. 3.11, fig. 3.13). This dissociation demonstrates that LSTMs implement task-relevant computations through their cell states, which are directly affected by forget and input gates. Since the cell states are not directly used for predictions, they store the task parameter estimate, while the hidden states serve primarily as a readout rather than causal drivers of behavior.

Together these results reveal that LSTM networks solve the Bouncing Ball task through sparse, functionally specialized units that implement parallel leaky integrator strategies. Hazard rate units continuously accumulate evidence over time, while contingency units selectively accumulate evidence at velocity change events. This provides a mechanistic account of what the LSTM is representing to allow simultaneous temporal and event-based in-context learning.

3.3.5 Summary of Key Findings

Five main findings emerged from our analyses:

1. **Sparse Encoding of Task Parameters**: LSTM networks encode hazard rate and contingency through distinct, minimal subsets of units. Elastic net regularization revealed that just two units (out of 32) were sufficient for maintaining hazard rate decoding and contingency decoding, with these critical units appearing at corresponding indices in hidden and cell states.

2. **Leaky Integrator Implementation**: Both hazard rate and contingency units implement leaky integrator computations but with different triggers. Hazard rate units continuously accumulate evidence over time, decaying between changes, while contingency units only accumulate evidence at velocity changes.

3. **Functional Independence of In-Context Learning Circuits**: Causal interventions demonstrated complete independence between temporal and event-based learning. Hazard rate manipulations affected only time-dependent predictions while leaving bounce-triggered predictions intact, and contingency interventions showed the complementary pattern, confirming circuit independence.

4. **Cell States as Causal Controllers**: While both hidden and cell states showed strong correlations with task variables, only cell state interventions produced behavioral changes. This architectural constraint reveals that LSTMs implement in-context learning computations through their persistent memory, while hidden states serve as readouts.

5. **Convergent Solution Across Networks**: The sparse encoding architecture emerged across all trained models for hazard rate and across models that learned contingency. This convergence suggests that sparse, specialized units implementing leaky integration represent an optimal computational solution for multi-timescale in-context learning rather than an idiosyncratic implementation.

3.4 Discussion

The mechanistic analyses revealed that LSTM networks solve the Bouncing Ball task through sparse circuits implementing parallel leaky integration. Just two units per parameter suffice for encoding hazard rate and contingency, with both using the same accumulation strategy but triggered by different events. These findings connect normative theories of Bayesian inference to concrete neural implementations while revealing

fundamental constraints on in-context learning architectures.

3.4.1 Computational Principles of In-Context Learning

The leaky integrator circuits we identified provide a mechanistic account of how networks approximate Bayesian inference. The Ideal Bayesian Observer maintains Beta distributions over hazard rate and contingency, updating counts when observing relevant events. Our neural units implement a simplified version of this computation through exponential accumulation and decay. For hazard rate, units increment with each color change and decay between changes, effectively tracking the rate of spontaneous events over time. For contingency, the same accumulation process occurs but only at velocity changes, with upward steps for color changes and downward steps for non-changes. This shared computational motif suggests that in-context learning across different timescales doesn't require fundamentally different mechanisms, just different triggers for the same integration process.

The biological plausibility of these integrator circuits deserves consideration. Neural integrators have been identified in multiple brain regions, from oculomotor control to decision making. The decay we observe matches known properties of synaptic depression and adaptation. But more interesting is the selective triggering mechanism, where contingency units only update at specific events. This resembles gating mechanisms in basal ganglia-thalamo-cortical loops, where specific conditions trigger updating of working memory representations. While we can't claim LSTMs directly model these circuits, the convergent solution suggests similar computational constraints shape both biological and artificial systems solving in-context learning problems.

3.4.2 Architectural Constraints and Cell State

Our interventions revealed an unexpected asymmetry between hidden and cell states in driving behavior. Despite both showing strong correlations with task variables, only

cell state manipulations produced behavioral changes. This makes sense given the LSTM's design, where cell states maintain information across time while hidden states get reconstructed each timestep through the output gate. The cell state provides the persistent memory that enables integration across hundreds of time steps, while hidden states merely read out current values for immediate use.

This architectural constraint has implications for understanding memory-based flexibility. The separation between storage (cell state) and readout (hidden state) allows the same information to be maintained while its influence on behavior gets modulated by gates. In our task, this means hazard rate and contingency estimates persist in cell states even when they're not immediately relevant for prediction. The network learns not just what to track but when to use it. This echoes theories of working memory where information gets maintained in activity-silent states until task demands require retrieval.

3.4.3 Convergence and Optimality

The same sparse encoding emerged across most networks, but not all. Seven of ten LSTM seeds developed the sparse circuits we identified, while three failed to consolidate contingency information into dedicated units. This 70% success rate reveals something important about the learning dynamics. The networks that succeeded all showed a characteristic training trajectory where contingency sensitivity remained low initially, matching human performance for the first 5% of training epochs, before undergoing rapid improvement. The three that failed never experienced this transition, their contingency sensitivity plateauing at the RNN level despite continued training. Examining the decoder accuracy further confirmed that contingency was not being represented in the states.

The correspondence between human balanced responders and LSTMs at 78.9% versus 80.7% optimal contingency performance suggests convergent computational solutions. Both populations demonstrate the same evidence accumulation capability when discrete, countable events provide the input. The RNN's uniformly lower performance across both

parameters leaves ambiguous whether it lacks accumulation machinery or counting ability or both. It is likely that the root of the LSTM's success comes not from fundamentally different computations, but from having sufficient memory capacity, through its cell states, to maintain counts across longer timescales while still processing incoming information.

3.4.4 Limitations and Future Directions

Several limitations constrain our conclusions. We only examined LSTM architectures, though the principles might extend to Transformers or other sequence models. Comparing integration strategies across architectures could reveal whether leaky integration is universally optimal or specific to recurrent processing.

Mechanistically, the sparse states we identified describe what the models are doing to generate predictions. By representing the hazard rate and contingency through directly accumulating their respective events, the models can adapt their outputs in-context. However, this naturally leads to the question of how the model is actively engaging with these units, and consequently, how are they being used to generate predictions.

The previous chapter showed that the primary difference between balanced responders and LSTMs was the ability to count eligible time steps, not the capacity for accumulation or Bayesian inference. Both populations exhibited comparable contingency sensitivity because discrete bounces are countable, but not for hazard rate sensitivity. But this leaves unexplored how LSTMs actually implement the counting that enables their hazard rate performance. A more detailed network analysis could reveal how the LSTM selectively activates contingency units only at velocity changes while maintaining perpetual decay in hazard rate units between color changes.

Additionally, the mechanism connecting unit activity to behavioral output remains unexplored. We know hazard rate units accumulate evidence, and we know their manipulation shifts color predictions, but the intermediate computation is unclear. How does a specific activity level in these two units get transformed into a three-way color probability?

The network must somehow combine temporal and contingent evidence, weighing them appropriately based on context. Straight path trials should ignore contingency units entirely, while bounce trials need both signals integrated. Understanding this gating and combination would reveal how sparse representations enable flexible behavior.

Future work should trace the full circuit from input through critical units to output. This means identifying which input features trigger stepping in accumulator units, how gates control information flow to these units, and how the output layer reads their activity. The LSTM's ability to maintain two independent integration processes while selectively deploying them based on task demands suggests sophisticated routing mechanisms we haven't yet characterized. Understanding these would bridge the gap between knowing what computations occur and understanding how the network orchestrates them.

References

Arpit, D., Zhou, Y., Ngo, H., and Govindaraju, V. (2016). Why Regularized Auto-Encoders learn Sparse Representation? arXiv:1505.05561 [stat].

Bakaraju, V. and Konda, K. R. (2019). Only sparsity based loss function for learning representations. arXiv:1903.02893 [cs].

Cain, N., Barreiro, A. K., Shadlen, M., and Shea-Brown, E. (2013). Neural integrators for decision making: a favorable tradeoff between robustness and sensitivity. *Journal of Neurophysiology*, 109(10):2542.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2):179–211.

Goldman, M. S., Compte, A., and Wang, X. J. (2009). Neural Integrator Models. In Squire, L. R., editor, *Encyclopedia of Neuroscience*, pages 165–178. Academic Press, Oxford.

Gupta, K., Bastani, O., and Solar-Lezama, A. (2023). SPARLING: Learning Latent Representations with Extremely Sparse Activations. arXiv:2302.01976 [cs].

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Liu, X., Zhen, Z., and Liu, J. (2020). Hierarchical Sparse Coding of Objects in Deep Convolutional Neural Networks. *Frontiers in Computational Neuroscience*, 14. Publisher: Frontiers.

Sanchez, K. and Rowe, F. J. (2018). Role of neural integrators in oculomotor systems: a systematic narrative literature review. *Acta Ophthalmologica*, 96(2):e111–e118. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/aos.13307.

Smith, L. N. (2017). Cyclical Learning Rates for Training Neural Networks. arXiv:1506.01186 [cs].

Tinsley, C. J. (2008). Coding of distributed, topographic and non-specific representations within the brain. *Biosystems*, 92(2):159–167.

Werbos, P. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Zou, H. and Hastie, T. (2005). Regularization and Variable Selection Via the Elastic Net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320.

CHAPTER 4

Discussion and Future Directions

4.1 Research Problem and Questions

4.1.1 The Challenge of In-Context Learning

Navigating naturalistic environments requires an observer to continuously track statistical regularities that govern when and how environmental features change (Glaze et al., 2015; Yu et al., 2021). This challenge becomes particularly difficult when changes occur at multiple time scales, with some unfolding spontaneously over time while others are triggered by specific environmental events. Optimally responding to these challenges requires, for each environmental feature, not only detecting that changes have occurred, but also inferring which underlying processes caused them, all while observations remain partially observable. And despite the problem being ubiquitous in naturalistic environments, few experimental paradigms are designed for investigations into how biological and artificial systems meet these challenges. Previous work has examined temporal probability tracking and event-contingent learning in isolation, or how they apply to biological or artificial systems in isolation, preventing direct comparison of how these distinct forms of in-context learning interact.

To address this gap, we developed the Bouncing Ball task, a novel predictive inference paradigm where observers must estimate two latent parameters governing color changes on a trial-by-trial basis. The hazard rate controls spontaneous color changes and contingency controls color changes to velocity events. This paradigm allows us to ground behavioral outcomes of humans and neural networks to predictions from Bayesian inference, establishing a baseline comparison to all observers. The insight from our approach is that optimal in-context learning depends not only on inferential capacity, but on the ability to extract appropriate evidence from the different sensory streams, a process referred to in this thesis as the counting bottleneck. Understanding how biological and artificial systems are subject to this bottleneck provides a window into the computational principles that enable flexible behavior and the architectural constraints that limit it.

4.1.2 Research Problem

This dissertation investigated the computational and cognitive mechanisms that enable in-context learning of multiple temporal statistics in biological and artificial systems. By examining how humans and recurrent neural networks solve the Bouncing Ball task when compared to an ideal Bayesian observer, we sought to identify both universal computational principles and observer-specific constraints that shape in-context learning. Our investigation spanned multiple levels of analysis, from behavioral performance to the neural mechanisms that implement these computations. Specifically, we examined how observers represent estimates of multiple latent parameters from partially observable sequences, update these estimates mid-sequence, and integrate them to guide predictions. This work bridges normative theories of Bayesian inference with mechanistic accounts of neural implementation, using the counting bottleneck as a framework to explain learning capabilities. Through this approach, we demonstrate how computational constraints shape the in-context learning profiles observed across biological and artificial intelligence.

4.1.3 Research Questions

To systematically investigate the computational mechanisms and behavior underlying in-context learning, we organized our investigation around two primary research questions.

RQ1: How do humans and recurrent neural networks perform in-context learning of multiple statistics compared to optimal inference?

This question investigated whether both biological and artificial observers exhibited the following patterns of behavior predicted by Bayesian inference:

1. Time-dependent sensitivity for hazard rate.

2. Event-dependent sensitivity for contingency.

Our analyses confirmed that all observers displayed the expected signatures, establishing the Bouncing Ball task as a platform for evaluating in-context learning. Having validated these behavioral patterns, we quantified the degree of optimality across different observers and parameters.

We found that humans achieved only 15.5% of optimal hazard rate sensitivity but reached 47.7% of optimal contingency sensitivity. The most engaged humans (C2) achieved 17.2% of optimal hazard rate sensitivity but 78.9% of optimal contingency sensitivity. RNNs showed minimal learning with 21.5% and 10.2% respectively. LSTMs reached 50.3% and 80.7%, respectively, though three seeds failed to exceed RNN-level performance. The selective capabilities of the humans and LSTMs pointed to a computational constraint, which we call the counting bottleneck, where performance depends not on the ability to accumulate evidence but on extracting the appropriate counts from continuous sensory streams.

RQ2: What neural mechanisms enable successful in-context learning of multiple statistics?

This investigation demonstrated that LSTMs implement in-context learning by sparse representations (n=2/32) of the hazard rate and contingency estimates. These hazard rate and contingency units employ computational strategy of leaky integration with different triggers and updating schema. Hazard rate units displayed continuous temporal updating with discrete color-change driven accumulation, whereas contingency units strictly displayed discrete color-change driven accumulation. Targeted interventions on single units demonstrated causal behavioral control, with manipulations of hazard rate units affecting only temporal predictions and contingency units affecting only event-based predictions. This functional independence demonstrates an instance of how abstract Bayesian computations map onto concrete neural architectures.

Together, these two questions charted a path from validating the Bouncing Ball task as a testbed for in-context learning, through explaining selective capabilities via computational constraints, to finally revealing the minimal neural mechanisms sufficient for implementation. The following sections synthesize our findings across the levels of analysis, demonstrating how the counting bottleneck shapes (and constrains) in-context learning in both biological and artificial systems.

4.1.4 The Bouncing Ball Task

The Bouncing Ball task provides a controlled environment for studying in-context learning by tasking observers with predicting the color of a ball moving through an arena where a central gray filter (gray zone) occludes color. The mechanics of the color changes are controlled by the hazard rate for random color changes and contingency for bounce color changes. We use a combination of the color choice made by an observer, weighted by their confidence, to define a metric called the confidence-weighted choice (CWC). When traveling in straight paths through the gray zone, Bayesian inference predicts that as the

ball passes through the gray zone, the CWC should trend from change to no-change at a rate directly proportional to the rate of random color changes, or hazard rate. For a lower and higher hazard rate, this should manifest as a difference in the rate of increase through time, which we can define as the hazard rate sensitivity. On trajectories where the ball bounces off a wall, Bayesian inference predicts that the CWC should scale proportionally to the number of observed bounce color changes. This increase in CWC as the contingency condition increases reflects their sensitivity to the contingency task parameter.

Unlike previous paradigms that examine either temporal or event-based learning in isolation, the Bouncing Ball task requires tracking of both statistics simultaneously within each trial and can be administered to both biological and artificial systems. This multi-parameter design was essential for determining the counting bottleneck, as it requires insights from selective learning of one parameter over another. Through our investigation using the Bouncing Ball task, we uncovered the computational and behavioral mechanisms of in-context learning. The following sections synthesize these findings, demonstrating how computational constraints at the level of event extraction lead to the behavioral differences we observed.

4.2 Summary of Key Findings

Our investigation of in-context learning using the Bouncing Ball task demonstrated patterns of selective capability across humans and neural networks, with performance being dependent on the evidence being tracked. Here, we summarize the key findings that address each research question.

4.2.1 Addressing RQ1: Behavioral Signatures of In-Context Learning

Through Bayesian inference, we can define the expected behavioral signatures of in-context learning on the Bouncing Ball task. All tested observers (humans, LSTMs, and RNNs) demonstrated statistically significant hazard rate and contingency sensitivity ($p < 0.05$), confirming the predictions for in-context learning. Confidence-weighted choices (CWC) for hazard rate increased as time in the gray zone increased, this rate of increase scaled for higher hazard rate values, and CWC for contingency scaled with the probability of color changes due to velocity changes. Taken together, these outcomes validated the Bouncing Ball task as an effective paradigm for in-context learning, and demonstrated that both biological and artificial observers extract statistical regularities from partially observable sequences.

4.2.2 Addressing RQ1: Selective Learning

Despite all observers showing statistically significant sensitivity, they varied in the degree to which they were sensitive. Using the ideal Bayesian observer as the benchmark, the full human participant group achieved 15.5% of optimal hazard rate sensitivity and 47.7% of optimal contingency sensitivity. This difference was more pronounced in the C2 subgroup (n=42), who achieved the near-optimal contingency sensitivity of 78.9% while maintaining a similar hazard rate sensitivity of 17.2%. The C1 subgroup (n=36) adopted a "no-change" strategy, achieving 13.6% and 11.4% of optimal sensitivity for hazard rate and contingency respectively.

Neural networks showed different patterns of sensitivity, where the vanilla RNNs demonstrated uniformly poor performance at 21.5% (hazard) and 10.2% (contingency) of optimal, matching C1 human performance on both parameters. LSTMs achieved the highest hazard rate sensitivity of the observers at 50.3% of optimal and matched the C2 responders at 80.7% optimal contingency sensitivity. However, this level of contingency

sensitivity was not consistently achieved between all model seeds, as 30% failed to develop sensitivity beyond RNN-level performance. The difference between RNN and LSTM performance, despite being matched in number parameters, highlights the architectural requirements for in-context learning. Inspecting the LSTM's architecture gives theoretical grounding for what enables its learning capacity, however empirical observations show contingency sensitivity comes much later in training, if at all (70% of seeds) when compared to hazard rate sensitivity (<5% of total epochs).

The random velocity change experiment added an additional trial type to the Bouncing Ball task to test contingency sensitivity on velocity-change driven color changes that do not occur on a wall. Participants showed statistically equivalent sensitivity to wall bounces (0.231 ± 0.059) and random velocity changes (0.191 ± 0.055) and high correlations ($r = 0.592$, $p < 0.001$). This generalization from predictable wall bounces to unpredictable random velocity changes demonstrates that color changes do not need an anticipatory component to discern the different contingency values.

4.2.3 Addressing RQ2: Neural Implementation

Mechanistic analysis of the LSTM states demonstrated sparse representations for each of the task parameters. Of the 32 units available, critical units selective for hazard rate and contingency were found in pairs across all models that demonstrated high levels of sensitivity. These units implemented variations of the leaky-integrator computational primitive but with task-relevant triggers. Hazard rate units showed continuous leak behavior and clear activity stepping upon discrete random color changes. Contingency units showed clear activity stepping on bounce events, where positive deflections occurred on contingent color changes and negative deflections occurred when no changes occurred. Both units displayed clear signed of evidence accumulation based on the variables relevant for the task.

Causal validation through targeted interventions confirmed color predictions are func-

tionally dependent on critical unit activity. Administering graded interventions to units of one task parameter led to dose-dependent shifts in color-predictions selective for only that parameter. Hazard rate interventions had no effect on bounce-dependent color predictions, and contingency interventions had no effect on time-dependent color predictions. Single-unit interventions revealed that cell states causally drive predictions, while hidden state interventions showed no effect suggesting its role as a readout rather than a behavioral driver.

4.3 Interpretation and Synthesis

4.3.1 A Platform for Investigating In-Context Learning

The Bouncing Ball task was developed as a platform to investigate the effects of in-context learning in both humans and recurrent models. By grounding the outcomes we found in the predictions laid out by the ideal Bayesian observer, we validated that the task successfully captures in-context learning across multiple parameters. This validation establishes proof of concept for a new family of paradigms that can probe in-context learning along several axes. Adding spatial contingencies would test tracking of temporal, event-based, and location-based statistics simultaneously. Hierarchical dependencies with other features, such as multiple color sequences, would better approximate real-world non-stationarity. Adding more balls with independent parameters would probe capacity limits and resource allocation. The IBO framework can be extended to accommodate any of these additions.

4.3.2 Strategy Heterogeneity

The clustering analysis revealed two subgroups of participants whose behavioral responses to the task go beyond simple performance differences. The C1 group's conservative strategy of responding "no change" to 91.4% of trials actually yielded higher overall on

the task accuracy (69.5%) than C2's more balanced approach (63.7%). This wasn't a failure to understand the task but rather the rational (and more optimal) response to the high base rates of the task. Despite C1's strategy yielding better accuracy, 54% of participants chose to attempt inference accuracy for engagement with the underlying statistical structure. For the purposes of the discussion, when making comparisons to models or trying to understand human behavior, we tend to refer to the C2 group specifically because they attempted inference, and therefore will better reflect human in-context learning capabilities.

4.3.3 The Counting Bottleneck Hypothesis

Synthesizing across the behavioral and mechanistic findings, our results point to a single hypothesis that provides a mechanistic account for:

1. Selective learning of task parameters in participants (low hazard rate sensitivity, high contingency sensitivity)
2. Preservation of contingency sensitivity in the random velocity change experiment
3. LSTM asymmetric training time to achieve above-RNN contingency sensitivity (<5% for hazard rate, random if at all for contingency)

We hypothesize that these outcomes stem not from limitations in evidence accumulation machinery, but the ability to appropriately extract countable events from the task sequence. The initial insight comes from recognizing that the ideal Bayesian observer demonstrates that both hazard rate and contingency require identical evidence accumulation machinery, and the variation in complexity between representing one parameter or another comes from determining the counts rather than performing the evidence accumulation. In this framing, the Bayesian update represents the last operation before a prediction, whereas determining whether an observable constitutes evidence can be arbitrarily complex. Each of the behavioral outcomes above become coherent when interpreted through challenges

to counting unique to each observer.

On any specific video, velocity changes create discrete events that are both salient and segment the continuous stream into countable units. Additionally, their corresponding color changes are similarly salient and countable. Each bounce and color change represents a memorable event, occurring perhaps 3-10 times per trial. Humans excel here because discrete events align with our cognitive architecture for attention and working memory, where we naturally parse experience into episodes, and bounces provide exactly these episodic boundaries (Radvansky and Zacks, 2017; Khemlani et al., 2015). This would explain why C2 responders achieve 78.9% of optimal contingency performance: when events are countable, humans can perform evidence accumulation to their fullest. For humans, it is clear that contingency sensitivity arises due to perceptual systems that are geared towards operating with discrete and countable events.

This story reverses for hazard rate estimation, where participants now need to not just to track the countable number of color changes but also the hundreds of frames where the change did not occur. Maintaining a long running count of non-events requires sustained effort but would be akin to estimating elapsed time, an ability that has been shown to be effective and accurate in humans for shorter timescales. However, this accuracy is easily disrupted by additional tasks of increasing complexity, which the Bouncing Ball task definitely demands of the participants. Even the C2 responders, who demonstrate clear evidence accumulation abilities through their contingency performance and attempts at inference through their balanced sensitivity and specificity, only reach 17.2% of optimal hazard rate sensitivity. And because they are able to estimate the counts of color changes (due to contingency sensitivity) and those color changes do not have to be predictable (nonwall contingency sensitivity), we can rule out the main variables relating to color change counts for hazard rate, leaving the estimates for time as the most-likely culprit.

When interpreting the results of recurrent models, the effect of the counting bottleneck is reversed and the high volume of frames enhances learning for hazard rate. Since

LSTMs do not rely on attention mechanisms to selectively choose frames of interest, they fully process each frame, allowing them to quickly gain an estimate for the non-changes, which can be combined with the countable random changes, leading to rapid sensitivity. However, this equal weighting across frames induces a signal detection problem when learning contingency. With only 1-10 bounces per video, this would provide a mechanistic explanation for why contingency sensitivity emerges later in training, if it all: In the early phases of training, where the loss is still high, gradient updates generated by contingent color changes are quickly undone in the extended periods of time between each bounce. It is only later in the training phases, when enough updates are successfully committed, that contingency can be properly learned.

The counting bottleneck shows that, for in-context learning, the variation in performance we observed on the Bouncing Ball task most likely points to variations in evidence assessment rather than accumulation.

4.4 Limitations

4.4.1 Ecological Validity

While the Bouncing Ball task successfully isolated temporal and event-based in-context learning, there are some aspects worth consideration. The counting bottleneck hypothesis proposes an explanation for why the human participants might inherently not be suited to display hazard rate sensitivity. An alternative explanation could be that the gray zone distance is not long enough for temporal counting systems to take effect. As it is currently implemented, the gray zone occupies the middle third of the screen which only translates to 1-2 seconds where ball is completely hidden. Extending the length of the gray zone may lead to sensitivity levels that better reflect the true capacity of human inference.

4.4.2 Modeling And Analysis

Training Setup

The Bouncing Ball task was designed to be a visual predictive inference task for both humans and models. And while the humans have to perform visual predictive inference, this does not fully extend to the models as they perform inference over a vector of positions and colors. This reduced setup kept training times low and allowed for rapid iteration, but raises questions about the scope of the comparisons that can be made between them and humans.

Model Size and Sparsity

The LSTMs discussed in this thesis were set to be the smallest size that continue to learn the task within reasonable time frames. Increasing the number of parameters beyond 16 units led faster training time but did not affect the final range of hazard rate and contingency sensitivities (above a threshold of around 32). While we discovered sparse representations for the hazard rate and contingency after running the elasticnet pipeline on these models, it is likely that we would see a more distributed representation in a model with 128 units, as the model would not need to maximize resourcefulness given the abundance of representational units.

4.5 Future Directions

4.5.1 The Leaky-Integrator Circuit

Chapter 3 of this dissertation identified critical units that encoded hazard rate and contingency, where each had a corresponding pair of units. We then demonstrated that these units implement a leaky-integrator, where for hazard rate the unit displayed a steady decay in activity and discrete stepping on random color changes. Contingency units showed a similar setup, where they showed the stepping response selectively to bounce

events and would step in one direction for bounce-dependent color changes and in the other for bounces without a color change. We next showed these units were functional by performing targeted interventions and demonstrated dose-dependent changes in color change predictions. While this analysis made clear that the LSTM implements a leaky integrator where the unit encodes the estimate of the relevant statistic, it is unknown how the other components of the LSTM participate this circuitry.

An inspection of the LSTM gates provides some insights into how it could be implemented. The forget gate acts upon the cell state in a manner to remove activity, which would be a natural place for the leak to be implemented. A simple experiment to test this would be to let the model observe a Bouncing Ball sequence where no color changes occur, and ablate the effects of the forget gate. If the leak is implemented in the forget gate, then the unit's activity should remain stable in time rather than decay. A similar setup could be used to probe contingency where we ablate forget gate and see if there is a down-step in the units activity. Identifying where all the components of the integrator lie would paint the complete picture of how these task statistics are encoded in the LSTM.

4.5.2 The Role of Attention

Under the counting bottleneck hypothesis, between groups of observers that are capable of implementing Bayesian inference, performance differences will largely reflect differences in capacity for evidence accumulation. Human participants were found to have abilities more suited to counting discrete events, whereas LSTMs were demonstrated more suitability to learning hazard rate, evidenced by their rapid sensitivity to the hazard rate. In humans, this is mostly likely to be an attentional effect, where attending to no-changes continuously over time is cognitively taxing. In LSTMs, this effect is due to the lack of an attentional system, leading to every timestep being processed in its entirety, drowning out gradient updates by infrequent, discrete events, such as bounce color changes. More modern architectures, such as the transformer, have seen wide usage

due to their attention mechanisms, allowing for richer representations to be learned when working with sequences (Vaswani et al., 2017). Examining the behavior of the transformer could shed light on some of its computational capabilities, especially in comparison to the LSTM. While it's possible for models to implement a different computational primitive from the LSTM (leaky-integrator), the manner in which they do so would shed light on how different connectionist architectures converge on the same computations.

References

Glaze, C. M., Kable, J. W., and Gold, J. I. (2015). Normative evidence accumulation in unpredictable environments. *eLife*, 4:e08825. Publisher: eLife Sciences Publications, Ltd.

Khemlani, S. S., Harrison, A. M., and Trafton, J. G. (2015). Episodes, events, and models. *Frontiers in Human Neuroscience*, 9. Publisher: Frontiers.

Radvansky, G. A. and Zacks, J. M. (2017). Event boundaries in memory and cognition. *Current Opinion in Behavioral Sciences*, 17:133–140.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention Is All You Need. *arXiv:1706.03762 [cs]*. arXiv: 1706.03762.

Yu, L. Q., Wilson, R. C., and Nassar, M. R. (2021). Adaptive learning is structure learning in time. *Neuroscience & Biobehavioral Reviews*, 128:270–281.

Supplementary Material

Supplementary Figures for Chapter 2

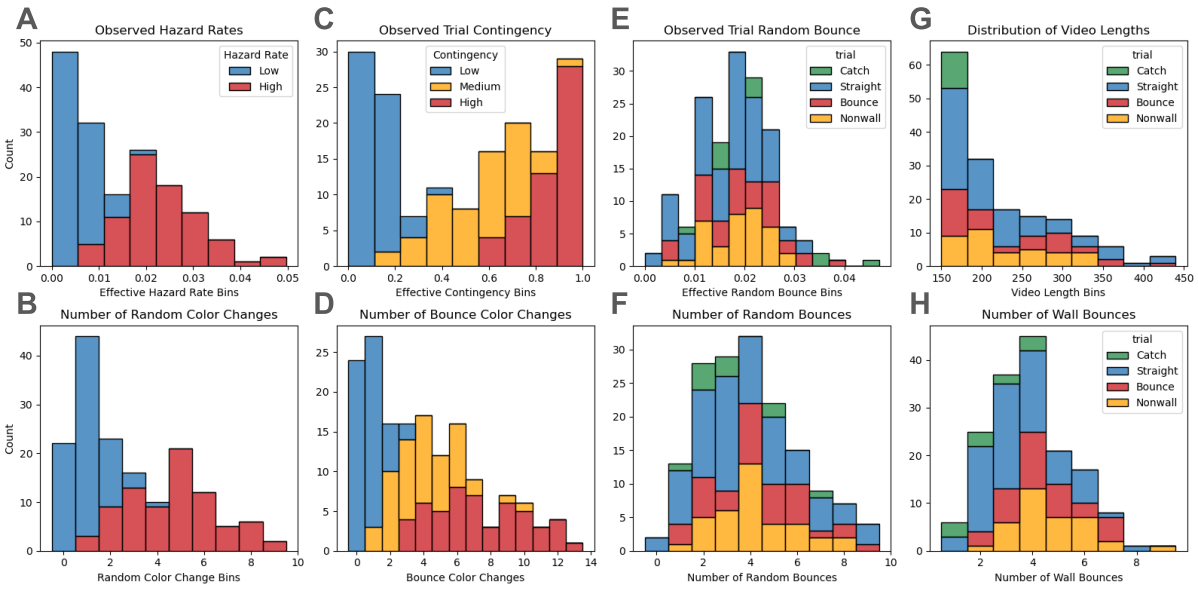


Figure S1: Random Velocity Change Videos Overall Statistics. Participant Task Videos Overall Statistics. **(A)** Effective hazard rates for each video colored by hazard rate. Computed as the ratio between total random color changes (including those in the grayzone) divided by the number of eligible time steps (total number of non-bounce time steps). **(B)** Histogram of total random color changes per video colored by hazard rate. **(C)** Effective contingency for each video colored by contingency. Computed as the ratio of bounce color-changes divided by number wall bounces. **(D)** Histogram of number of bounce color changes per video colored by contingency. **(E)** Effective random velocity change rate for each video colored by trial type. Computed as the ratio between total random velocity changes (including those in the grayzone) divided by the number of eligible time steps (total number of non-wall bounce time steps). **(F)** Histogram of total random velocity changes per video colored by trial type. **(G)** Histogram of video lengths in frames colored by trial type. **(H)** Histogram of number of bounces per video colored by trial type.

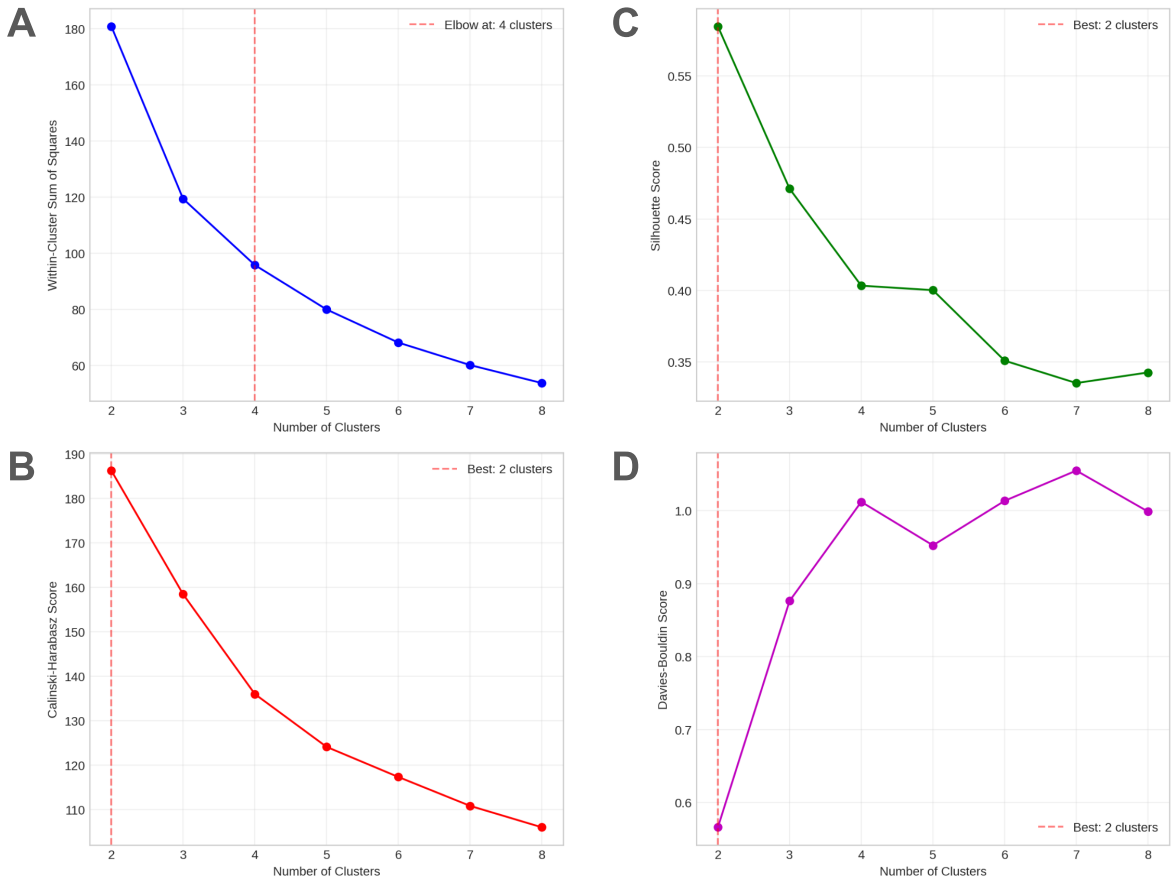


Figure S2: Participant Cluster Solution Evaluation. Four clustering validation metrics were computed for $k = 2$ through $k = 8$ clusters using hierarchical clustering with Ward's linkage on participants' normalized confusion matrices. **(A)** Within-Cluster Sum of Squares shows an elbow at $k = 4$ (red dashed line), suggesting four clusters. **(B)** Calinski-Harabasz Index maximum value at $k = 2$ (186.5, red dashed line), indicating maximum between-cluster variance relative to within-cluster variance with two clusters. **(C)** Silhouette Score peak at $k = 2$ (0.585, red dashed line) and decreases monotonically thereafter, indicating optimal separation and cohesion with two clusters. **(D)** Davies-Bouldin Score minimum at $k = 2$ (0.56, red dashed line), indicating minimal overlap and optimal separation with two clusters.

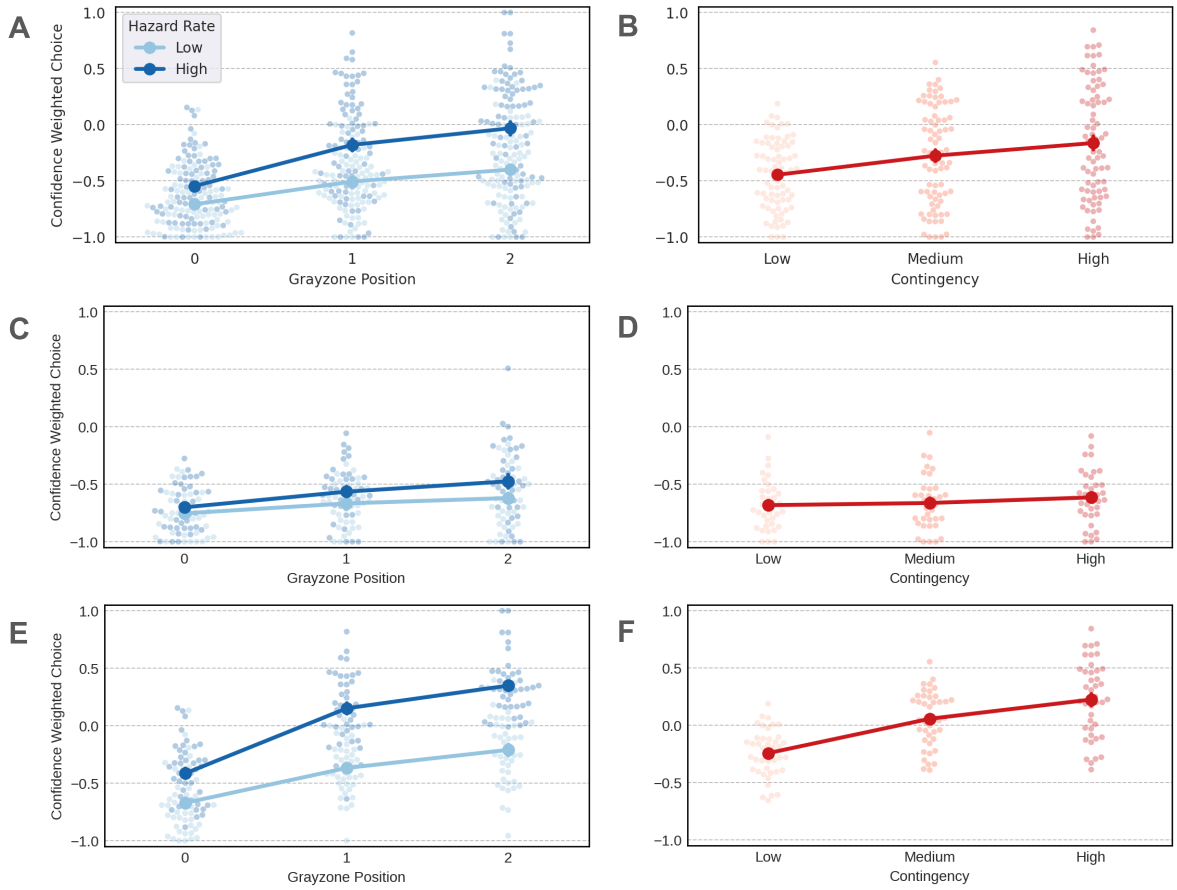


Figure S3: Participant and Subgroup Confidence-Weighted Choices. Confidence-weighted choice (CWC) combines binary color change decisions (-1 for no change, +1 for change) with confidence ratings, plotted across three grayzone positions (1, 11, and 22 timesteps) for straight paths (left) and three contingency levels (0.05, 0.5, 0.95) for bounce paths (right). **(A)** All Participants CWC on straight path trials. **(B)** All Participants CWC on bounce path trials. **(C)** C1 responders CWC on straight path trials. **(D)** C1 responders CWC on bounce path trials. **(E)** C2 responders CWC on straight path trials. **(F)** C2 responders CWC on bounce path trials.

Supplementary Figures for Chapter 3

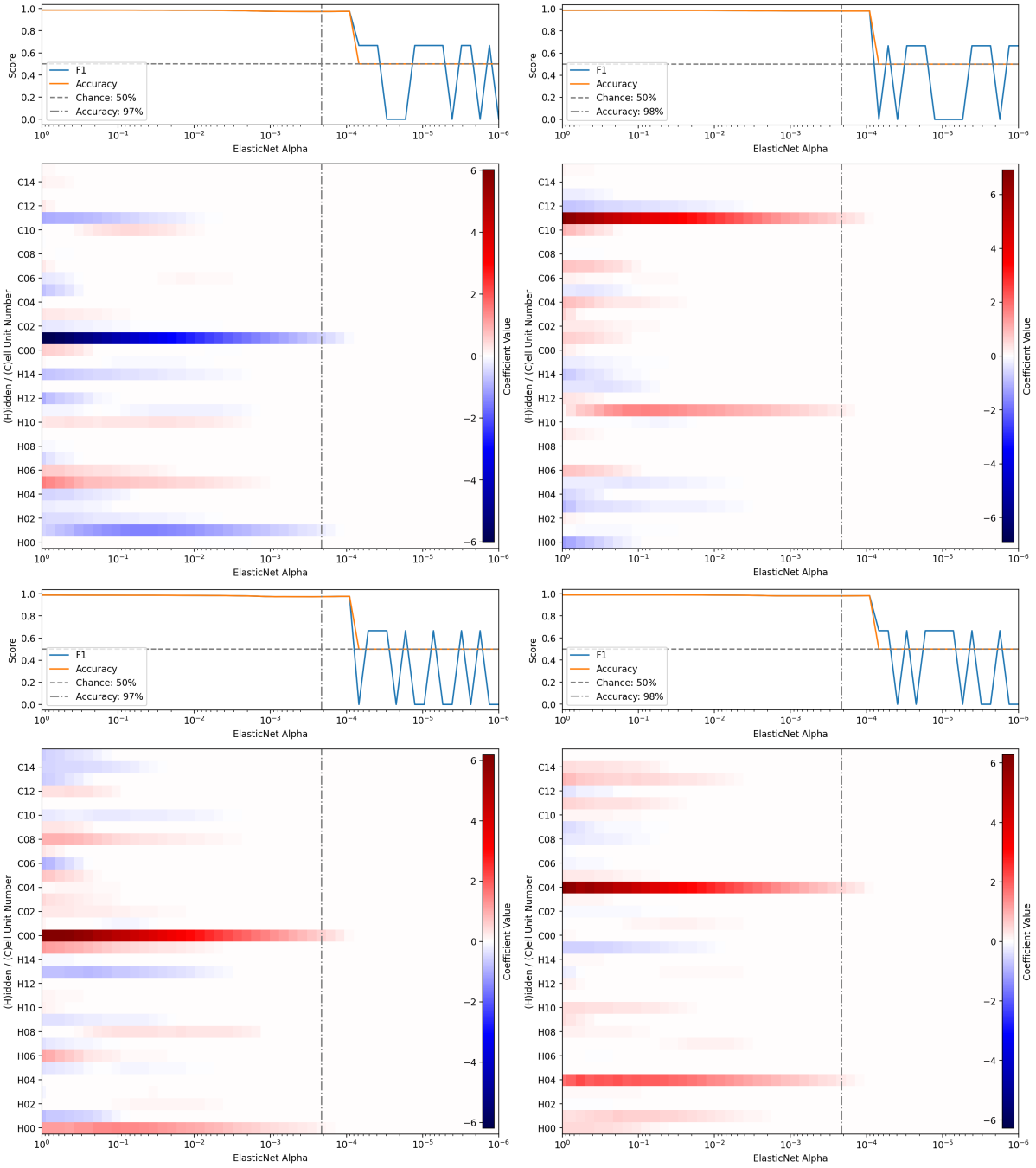


Figure S4: Additional Critical Units for Hazard Rate.

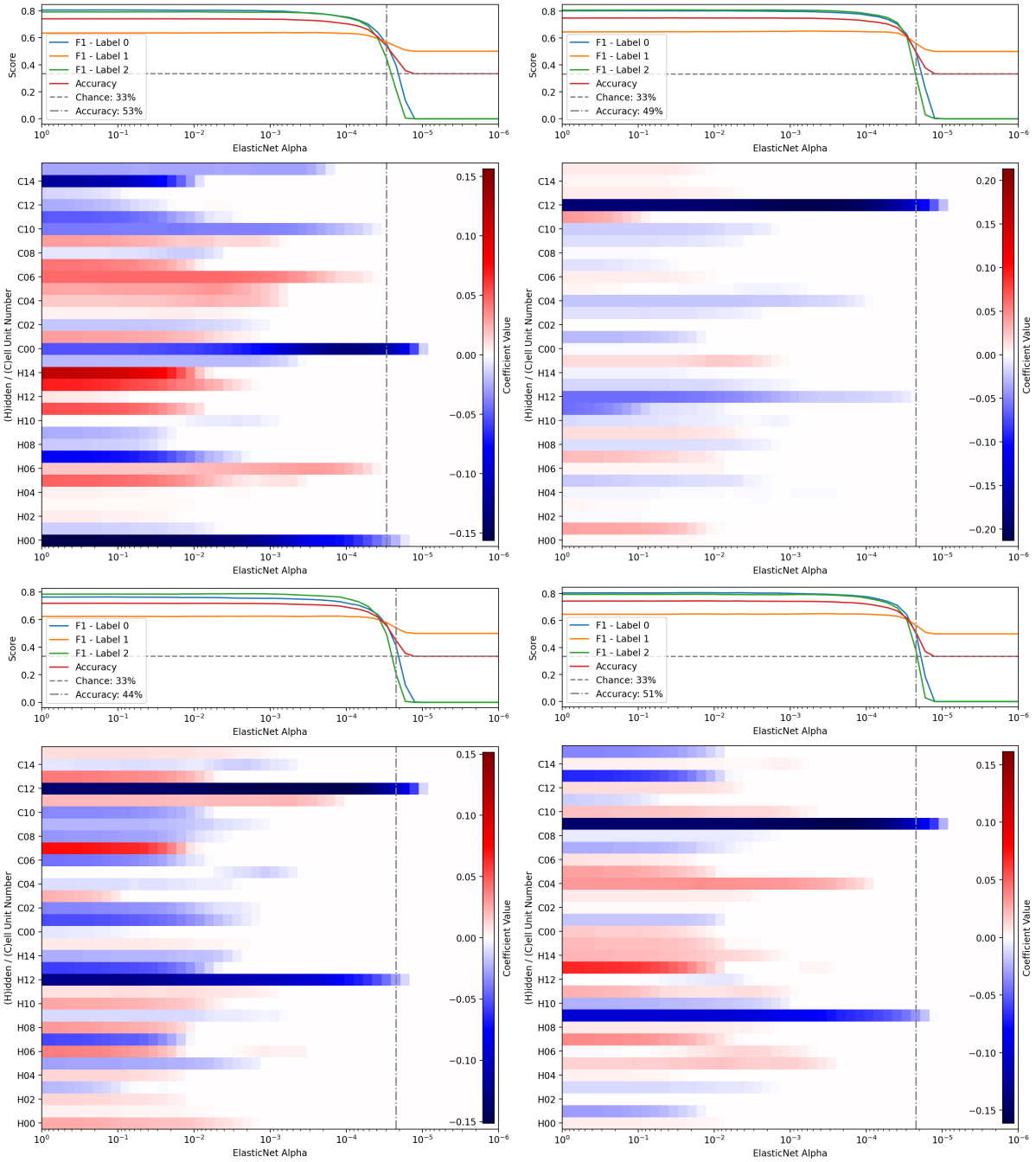


Figure S5: Additional Critical Units for Contingency.

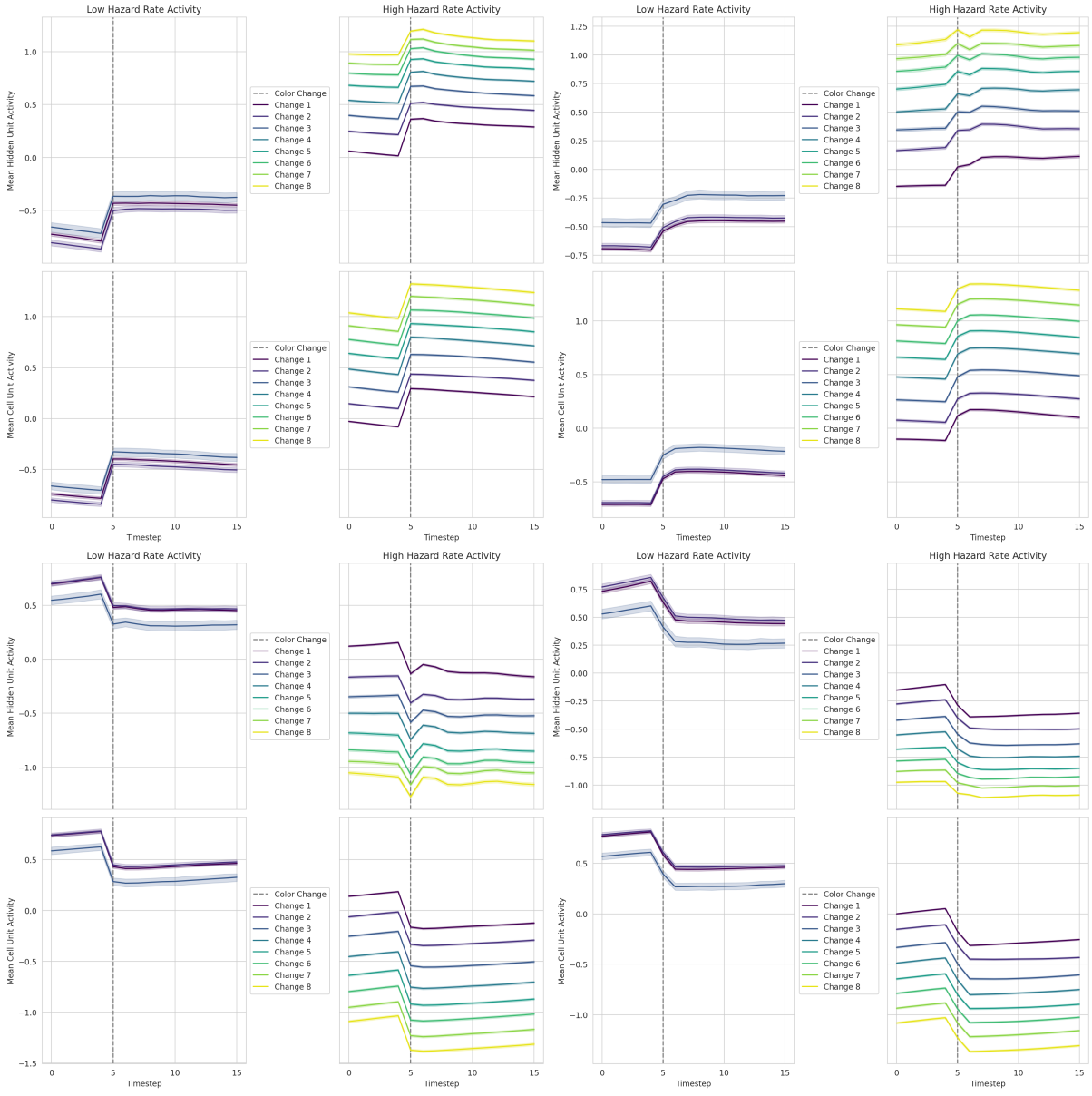


Figure S6: Additional Tuning Profiles for Hazard Rate Critical Units.

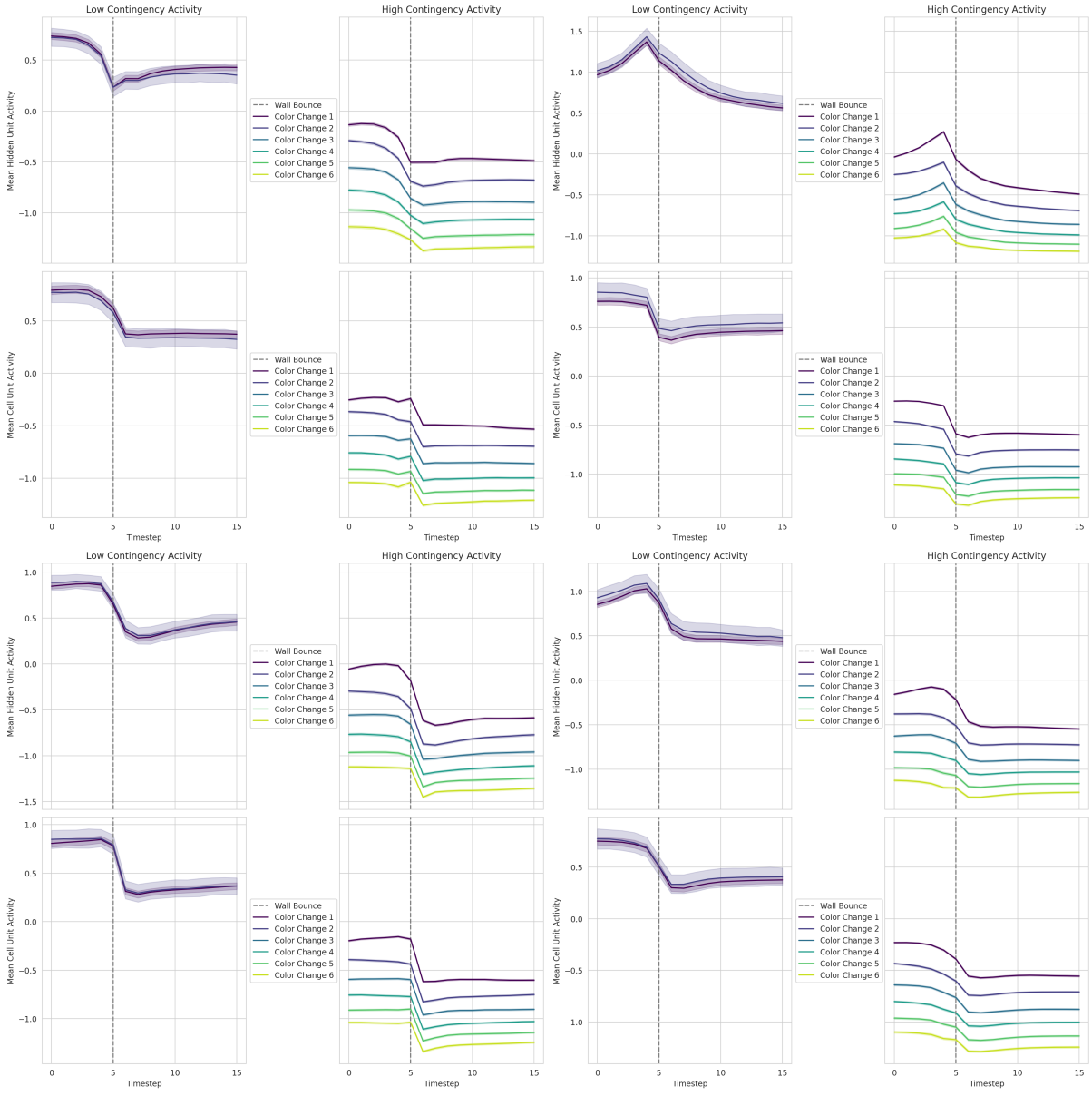


Figure S7: Additional Tuning Profiles for Contingency Critical Units on Color Changes.