Integrative Approaches in Machine Learning and Biology

By

AVINASH AVINASH
DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Physics

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Mark S. Goldman, Chair

Daniel L. Cox

Randall C. O'Reilly

Committee in Charge

2024

In dedication to my wife.

Contents

Abstract

This dissertation delves into the rich intersection of machine learning (ML) and biology, a field marked by significant progress yet full of opportunities for further breakthroughs. The work presented here is divided into three main parts. The first two focus on employing ML to investigate reward-based learning in animals and to study protein-protein interactions within viruses. The third part shifts focus to explore the development of robust computer vision models, drawing inspiration from biological insights.

In the first part, we apply the reinforcement learning (RL) framework to gain insights into reward-based learning in animals. We explore how key neural circuits and neurotransmitter systems, specifically the prefrontal cortex, ventral striatum, and the dopaminergic pathways, contribute to the implementation of RL algorithms to learn the association of actions with delayed outcomes, known as the credit assignment problem. We first identify a distinct pattern of neural activity in the prefrontal cortical inputs to the ventral striatum: activity that is sequential over time and selective to a given choice. We then use computational modeling to show how these inputs provide an effective state representation for the ventral striatum, enabling it to calculate accurate value signals for each choice at any given time point. This is demonstrated through the implementation of two neural circuit models of reinforcement learning, where reward prediction error drives learning either by inducing rapid synaptic plasticity or by altering neural dynamics. Additionally, we test and confirm our circuit model predictions experimentally through direct manipulation of the input neurons to the ventral striatum.

In the second part, we conduct two computational studies of SARS-CoV-2 to understand its spike protein interactions and its implications in viral transmission and immune response evasion. To achieve this we employ two key computational tools: molecular dynamic simulations and AlphaFold2, an advanced deep learning model designed for predicting protein structures. The study is divided into two main parts. The first part examines the biophysical properties of the SARS-CoV-2 Omicron variants compared to the wild type and Delta variants, analyzing the spike protein binding to (i) the ACE2 receptor protein, (ii) antibodies from all known binding regions, and (iii) the furin binding domain. Our findings indicate that the Omicron variant shows reduced binding to the ACE2 receptor, but increased immune evasion, consistent with preliminary observations. The second part delves deeper into the interactions between the Furin Cleavage Domain (FCD) of SARS-CoV-2 variants and other coronaviruses with the furin enzyme. Here, we demonstrate that the Delta variant exhibits the strongest possible binding with the furin enzyme, and we identify key sequences, both observed and unobserved, that could exhibit similar binding strengths.

In the final part, we explore how integrating biological insights, particularly from the primary visual cortex area V1, can improve the robustness of Convolutional Neural Networks (CNNs) against various image corruptions. For this purpose, we utilize VOneNet, a hybrid CNN containing a model of V1 as the front-end, followed by a standard trainable CNN architecture. We first observe that different variants of the V1-inspired model exhibit performance trade-offs for different corruptions. Building on this, we develop a new model using an ensembling technique, which combines multiple individual models with different V1-inspired variants. This model effectively harnesses the strengths of each individual model, leading to significant improvements in robustness across all corruption categories. Further, we demonstrate that knowledge distillation can help compress the knowledge in the ensemble model into a single, more efficient V1-inspired model. Overall, we demonstrate that by merging the unique strengths of various neuronal circuits in V1 we can significantly enhance the robustness of CNNs against a wide array of perturbations.

Acknowledgments

I extend my sincerest appreciation to my advisor, Mark Goldman, whose guidance and expertise have been the cornerstone of my academic and personal development throughout my PhD journey. His mentorship has not only shaped my research path but also enriched my understanding of the vast field of neuroscience and machine learning. His influence extended beyond traditional research mentorship, markedly improving my abilities in communication, presentation, and writing. These skills have been vital in my development as both a scholar and an effective communicator. His willingness to be accommodating has provided me with the flexibility to explore my interests deeply and navigate the challenges of academia with confidence.

A special note of gratitude goes to Daniel Cox, not only a committee member but a fundamental figure in my academic journey. His instrumental role in introducing me to biophysics and our collaborative efforts on protein folding have been invaluable. I am also thankful to Randall O'Reilly for his insightful feedback and constructive criticism throughout the thesis process.

The environment in the Goldman lab has been exceptional, and I am grateful to my lab mates for this. Specifically, Ben Lankow, Yiheng Wang, and Jiacheng Xu have contributed significantly through discussions on neuroscience, which has expanded my knowledge significantly.

Similarly, my time in the Cox lab has been greatly enriched by my peers, especially Muhammad Zaki Jawaid. Our discussions on the protein folding project have been both enlightening and invigorating.

My collaborative work has been greatly enriched by the expertise and knowledge shared by my collaborators, particularly Ilana Witten and Tiago Marques. Their contributions have significantly enhanced my understanding of neuroscience and machine learning, and for this, I am very grateful.

I extend my deepest appreciation to my family for their endless love and support. My wife, Payton Hovey, has been my constant source of love, support, and inspiration. Her presence has not only been pivotal for the completion of my PhD but has also immensely enriched my life. My parents, Baidya Nath and Sharmila Prasad, along with my sister, Anwesha, have consistently supported me in all my pursuits. Their sacrifices and love have immensely contributed to my education and personal development. My uncle, Subodh Gupta, deserves a special mention for his significant contribution to my education and his encouragement towards my pursuit of science. My gratitude also extends to all of my extended family, whose support has been a cornerstone of my journey. My in-laws, Caroline and Stephen Hovey, along with Payton's entire extended family, have warmly welcomed me, ensuring I feel at home in a new country, for which I am very thankful. Furthermore, I want to extend my gratitude to my dog, Wylla, whose calming presence has provided endless comfort throughout my PhD.

Lastly, I want to express my heartfelt gratitude to all my friends. The moments we have shared playing cricket, tennis, and pickleball, along with the birthday celebrations and trips, have provided much-needed balance in my life. Their support and camaraderie have been a constant source of happiness and encouragement.

This thesis is not only a reflection of my efforts but a testament to the collective support, guidance, and encouragement of everyone mentioned and many more unmentioned. Thank you all for being part of this journey.

CHAPTER 1

Introduction

In today's world of rapid technological advancements and our deepening understanding of biological systems, we are witnessing a significant intersection of machine learning (ML) and biology. Over the past decade, machine learning has showcased remarkable achievements, driven in part by its emulation of certain computational processes of the human brain. Deep neural networks, inspired by the brain's neural architecture, have enabled computers to perform a range of tasks, from driving vehicles to creating art and music, tackling complex mathematical problems, and even conversing in a human-like manner. On the flip side, machine learning with its generative capabilities and the ability to understand complex data and extract hidden patterns has led to significant breakthroughs in biology, such as solving the complex protein-folding problem and advancing the development of brain-computer interfaces. Despite these achievements, the intersection of ML and biology still holds vast, untapped potential.

This dissertation delves into this fascinating interplay between ML and biology, divided into three main sections. The first two sections focus on the application of ML to enhance our understanding of reward-based learning in animals and to contribute to the computational study of the SARS-CoV-2 virus. The final section investigates how biological insights, specifically from the primary visual cortex, can be leveraged to improve the robustness of computer vision models. In this chapter, we provide a detailed introduction to each section.

1.1. Part 1: Reinforcement learning in animals

1.1.1. Background and Motivation

Introduction to Reinforcement Learning

Reinforcement Learning (RL) is a machine learning framework that is inspired by behavioral psychology and focuses on how agents learn to make decisions in an environment. It is based on the concept of agents interacting with an environment, learning optimal behaviors through trial and error, and adapting their actions based on rewards and punishments. This learning process mimics how humans and animals learn from their experiences, constantly adjusting their actions to achieve better outcomes.

Basic Concepts in RL

Agent and Environment: The agent is the decision-maker. The environment includes everything the agent interacts with.

States: States, denoted by S_t, represent the agent's situation or context within the environment at the current time t.

Actions: Actions, denoted by A_t, are the decisions an agent makes at time t.

Policy: A policy $\pi(A_t|S_t)$ represents the agent's probability for selecting action A_t given the current state S_t.

Rewards: Rewards, denoted by R_t, are the feedback from the environment based on the actions taken by the agent at time t.

Value function: The value function is given by the discounted sum of all future rewards, thus providing an estimate of how beneficial it is for the agent to be in a given state, or how good an action is, considering all future rewards. There are two main types of value functions:

State-value function - The state-value $V_t^\pi(s)$ of a state s under a policy π is the expected cumulative future reward starting from that state and following policy π:

$$V_t^\pi(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s \right] \tag{1.1}$$

where $\mathbb{E}_\pi$ denotes the expected value given that the agent follows policy π and γ denotes the rate at which future rewards are discounted.

Action-value function - The action-value $Q_t^\pi(s, a)$ is the expected cumulative future reward from taking an action a in a given starting state s and thereafter following policy π

$$Q_t^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a \right] \tag{1.2}$$

A typical RL process (Figure 1.1) involves an agent interacting with its environment through a series of actions. At any given time t, the agent is in a state S_t, and it takes an action A_t. As a consequence of this action, the agent transitions to a new state S_{t+1} and receives reward feedback R_{t+1} from the environment. The goal of the agent is to learn a policy π that maximizes the cumulative reward (value) over time.

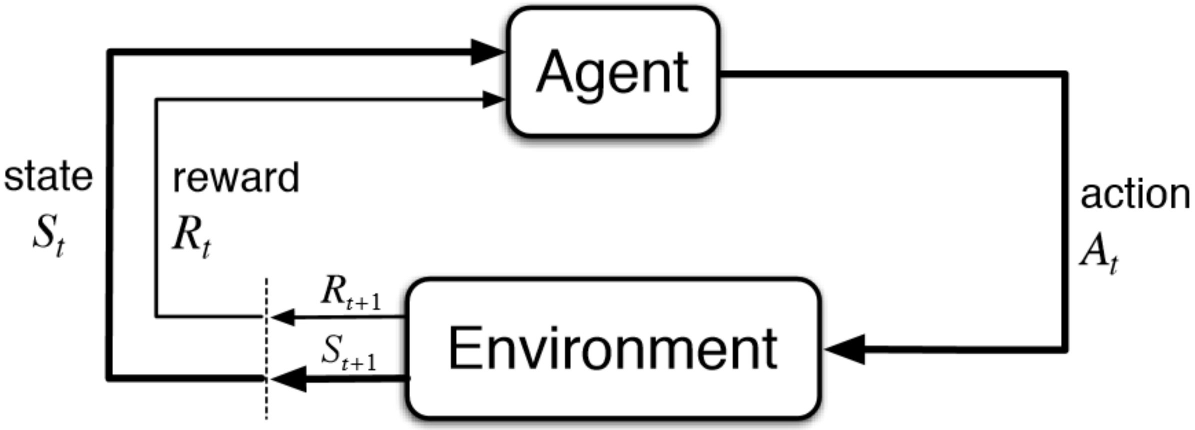


FIGURE 1.1. **RL cycle**. Adapted from "Reinforcement learning: An introduction" by Richard S. Sutton and Andrew G. Barto, 2018, MIT press, p. 48.

RL Methods

RL algorithms can be broadly categorized into three main types: policy-based methods, value-based methods, and actor-critic methods. Each method employs different strategies for learning the optimal policy to maximize cumulative future rewards.

Value-based method: The objective of value-based learning is to learn the value function for each state $V(s)$ or state-action pair $Q(s, a)$ in order to derive the optimal policy $\pi(a|s)$ indirectly. For example, given the state-value function $Q(s, a)$, the policy $\pi(a|s)$ is estimated as:

$$\pi(a|s) = \arg\max_{a} Q(s, a) \tag{1.3}$$

Policy-based method: The objective of policy-based methods is to model and learn an optimal policy $\pi(a|s)$ directly that maximizes the expected return from each state.

Actor-critic method: A popular class among the policy-based methods is the actor-critic method, which combines the benefits of both policy-based and value-based approaches. It consists of two main components: the actor, which is responsible for selecting actions given the current state, essentially learning the policy $\pi(a|s)$, and the critic, which evaluates these actions by computing the value function. The actor aims to learn a policy that optimizes future rewards, while the critic's feedback on the actor's choices is crucial for optimizing the policy.

Reward Prediction Error

In RL, learning is driven by reward prediction errors (RPE), which quantify the discrepancy between expected future rewards and actual future rewards. This error signal is vital for updating the value estimates and refining policies. Mathematically, the general form of RPE can be represented as:

$$\text{RPE} = G_t - V(S_t) \tag{1.4}$$

where $G_t = \sum_{t=0}^{\infty} \gamma^t R_{t+1}$ represents the actual sum of discounted future rewards received after time t. This formulation captures the essence of RPE but presents a practical challenge: the experienced sum of future rewards G_t can only be fully determined at the end of a task.

Given the sequential and dynamic nature of most RL tasks, waiting until the end to calculate RPE is impractical for learning purposes. To address this issue, we can use the Bellman formulation of the value function, which enables estimating future rewards at any point in a task based on current knowledge. In this formulation, the value function for a given state s (Equation 1.1) is written recursively as follows:

$$V_t^\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma V^\pi(S_{t+1})|S_t = s]. \tag{1.5}$$

Here R_{t+1} denotes the immediate reward received after transitioning from state S_t to state S_{t+1}. This equation enables computing an updated expectation of future rewards in terms of the newly experienced reward R_{t+1}. Replacing G_t in the original formulation of RPE (Equation 1.4) with this updated value function, we obtain the Temporal Difference (TD) RPE:

$$\text{TD RPE} = R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \tag{1.6}$$

The TD RPE measures the difference between the old estimated value of S_t, $V(S_t)$, and the new value estimate, $R_{t+1} + \gamma V(S_{t+1})$, after observing the new state S_{t+1} and receiving feedback R_{t+1}. It serves as a more practical form of RPE enabling learning at every step of the task.

RL in Continuous State Spaces

In tasks with continuous state spaces, representing value functions and policies becomes challenging due to the infinite number of possible states. To manage this complexity, basis functions offer a powerful solution by providing a finite set of features that can approximate the value functions or policies over the continuous space. These functions can be approximated as a linear combination of the basis functions or through more complex functions that can be approximated by deep neural networks (DNNs).

Consider, for example, a set of basis functions $\phi_i(s)$. The state-value function can be approximated linearly as

$$V(s) \approx \sum_{i=1}^{N} w_i \phi_i(s), \tag{1.7}$$

where w_i are the weights and N is the number of basis functions, or non-linearly using DNNs as

$$V(s) \approx f(\phi_1(s), \phi_2(s), ..., \phi_N(s); \theta) \tag{1.8}$$

where f represents the DNN with network weights θ. The linear approach is computationally simple and provides ease of interpretation, whereas DNNs offer the flexibility to approximate highly complex mapping that would be infeasible to represent with linear models.

Action-value functions and policies can be approximated in a similar manner. The use of basis functions in representing value functions and policies allows for scalable and efficient learning in environments with continuous state and action spaces.

RL in Neuroscience

In neuroscience, RL has emerged as an important framework for studying reward-based learning in animals. It is primarily focused on understanding how the brain learns from rewards and adapts behavior accordingly. The RL algorithm is believed to be mediated by various neural circuits and neurotransmitter systems. The nucleus accumbens (NAc), a part of the ventral striatum, plays an important role in reward-based learning and decision-making [1,2,3,4,5,6,7,8,9,10,11,12,13,14] and is believed to carry a neural representation of value. This region notably receives glutamatergic inputs from multiple areas [15,16,17,18,19,20,21,22,23,24,25], including the prefrontal cortex, which is thought to represent the agent's state, among other things. Additionally, the NAc receives dopaminergic projections from the ventral tegmental area (VTA), which are thought to signal reward prediction errors [26].

A fundamental question in the study of RL within neuroscience is how the brain associates actions with delayed outcomes (the credit assignment problem). Given the roles of these brain regions, they are presumed to play a significant part in this mechanism. Yet several questions remain. First, it is unclear whether the glutamatergic projections to NAc actually convey a representation of the current decision, which forms the basis for the state representation in this problem. Moreover, assuming these inputs do reflect the current decision, the task remains to develop a biologically plausible neural circuit model based on the observed inputs that implement an RL algorithm to solve the credit assignment problem.

1.1.2. Contribution

In Chapter 2, we provide both experimental and computational insights on the role of these brain areas in solving the credit assignment problem. The study involved neural recordings by our experimental collaborators of the glutamatergic inputs to the NAc in mice performing a probabilistic reversal learning task. In this task, mice are presented with a choice between two levers: left or right. Selecting one lever yields a high reward probability of 70%, while choosing the other offers a low reward probability of 10%. The assignment of high and low reward probabilities to the levers changes randomly after a random number of trials. The objective for the mice in this task is to consistently choose the lever with the high reward probability to maximize their overall rewards. The experiments demonstrate that the glutamatergic inputs from the prefrontal cortical area PL to NAc (PL-NAc inputs) encode choice-selective sequential activity, i.e., the activity is selective to a given choice, with neurons firing sequentially in time in a set order. This sequence bridges the time from making a choice to receiving reward feedback, and continues until the beginning of the next trial.

Driven by these findings, we aimed to understand how these inputs contribute to solving the credit assignment problem. Through computational modeling, we find that the distinct choice-selective and sequential patterns in PL-NAc input activity provide an accurate state representation (temporal basis functions) necessary to compute the value at any given time. We demonstrated this through a neural circuit model of reinforcement learning that relies either on synaptic plasticity or neural dynamics.

Synaptic plasticity model: In this model, the PL-NAc inputs serve as temporal basis functions $f_i^R(t)$ and $f_i^L(t)$ corresponding to the right-choice and left-choice preferring neurons. These inputs are linearly combined within the NAc to estimate the value of making a left $V_L(t)$ or right $V_R(t)$ choice as follows:

$$\begin{aligned} V_R(t) &= \sum_{i=1}^{n_R} w_i^R(t) f_i^R(t) \\ V_L(t) &= \sum_{i=1}^{n_L} w_i^L(t) f_i^L(t) \end{aligned} \tag{1.9}$$

where the overall estimated value at time t, $V(t)$, is given by the sum over both the left and right neurons as

$$V(t) = V_R(t) + V_L(t) \tag{1.10}$$

Here, n_R and n_L are the number of right- and left-preferring choice-selective neurons respectively, and $w_i^{R,L}$ are the weights between the i^{th} PL neuron and the NAc, which multiply the corresponding basis functions. Learning in this model occurs in the PL-NAc weights $w_i^{R,L}$ using the TD learning algorithm. In this algorithm, the weights $w_i^{R,L}$ are adjusted based on the TD RPE signaled by dopamine neurons, aiming to minimize the TD RPE through the loss function:

$$L_{TD}(t) = \delta(t)^2 = (r(t) + \gamma V(t) - V(t - \Delta))^2 \tag{1.11}$$

where $\delta(t)$ represents the TD RPE at time t, $r(t)$ is the reward received at the current time t, and $V(t)$ and $V(t-\Delta)$ represent the value estimates at current and previous times, respectively.

The update rule for the synaptic weights, $w_i^{R,L}$, are then given by gradient descent on the TD loss function:

$$\begin{aligned} \Delta w_i^{R,L}(t) &= -\alpha \cdot \frac{\mathrm{d}L_{TD}(t)}{\mathrm{d}w_i^{R,L}} \\ &= \alpha \cdot \delta(t) \cdot f_i^{R,L}(t) \end{aligned} \tag{1.12}$$

Therefore, learning in the PL-NAc synapses is driven by the correlation of the dopamine RPE signal with the PL-NAc input activity. By adjusting the synaptic weights to capture the appropriate value of each choice, the model learns to solve the task.

Neural dynamics model: In contrast to the synaptic plasticity model, the dopamine error signal drives the dynamics of a recurrent neural network (RNN), rather than fast synaptic plasticity, to update the values and corresponding selection of actions. The initial learning of the neural network's synaptic weights is based on an actor-critic reinforcement method [27], enabling the model to learn the dynamics of the task. However, once the weights are learned initially, the synaptic weights remain fixed during the performance of the task, with the dopamine RPE serving only to alter neural dynamics.

Similar to the synaptic plasticity model, the input PL-NAc activity serves as the temporal basis functions for value computation. However, in this case, the temporal basis function are non-linearly combined within the NAc to generate value using a recurrent neural network (RNN):

$$V(S_t;\theta_v) = g(f_1^R(t), f_2^R(t), ..., f_{n_R}^R(t), f_1^L(t), f_2^L(t), ..., f_{n_L}^L(t);\theta_v), \tag{1.13}$$

where g represents the RNN with network weights θ_v. This forms the critic network. Similar to the synaptic plasticity model, the critic parameters θ_v are updated to accurately learn the value function by minimizing the TD loss:

$$\begin{aligned}\Delta\theta_v(t) &= -\alpha_v \cdot \nabla_{\theta_v} L_{TD}(t;\theta_v) \\ &= \alpha_v \cdot \delta(t) \cdot \nabla_{\theta_v} V(S_t;\theta_v),\end{aligned} \tag{1.14}$$

where α_v is the learning rate for the critic.

Unlike the synaptic plasticity model, which leverages a value-based method, this model adopts an actor-critic framework and thus incorporates both an actor and a critic network. The actor network models the policy using an RNN with network weights θ_π.

The actor's objective is to maximize the expected future returns $J(t;\theta_\pi)$ of the policy π parameterized by θ_π, defined as:

$$J(t;\theta_\pi) = \mathbb{E}_{\pi_{\theta_\pi}}[G_t] \tag{1.15}$$

where G_t is the cumulative sum of rewards starting from time step t.

The gradient of $J(t;\theta_\pi)$ with respect to the policy parameters θ_π can be computed using the policy gradient theorem [28]:

$$\nabla_{\theta_\pi} J(t;\theta_\pi) = \mathbb{E}_{\pi_{\theta_\pi}}[G_t \cdot \nabla_{\theta_\pi} \log \pi(A_t|S_t;\theta_\pi)]. \tag{1.16}$$

Since G_t represents the cumulative rewards received from choosing action A_t in state S_t in the above equation, it can be replaced by $Q(S_t, A_t)$:

$$\nabla_{\theta_\pi} J(t;\theta_\pi) = \mathbb{E}_{\pi_{\theta_\pi}}[Q(S_t, A_t) \cdot \nabla_{\theta_\pi} \log \pi(A_t|S_t;\theta_\pi)]. \tag{1.17}$$

However, directly using $Q(S_t, A_t)$ in Equation 1.17 can lead to high variance in the gradient estimates, particularly in environments with stochastic dynamics or rewards, potentially resulting in slow and unstable learning. To address this, $Q(S_t, A_t)$ can be replaced with the ”advantage function” $A(S_t, A_t) = Q(S_t, A_t) - V(S_t)$. This function measures the relative benefit of taking action A_t in state S_t compared to the average value of the state. By doing so, it reduces the variance in the gradient estimates without altering the expected gradient, since the value of the state is not dependent on the policy parameters.

The advantage function can be closely approximated by the TD RPE $\delta(t)$, since $Q(S_t, A_t)$ can be approximated by $r(t) + \gamma V(S_{t+1})$. Substituting $Q(S_t, A_t)$ with the TD RPE in Equation 1.17, the gradient of the expected future returns $J(t; \theta_\pi)$ with respect to the policy parameters θ_π is represented as:

$$\nabla_{\theta_\pi} J(t; \theta_\pi) = \mathbb{E}_{\pi_{\theta_\pi}}[\delta(t) \cdot \nabla_{\theta_\pi} \log \pi(A_t | S_t; \theta_\pi)]. \tag{1.18}$$

The update equation for the actor parameters is given by stochastic gradient ascent on the expected future returns $J(t; \theta_\pi)$:

$$\begin{aligned} \Delta\theta_\pi(t) &= \alpha_\pi \cdot \nabla_{\theta_\pi} J(t; \theta_\pi) \\ &= \alpha_\pi \cdot \delta(t) \cdot \nabla_{\theta_\pi} \log \pi(A_t | S_t; \theta_\pi), \end{aligned} \tag{1.19}$$

where α_π is the learning rate for the actor.

1.2. Part 2: Computational study of SARS-CoV-2

1.2.1. Background and Motivation

The ongoing global health crisis caused by Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) has created an urgent need for a comprehensive understanding of its virology and pathogenesis. SARS-CoV-2 belongs to the broad family of viruses known as coronaviruses. Coronaviruses infect humans, other mammals, including livestock and companion animals, and avian species.

SARS-CoV-2 (Figure 1.2A) is a positive-sense single-stranded RNA virus, characterized by club-like spikes projecting from its surface and an unusually large RNA genome. The genome of SARS-CoV-2 encodes four major structural proteins (Figure 1.2A): the spike (S) protein, nucleocapsid (N) protein, membrane (M) protein, and envelope (E) protein. The N protein encapsulates the RNA genome, while the S, E, and M proteins collectively form the viral envelope.

Central to SARS-CoV-2's entry into host cells is its S glycoprotein [29, 30], which comprises two functional subunits (Figure 1.2B): the S1 subunit, containing the receptor-binding domain (RBD) and N-terminal domain (NTD), and the S2 subunit, responsible for mediating the fusion of viral and host cell membranes.

The SARS-CoV-2 S protein binds to the angiotensin converting enzyme 2 (ACE2) receptor with high affinity at the surface of host cells, initially through the S1 RBD [29, 30] (Figure 1.2B). Following this initial binding, S1 is released, enabling the S2 subunit to facilitate membrane fusion [29, 30]. This process depends on the activation of the S protein through cleavage at the Furin Cleavage Domain (FCD) [29, 30] (Figure 1.2B), located between the S1 and S2 subunits, both by Furin and by transmembrane serine proteases, especially TMPRSS2. This cleavage is believed to enable the fusion of the viral capsid with the host cell to permit viral entry [29, 30]. The role of the NTD in this process is still unclear but it is believed to play an important role in escaping antibodies [31].

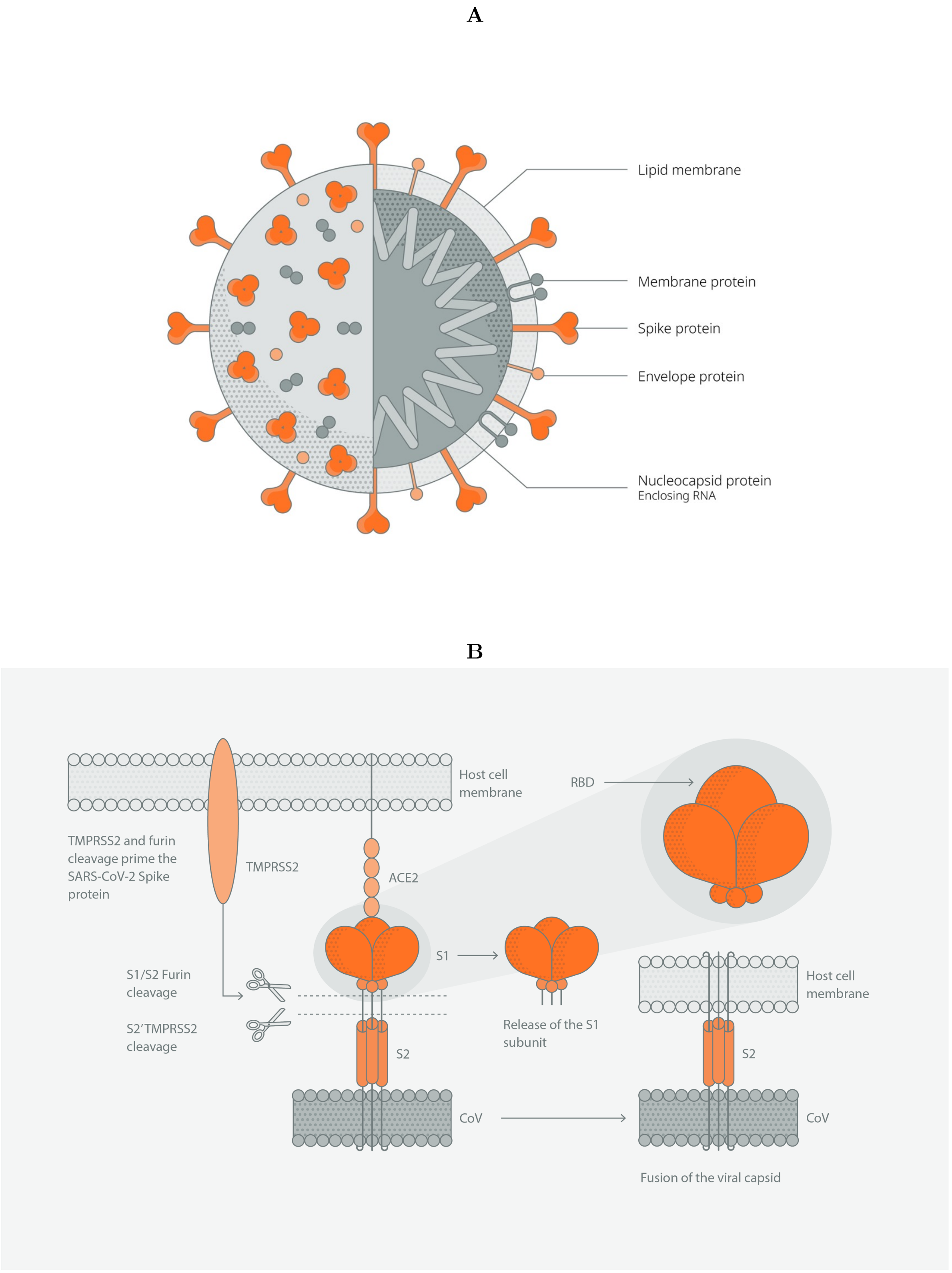


FIGURE 1.2. **SARS-CoV-2.** A. Structure B. Entry mechanism. [1]

Although the basic mechanism of SARS-CoV-2 entry into host cells is established, the nuances of how this process varies among different variants of the virus remain unclear. Specifically, there is a lack of clarity regarding how variations in binding impact the virus's severity and its ability to spread. Additionally, the critical role of the Furin Cleavage Domain (FCD) in viral entry necessitates a deeper understanding of its structure and interaction with the furin enzyme. Yet, a detailed study of the FCD's binding characteristics has been challenging for several reasons. Firstly, the FCD is situated within a rapidly fluctuating random coil region of the spike protein which has not been resolved by structural probes. Secondly, because the furin enzyme rapidly cleaves the spike protein at the FCD, it complicates efforts to determine the FCD-Furin bound structure. As a result, there is a lack of structural information regarding the FCD in its bound state.

1.2.2. Contribution

To tackle the challenges outlined previously, this dissertation consists of two computational studies, detailed in chapters 3 and 4, focusing on the biophysics of the SARS-CoV-2 variants and related viruses.

Chapter 3. **SARS-CoV-2 omicron spike simulations: broad antibody escape, weakened ACE2 binding, and modest furin cleavage**: This chapter delves into the biophysical properties of the SARS-CoV-2 Omicron variants, comparing them with the original wild type and Delta variants. We conduct a comprehensive analysis of the binding strengths in various interactions: RBD-ACE2, RBD-antibody (AB), FCD-Furin, and NTD-antibody for these SARS-CoV-2 variants. In line with initial observations specific to the Omicron variant, our findings reveal (i) a significant increase in antibody evasion across all regions compared to the wild type and Delta variants, (ii) an intermediate level of FCD binding to furin between the wild type and Delta variants, and (iii) a reduced affinity for the ACE2 receptor compared to both the wild type and Delta variants.

Chapter 4. **Computational study of the furin cleavage domain of SARS-CoV-2: delta binds strongest of extant variants**: This chapter presents a comprehensive analysis of the binding interactions between the Furin Cleavage Domain (FCD) of SARS-CoV-2 variants and other

[1] Source: https://www.abcam.com/content/structural-and-functional-mechanism-of-sars-cov-2-cell-entry

coronaviruses with the furin enzyme. We discover that the Delta variant exhibits the strongest possible binding with the furin enzyme. The study also identifies critical sequences, both observed and unobserved, that could exhibit similar binding strengths. Additionally, we predict various binding modes between the FCD and the furin enzyme, while verifying our predictions through comparison with existing crystal structures of furin inhibitors and furin complexes.

To generate the protein-protein interactions (bound structure) between the spike proteins of SARS-CoV-2 variants (and other viruses) with human cell receptors, we utilized two key computational tools: molecular dynamics simulations and AlphaFold2. While the interactions involving the ACE2-RBD, RBD-antibody, and NTD-antibody of the wild-type SARS-CoV-2 have been previously characterized, the structure of the complex formed between the FCD of the wild-type SARS-CoV-2 and the Furin enzyme remains unknown. To address this gap, we utilized AlphaFold2 to predictively model the FCD-Furin complex. To simulate the interactions for other viral sequences, we initialized molecular dynamics simulations with the corresponding wild-type bound structures, modified to incorporate mutations reflecting the differences between the wild-type (WT) sequence and the given viral sequence.

Molecular dynamics

Molecular dynamics is a widely used computational tool to study the dynamics of large atomic systems. This method calculates the positions of atoms at different time points by solving their equations of motion. The process begins with a precise description of the potential energy that captures all atomic interactions within the system. From this potential energy, MD calculates the forces acting on the atoms using the gradients of the energy. By applying Newton's second law, these forces determine the atoms' accelerations, which, in turn, dictate their motion. A numerical integration algorithm then solves the equations of motion, using initial positions and velocities, to simulate the trajectory of each atom across discrete time steps.

The accuracy and effectiveness of MD simulations hinge on several initial conditions. These include:

1. **Initial Coordinates**: The starting positions of all the atoms are required to generate the trajectory coordinates of the system.

2. **Force-Field**: The potential energy description of atoms, known as the force-field, is crucial. It defines the energy landscape of the system through non-bonded interactions like Lennard-Jones and electrostatic Coulomb potentials, as well as bonded interactions like bonded potentials and angular potentials. The primary force field model used in this work is AMBER14 [32], which models the potential energy [33] as:

$$\begin{aligned}E_{\text{total}} = &\sum_{i\in\text{bonds}} K_r(r_i - r_{i,eq})^2 + \sum_{i\in\text{angles}} K_{\theta_i}(\theta_i - \theta_{i,eq})^2 \\ &+ \sum_{i\in\text{dihedrals}} \sum_n \frac{V_{i,n}}{2}[1 + \cos(n\omega_i - \gamma_{i,n})] \\ &+ \sum_{j=1}^{N-1} \sum_{i=j+1}^{N} \left\{ \left[\left(\frac{A_{ij}}{R_{ij}}\right)^{12} - \left(\frac{B_{ij}}{R_{ij}}\right)^{6} \right] + \frac{q_i q_j}{4\pi\epsilon_0 R_{ij}} \right\}.\end{aligned}$$

Here, K_r and K_{θ_i} specify the force constants for bond stretching and angle bending, respectively. The variables r_i and θ_i denote the current bond length and bond angle between the two atoms or the two bonds, respectively, with $r_{i,eq}$ and $\theta_{i,eq}$ as their equilibrium values. The dihedral angle, ω_i, describes the clockwise angle between two planes, each of which is determined by a set of three atoms. These two planes share a common bond, formed by the two atoms that are part of both sets. $V_{i,n}$ is the amplitude of the torsional barrier for the n^{th} dihedral rotation. n represents the periodicity of the dihedral potential, indicating the number of energy minima (or maxima) present within a 360° rotation. $\gamma_{i,n}$ is the phase shift in the dihedral potential, determining the angular position of the energy minima. The parameters A_{ij} and B_{ij} are the parameters for the Lennard-Jones potential, determining the strength of the van der Waals forces. R_{ij} measures the distance between the ith and jth atoms. Lastly, q_i and q_j are the partial electrostatic charges on atoms i and j, respectively.

The first term computes the energy between atoms that are covalently bonded. This term uses a harmonic potential, which is an accurate approximation when atoms are near their equilibrium bond lengths. The second term accounts for the energy due to the spatial arrangement of electron orbitals in covalent bonds. The third term, known as torsional energy, is related to the energy resulting from the twisting of bonds. The fourth term captures the non-bonded energy between all pairs of atoms, consisting of van der Waals interactions as well as electrostatic interactions. The van der Waals interactions are characterized using a potential that closely resembles the Lennard-Jones potential. For electrostatic (Coulomb) interactions, the model assumes that the atomic charges, resulting from protons and electrons, can be well approximated as single point charges.

3. **Solvent Model**: The choice of solvent model can significantly impact the simulation, particularly in systems where solvent interactions are important. The way the solvent is represented (explicitly or implicitly) affects the simulation. Explicit solvent modeling includes individual solvent molecules, while implicit modeling uses a mathematical representation to simulate the solvent effect. In this work, we explicitly modeled water as the solvent. We used the TIP3P [34] model of the water force fields.

4. **Ensemble Conditions**: Conditions such as temperature, pressure, volume, and energy define the statistical ensemble for the simulation, influencing the system's behavior and properties. For our simulations, we specified the temperature to be 298 K (approximately 25°C, which is room temperature) and the pressure to be 1 atm. These conditions define an NPT ensemble (constant number of particles, pressure, temperature), which is a statistical ensemble commonly used in molecular dynamics simulations to mimic real-life biological or chemical environments.

5. **Time Step Choice**: The time step choice in MD simulations is critical. It should be small enough to capture the dynamics accurately but not so small as to make the computation impractical. Typically, the time step is a fraction (at least a fifth) of the smallest period of oscillation in the system, often 1-2 femtoseconds (fs). In our simulations, the equations of motion were integrated with a time step of 1.25 fs for bonded interactions and 2.5 fs for non-bonded interactions.

AlphaFold2

This method [35] leverages machine learning to predict protein structures with remarkable accuracy. The model architecture (Figure 1.3) can be divided into three main components:

1. **Data Preprocessing Pipeline**: This process involves the extraction of data from protein databases, focusing on multiple sequence alignments (MSA) and structural templates relevant to the given protein sequence. MSAs are compilations of sequences from multiple proteins that are

aligned to identify regions of similarity or difference, created by searching large protein databases to find similar sequences to the target protein sequence. They provide insights into the functional, structural, or evolutionary relationships between the sequences. Templates are known protein structures that may share structural similarities with the target sequence. They help provide an initial guess for the target protein structure.

By integrating MSA and template information, the method generates two representations: MSA and pair representations. The MSA representation captures correlations between the protein residues (i.e., individual amino acids) using information about parts of the sequence that are more likely to mutate, while the pair representation utilizes both the input sequence and the templates to estimate the distances between residues in the protein's structure.

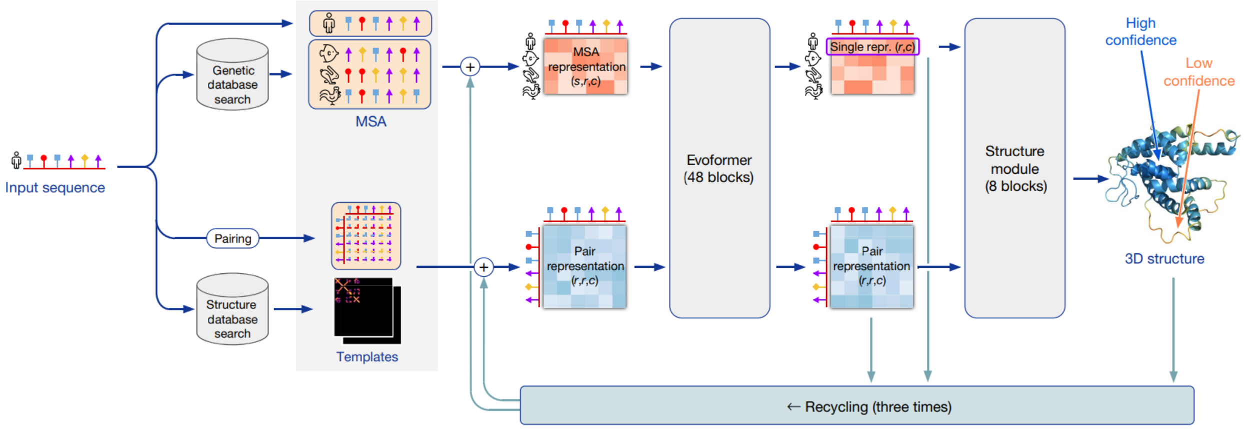

Input sequence
Genetic database search
MSA
Pairing
Structure database search
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)

FIGURE 1.3. **AlphaFold2 model architecture.** Arrows show the information flow among the various components described in this paper. Array shapes are shown in parentheses with s, the number of sequences; r, the number of residues; c, the number of channels. Adapted from "Highly accurate protein structure prediction with AlphaFold" by John Jumper, Richard Evans, and others, 2021, Nature, 596, p. 584.

2. **Evoformer**: This component refines the MSA and pair representations. It uses a transformer neural network. The key ingredient in this network is the attention mechanism. The objective of the attention mechanism is to identify which parts of the input are more important for the objective of the neural network, i.e., to identify which parts of the input it should pay attention to. The Evoformer consists of 48 transformer blocks, refining the MSA and pair representations iteratively. To refine the MSA representation, the network computes attention between residues in

a given sequence (row-wise attention) to identify which residues in the sequence are more related to each other and attention across sequences (column-wise attention) to identify which sequences are more important. The pair representation is refined using a triangular self-attention mechanism which ensures that predicted distances between amino acids can be realistically embedded in three-dimensional space by enforcing the triangle inequality principle from geometry.

3. **Structure Module**: This component generates the final three-dimensional structure of the protein. It models proteins as a residue gas with each amino acid being modeled as a triangle representing the backbone atoms, floating in space and moved by the network to form the structure. A key feature of the Structure Module is the Invariant Point Attention (IPA) module, which models rotations and displacements of the residues based on an attention mechanism that is invariant to translations and rotations. AlphaFold 2 operates end-to-end, continually refining its models of the protein through a feedback loop. Outputs from the Evoformer and the Structure Module are fed back into the process for further refinement, enhancing the connection between pairwise distance predictions and the 3D structure.

1.3. Part 3: Brain-inspired CNN

1.3.1. Background and Motivation

Introduction to CNNs

The introduction of Convolutional Neural Networks (CNNs) has been one of the most important developments in the field of computer vision. The structure of a CNN (Figure 1.4) is designed to imitate the hierarchical pattern recognition observed in the human visual cortex. This structure enables CNNs to automatically learn and generalize from image data, which is critical for tasks such as image classification and object detection.

The initial layer of a CNN is the input layer (Figure 1.4), which receives the raw pixel data from an image. Following this are the convolutional layers (Figure 1.4), which utilize filters, or kernels, to perform convolutions over the image. This produces feature maps that identify local patterns like edges and textures. Each convolutional layer is followed by a nonlinear activation function (Figure 1.4), such as the Rectified Linear Unit (ReLU), to enable the network to learn more complex patterns. The ReLU function transforms the input by rectifying it to ensure non-negativity:

$$\mathrm{ReLU}(x) = \max(0, x) \tag{1.20}$$

where x represents the input to the ReLU function.

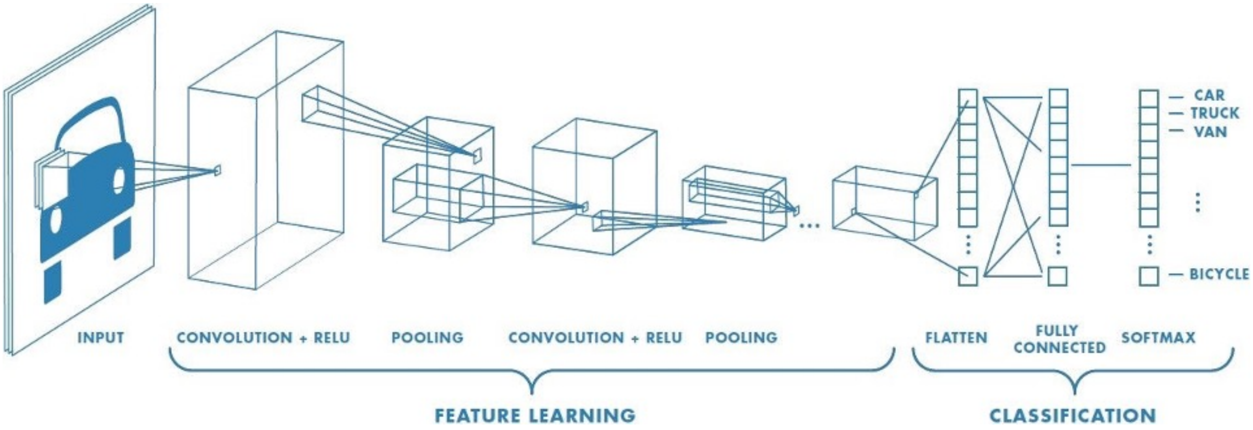


Figure 1.4. **CNN architecture.** Example CNN for object classification. [2]

After these are the pooling layers (Figure 1.4), which reduce the spatial size of the feature maps, decreasing parameters and computations, and help prevent overfitting. Pooling layers apply a downsampling operation, such as max pooling, which selects the maximum value within a defined window that slides over the input.

Deeper convolutional layers extract increasingly abstract features, which are then transformed into a 1D vector by a flattening layer (Figure 1.4). This vector is then fed through fully connected layers. In classification tasks, the final outputs, also known as the logits, of the fully connected layers are subsequently transformed into class probabilities by the softmax activation function:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{1.21}$$

where x_i represents the logits for class i. The softmax activation ensures that the ultimate output is a set of probabilities corresponding to each class.

Through these convolution, non-linearity, pooling, and fully connected layers, CNNs achieve high accuracy in image-based tasks. The architecture captures local and global patterns in an image, leading to effective recognition and classification.

Robustness in CNNs

Despite the progress made by CNNs, they still face significant robustness issues. These networks are notably vulnerable to adversarial examples [36,37,38]: inputs subtly modified to mislead the neural network into making dramatically incorrect predictions. This weakness is particularly concerning in areas like autonomous vehicles and medical diagnostics. Moreover, CNNs often struggle with common image corruptions [39, 40, 41] such as blurring, weather effects, or digital distortions, reducing their reliability in practical situations.

Recent efforts to improve the robustness of CNNs, particularly against commonplace corruptions, have been a key focus of research [41, 42, 43, 44, 45, 46, 47, 48, 49, 50]. The current state-of-the-art model for common image corruptions (DeepAugment+AugMix) [43] uses an image-to-image

[2] Source: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

network for adding perturbations to input images along with augmentation mixing. The image-to-image network transforms an original image into a modified version (augmented image) by applying various perturbations, effectively simulating a range of corruption types. Through augmentation mixing, where different perturbations are combined in varying proportions, this process generates a diverse set of augmented images. This exposes the model to diverse conditions and helps improve its robustness. Other data augmentation techniques include adding Gaussian noise [45] to small image patches to improve robustness. However, these methods have notable limitations. For instance, augmenting with Gaussian noise can enhance robustness but might adversely affect performance on unaltered images [45]. Additionally, applying Gaussian noise can create susceptibilities to low-frequency corruptions [46]. This suggests a compromise where increasing resilience to one type of corruption might reduce resistance to another.

Biologically-Inspired Methods

As a result of millions of years of evolution, human and animal visual systems exhibit remarkable robustness to visual disturbances. These systems can handle various environmental and visual challenges effortlessly. For instance, humans easily recognize objects in varying lighting, orientations, and even when partially hidden. This capability contrasts sharply with CNNs, which often falter under similar conditions. This disparity has sparked interest in exploring and emulating the principles of biological vision systems to enhance artificial ones.

Dapello, Marques et al. [51] showed that incorporating a model of the primary visual cortex area V1 in front of CNNs significantly enhances their robustness against white-box adversarial attacks, which are attacks where the attacker has complete knowledge of the model's architecture and parameters. In a similar work, [52] improved resilience to noise by substituting the initial convolutional layer of a standard CNN with Gabor filters. A Gabor filter is a filter used for texture and edge detection in images, characterized by its sinusoidal signal of particular frequency and orientation, modulated by a Gaussian envelope:

$$G(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \psi\right) \tag{1.22}$$

where x and y are the spatial coordinates in the original image, $x' = x\cos\theta + y\sin\theta$ and $y' = -x\sin\theta + y\cos\theta$ are the coordinates in the frame obtained after rotation by the filter's preferred orientation θ, λ denotes the wavelength of the sinusoidal component, θ represents the orientation of the normal to the parallel stripes of a Gabor function, ψ is the phase offset of the sinusoidal function, σ is the standard deviation of the Gaussian envelope, and γ is the spatial aspect ratio of the Gaussian ellipse.

Gabor filters are believed to be approximately implemented in area V1 [53]. Additional efforts in adopting biologically-inspired approaches for improvement in robustness include constraining the CNN models to align their representations more closely with those of mouse V1 [54], and training models to predict neural activity in the V1 area during image classification tasks [55].

VOneNet

In this study, we specifically examine the biologically-inspired CNN known as VOneNet [51]. The VOneNet (Figure 1.5) is a hybrid CNN that incorporates a model of the V1 area of the visual cortex as its front end, followed by a standard trainable CNN architecture. This design was inspired by research showing that the early stages of more robust models closely resemble neuronal responses in the macaque V1.

At the heart of VOneNet is the VOneBlock (Figure 1.5A), a specialized front-end module that encapsulates a fixed-weight neural network mimicking the early visual processing observed in primates. The VOneBlock is based on the linear-nonlinear-Poisson (LNP) model [56]. It includes a series of biologically-constrained Gabor filters (Gabor filter bank) (Figure 1.5A) designed to simulate V1's receptive fields [53]. These filters capture the orientation and spatial frequency information from visual stimuli, analogous to the processing done by simple cells in V1.

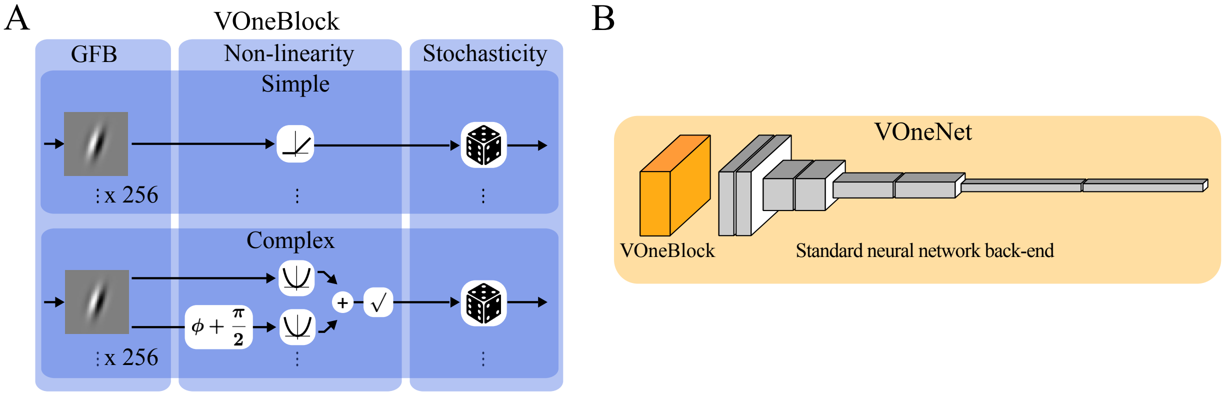


FIGURE 1.5. **VOneNet model architecture.** A. The VOneBlock is a model of V1 with a Gabor filter bank, a non-linear stage, and a stochasticity generator. B. VOneNet is the VOneBlock followed by a standard trainable CNN architecture.

After the Gabor filter bank, the model incorporates layers that represent simple and complex cell [57] nonlinearities (Figure 1.5A). For simple cells, the model applies a rectified linear transformation, ensuring outputs are strictly non-negative. Complex cells combine the responses of pairs of simple cells with identical spatial frequencies and orientations but offset by 90 degrees in phase. This is achieved by calculating the square root of the sum of the squared responses of these phase-offset filters, effectively measuring the spectral power across the quadrature phase pair. This process renders the complex cell's response invariant to the exact phase of the stimulus. These nonlinearities are critical, allowing VOneNet to maintain robustness against minor variations in the image, such as slight shifts or distortions, much like the complex cells in V1 that respond to broader features and patterns rather than precise details.

The nonlinearities are followed by a stochastic layer (Figure 1.5A), which injects neuronal variability akin to the stochastic nature of biological neurons [58]. This variability is not arbitrary but exhibits a distinct pattern; for instance, studies on awake monkeys have shown that the variability in their spike counts across trials tends to follow a Poisson distribution, where the variance of the spike count is approximately equal to its mean. To approximate this property of neuronal responses, independent Gaussian noise is added to each unit of the VOneBlock, with variance equal to its activation. It provides the network with a degree of randomness that is expected to be beneficial in handling a variety of visual perturbations.

The VOneBlock's parameters are not learned during training but are instead mathematically parameterized to closely approximate primate V1 neural processing [59, 60, 61]. The rest of the VOneNet architecture (Figure 1.5B), following the VOneBlock, uses conventional CNN layers that are trainable. By incorporating a V1-inspired module with a standard CNN, VOneNet not only retains high performance on benchmark datasets such as ImageNet but also exhibits increased robustness to adversarial attacks compared to its CNN counterparts.

The integration of a V1 model into CNNs led to substantial improvements in adversarial robustness [51], rivaling more computationally intensive methods like adversarial training. However, despite these advances, the VOneNet models did not exhibit significant gains in robustness against common image corruptions [51], including noise variations like Gaussian, shot, and impulse noise; blurs such as motion, defocus, zoom, and glass blur; weather-induced distortions like snow, frost, fog, and changes in brightness; and digital alterations like JPEG compression, elastic transformations, contrast adjustments, and pixelation. This shortfall highlights a crucial area for further research and development, underscoring the need for models that are robust not only to adversarial attacks but also to a variety of real-world image distortions.

1.3.2. Contribution

In Chapter 5, we demonstrate that by combining the specific strengths of different neuronal circuits in V1 to create diverse VOneNet variants, it is possible to improve the robustness of CNNs for a wide range of image corruptions. Each variant of VOneNet utilizes a unique VOneBlock configuration by either omitting or altering specific components. We explored eight distinct variants: the standard VOneBlock, variants without neural stochasticity, those with sub-Poisson stochasticity, and versions focusing solely on low, intermediate, or high spatial frequency filters, as well as those exclusively utilizing simple cells or complex cells. We first observed that different variants of the V1-inspired front-end result in trade-offs between accuracy and robustness. Motivated by this observation, our goal is to integrate these different variants to combine their individual strengths. To this end, we employ two primary machine learning techniques:

Ensembling This technique involves combining multiple models to create a more powerful composite model. This technique has been shown to enhance performance significantly, leading to superior outcomes compared to individual models [62,63,64,65,66,67]. In our case, we create an ensemble by combining the different V1 front-end variants. We achieve this by using the ensembling technique of uniformly averaging the logits (the outputs of the model before applying the softmax function) of the individual models. This combined model effectively leverages the strengths of each individual model, leading to significant improvements in robustness across all corruption categories and outperforms the base model by 38% on average.

Knowledge Distillation While the ensemble model showed significant improvement in robustness, it was also computationally more expensive. To address this, we utilize the knowledge distillation technique [68], which trains a smaller 'student' model to emulate the performance of a larger 'teacher' model. This technique involves constraining the student predictions, specifically the logits, to match that of the teacher, rather than just predicting binary labels. By fitting the student model to the logits of the teacher, it captures the nuanced probability distribution that the teacher model has learned over the classes. In our case, the 'teacher' was the robust ensemble of the VOneNet variants, and the 'student' was a single model with a V1 front-end. We find that this 'student' model successfully compressed the knowledge of the ensemble model, exhibiting improved robustness to all image corruption categories while maintaining its performance on clean images.

CHAPTER 2

Choice-Selective Sequences Dominate in Cortical Relative to Thalamic Inputs to NAc to Support Reinforcement Learning

This chapter appears as an article [69] published in Cell Reports 2022. This work was done in collaboration with Nathan F. Parker, Julia Cox, Laura M. Haetzel, Anna Zhukovskaya, Malavika Murugan, Ben Engelhard, Mark S. Goldman, and Ilana B. Witten.

2.1. Introduction

Multiple lines of experimental evidence implicate the nucleus accumbens (NAc, part of the ventral striatum) in reward-based learning and decision-making [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. The NAc is a site of convergence of glutamatergic inputs from a variety of regions, including the prefrontal cortex and the midline thalamus, along with dense dopaminergic inputs from the midbrain [15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25].

A central question in reinforcement learning is how actions and outcomes become associated with each other, even when they are separated in time [70, 71, 72, 73]. A possible mechanism that could contribute to solving this problem of temporal credit assignment in the brain is that neural activity in the glutamatergic inputs to the NAc provide a neural memory trace of previous actions. However, whether glutamatergic inputs to the NAc indeed represent memories of previous actions is unclear. More broadly, what information is carried by glutamatergic inputs to the NAc during reinforcement learning, and whether different inputs provide overlapping or distinct streams of information, has not been examined systematically. To date, there have been relatively few recordings of cellular-resolution activity of glutamatergic inputs to the NAc during reinforcement learning, nor comparison of multiple inputs within the same task, nor examination of the timescale with which information is represented within and across trials. Furthermore, if glutamatergic inputs do indeed provide memories of previous actions, construction of a neurally plausible instantiation of an algorithm for credit assignment based on the measured signals remains to be demonstrated (for review of biological instantiation of reinforcement learning algorithms, see [74]).

To address these gaps, we recorded from glutamatergic inputs to the NAc during a probabilistic reversal learning task and built circuit-based computational models to connect our data to prominent theories of reinforcement learning. In this task, dopamine neurons that project to the NAc encode RPE, and inhibition of dopamine neurons substitutes for a negative RPE [26]. To compare activity in major cortical and thalamic inputs to the NAc core, we combined a retrograde viral targeting strategy with cellular-resolution imaging to examine the inputs from prelimbic cortex ("PL-NAc", part of medial prefrontal cortex) and midline regions of the thalamus ("mTH-NAc"). We found that PL-NAc neurons preferentially encode actions and choices relative to mTH-NAc neurons, with choice-selective sequential activity that bridges the delay between choice and reward and that persists until the start of the subsequent trial. We demonstrated with computational modeling that these choice-selective sequences can support neural instantiations of reinforcement learning algorithms, either through dopamine-dependent changes in synaptic weights onto NAc neurons [75, 76, 77, 78], or dopamine-dependent changes in neural dynamics [79]. Finally, we test and confirm a prediction of our models through direct optogenetic manipulation of PL-NAc neurons. Thus, by recording and manipulating glutamatergic inputs to the NAc and integrating these data with computational modeling, we provide specific proposals for how reinforcement learning could be implemented by neural circuitry.

2.2. Results

2.2.1. Cellular resolution imaging of glutamatergic inputs to the NAc during a probabilistic reversal learning task

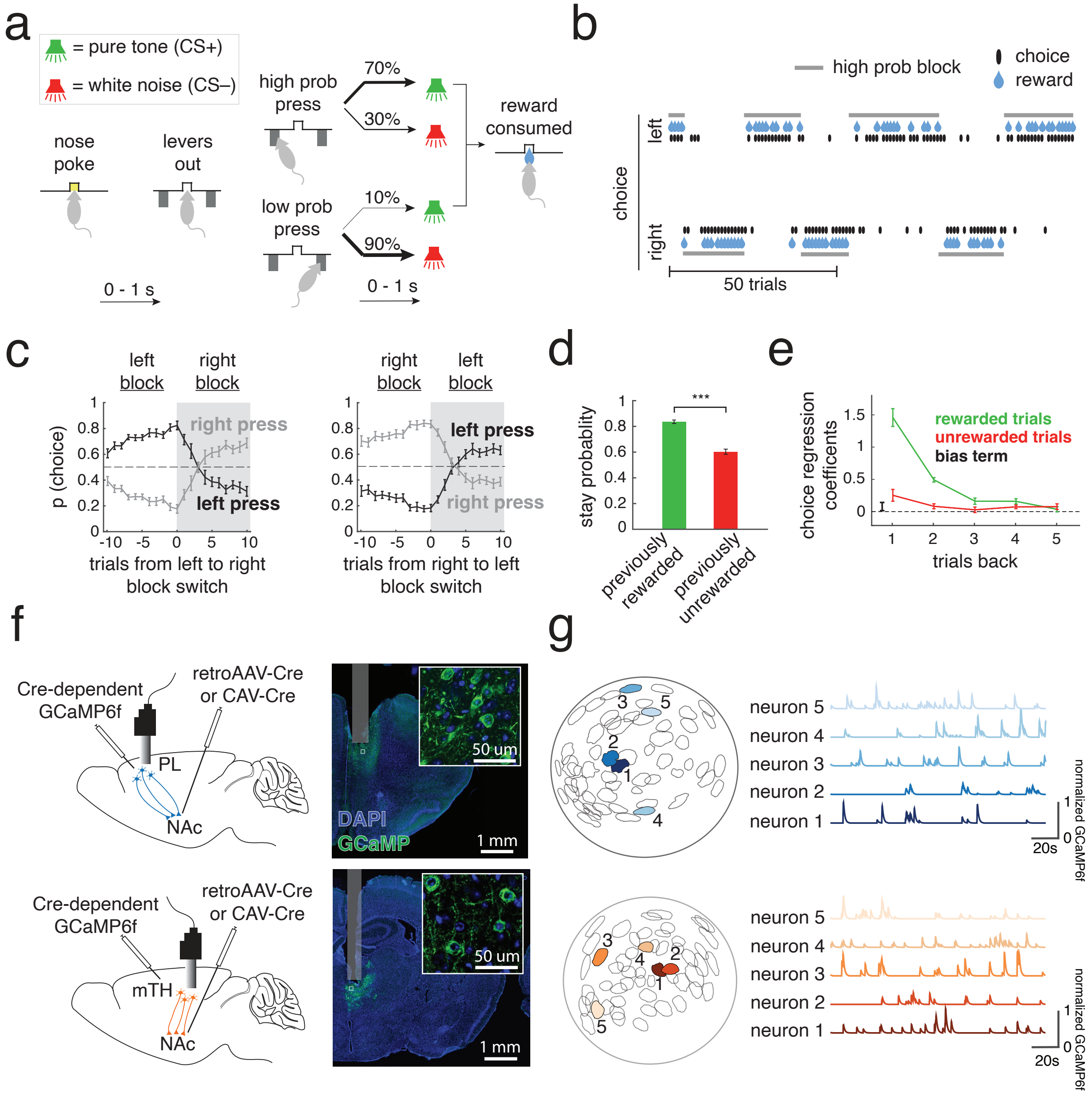


Figure 2.1. **Cellular-resolution imaging of PL and mTH neurons that project to the NAc in mice performing a reinforcement learning task.** (a) Schematic of probabilistic reversal learning task. (b) Example behavior during a recording session. The choice of the mouse (black marks) follows the identity of the high probability lever as it alternates between left and right (grey lines). (c) Left, Probability the mice choose the left or right lever 10 trials before and after a reversal from a left to right high-probability block. Right, same as left for right to left high-probability block reversals. (d) Mice had a significantly

higher stay probability following a rewarded versus unrewarded trial (*** p=5x10-9, two-tailed t-test, n=16 mice). (e) Coefficients from a logistic regression that uses choice and outcome from the previous five trials to predict choice on the current trial. Positive regression coefficients indicate a greater likelihood of repeating the previous choice. Data in c,d,e are represented as mean ± SEM across mice (n=16). (f) Left, surgical schematic for PL-NAc (top) and mTH-NAc (bottom) recordings showing the injection site and optical lens implant with miniature head-mounted microscope attached. Right, Coronal section from a PL-NAc (top) and mTH-NAc (bottom) mouse showing GCaMP6f expression in the recording sites. Inset: confocal image showing GCaMP6f expression in individual neurons. (g) Left, example field of view from a recording in PL-NAc (top, blue) and mTH-NAc (bottom, orange) with five representative regions of interest (ROIs). Right, normalized GCaMP6f fluorescence traces from the five ROIs on the left. For visualization, each trace was normalized by the peak fluorescence across the hour-long session.

Mice performed a probabilistic reversal learning task while inputs from thalamus or cortex were imaged (Figure 2.1a). A trial was initiated when the mouse entered a central nose poke, which prompted the presentation of a lever on either side after a variable delay of 0-1 s. Each lever had either a high (70%) or low (10%) reward probability, with the identity of the high and low probability levers reversing in an unsignaled manner after a variable number of trials (see Methods for block reversal probabilities). After a variable delay (0-1s), either a sound (CS+) was presented at the same time as a reward was delivered to a central reward port, or another sound (CS–) was presented that signaled the absence of reward.

As expected, mice switched the lever they were more likely to press following block reversals (Figure 2.1b,c). Similarly, mice were significantly more likely to return to the previously chosen lever (i.e. stay) following rewarded, as opposed to unrewarded, trials (Figure 2.1d), meaning that, as expected, mice were using previous choices and outcomes to guide behavior. A logistic regression to predict choice based on previous choices and outcomes indicated that mice relied on ∼3 previous trials to guide their choices (Figure 2.1e; see Methods for choice regression details).

To image activity of glutamatergic input neurons to the NAc during this behavior, we injected a retroAAV or CAV2 to express Cre-recombinase in the NAc as well as an AAV2/5 to Cre-dependently express GCaMP6f in either the PL or mTH (Figure 2.1f). A gradient refractive index (GRIN) lens was implanted above either the PL or mTH (see Supplementary Figure 2.8 for implant locations), and a head-mounted miniature microscope was used to image activity in these populations during behavior (Figure 2.1f,g, n=278 neurons in PL-NAc from n=7 mice, n=256 neurons in mTH-NAc from n=9 mice). Behavior between mice in the PL-NAc versus mTH-NAc cohorts was similar (Supplementary Figure 2.9).

2.2.2. Actions are preferentially represented by PL-NAc neurons, while reward-predicting stimuli are preferentially represented by mTH-NAc neurons

Individual PL-NAc and mTH-NAc neurons displayed elevated activity when time-locked to specific behavioral events in the task (Figure 2.2a). Given the correlation between the timing of task events, as well as the temporal proximity of events relative to the time-course of GCaMP6f, we built a linear encoding model to properly relate neural activity to each event [26, 80, 81, 82, 83, 84, 85, 86, 87]. Briefly, time-lagged versions of each behavioral event (nosepoke, lever press, etc) were used to predict the GCaMP6f fluorescence in each neuron using a linear regression. This allowed us to obtain "response kernels", which related each event to the GCaMP6f fluorescence in each neuron, while removing the potentially confounding (linear) contributions of correlated task events (Figure 2.2b; see Methods for details).

To visualize the response kernels, we plotted them as a heatmap, where each row was the response kernel for a particular neuron associated with each behavioral event. This heatmap was then ordered by the time of peak kernel value across all behavioral events. Visual observation revealed a clear difference between the PL-NAc and mTH-NAc populations: PL-NAc neurons were robustly modulated by the action-events in our task (Figure 2.2c; kernel values associated with 'nose poke, 'ipsilateral lever press', 'contralateral lever press' and 'reward consumption') while mTH-NAc neurons appeared to be most strongly modulated by the stimulus-events, specifically the positive reward auditory cue (Figure 2.2d, kernel values associated with 'CS+').

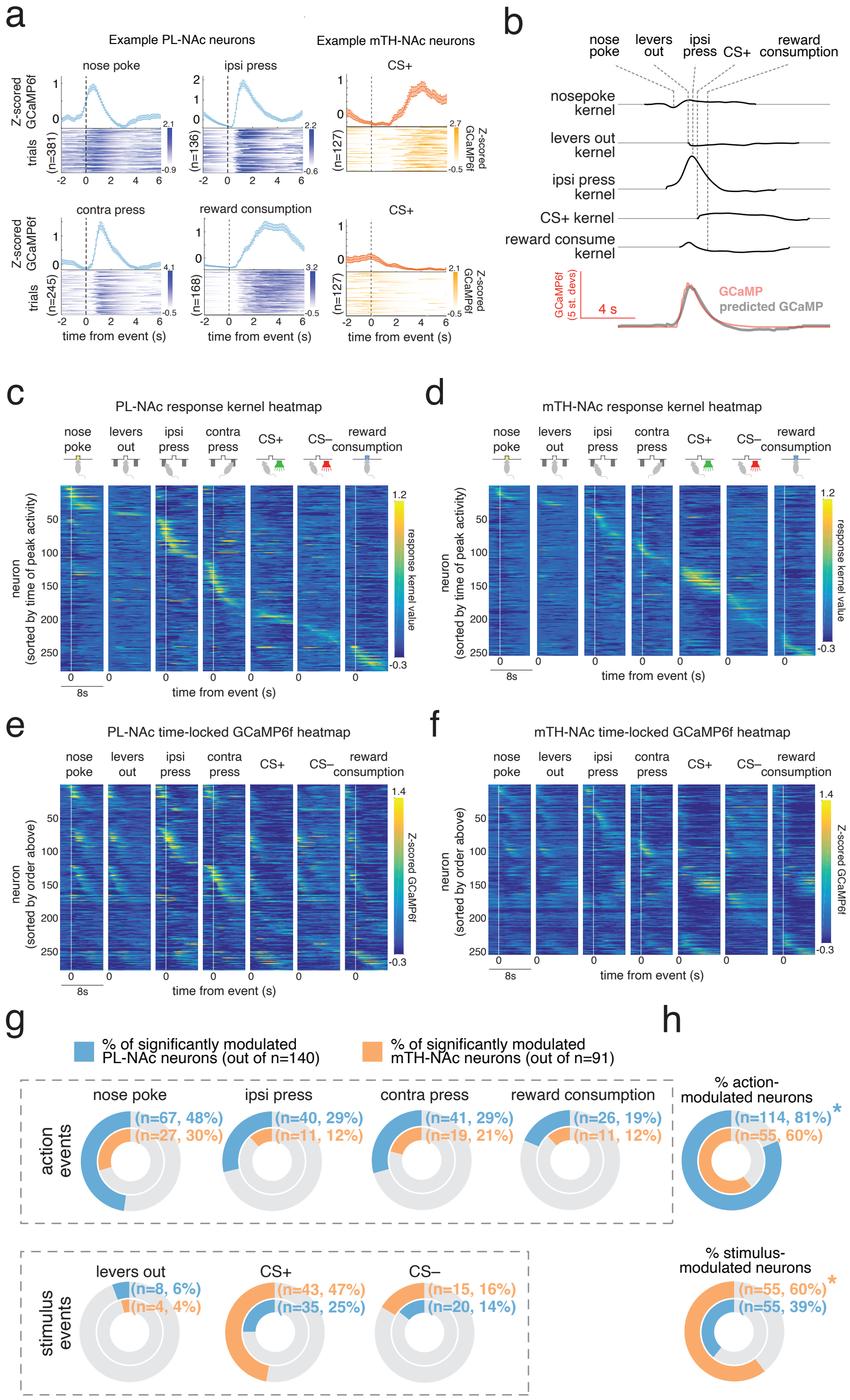


Figure 2.2. **PL-NAc preferentially represents action events while mTH-NAc preferentially represents the CS+.** (a) Time-locked responses of individual PL-NAc (blue) and mTH-NAc (orange)

neurons to task events. Data are represented as mean ± SEM across trials. (b) Kernels representing the response to each of the task events for an example neuron, generated from the encoding model. The predicted GCaMP trace is the sum of the individual response kernels (see Methods). (c) Heatmap of response kernels generated from the encoding model for all PL-NAc neurons. Heatmap is ordered by the time of the peak response across all behavioral events (n=278 neurons, n=7 mice). (d) Same as c except heatmap of response kernels from mTH-NAc neurons (n=256 neurons, n=9 mice). (e) Heatmap of mean Z-scored GCaMP6f fluorescence from PL-NAc neurons aligned to the time of each event in the task. Neurons are ordered as in c. (f) Same as e for mTH-NAc neurons. (g) Top row, fraction of neurons significantly modulated by action events in the PL-NAc (blue) and mTH-NAc (orange). For all action events, PL-NAc had a larger fraction of significantly modulated neurons than mTH-NAc. Bottom row, fraction of neurons in PL-NAc (blue) and mTH-NAc (orange) significantly modulated by stimulus events. 2 out of 3 stimulus events had a larger fraction of significantly modulated neurons in mTH-NAc than in PL-NAc. Significance was determined using the linear model used to generate response kernels in b (Methods). (h) Top, a significantly larger fraction of event-modulated PL-NAc neurons encode at least one action event (P=0.0004: two-proportion Z-test comparing fraction of action-modulated PL-NAc and mTH-NAc neurons). Bottom, a significantly larger fraction of mTH-NAc neurons encode a stimulus event (P=0.002: two-proportion Z-test comparing fraction of stimulus-modulated neurons between PL-NAc and mTH-NAc). For g,h, fractions are determined using the total number of neurons significantly modulated by at least one task event (n=140 for PL-NAc, n=90 for mTH-NAc).

Examination of the GCaMP6f fluorescence time-locked to each behavioral event (rather than the encoding model-derived response kernels) revealed similar observations of action encoding in PL-NAc and CS+ encoding in mTH-NAc (Figure 2.2e,f). While this time-locked GCaMP6f heatmap displays neurons which appear to respond to multiple events (Figure 2.2e, see neurons approximately 70-170 that show elevated activity to 'lever press', 'levers out' and 'nose poke'), this impression is likely a result of the temporal correlation between neighboring behavioral events, which our encoding model accounts for. To illustrate this, we applied our encoding model to a population of simulated neurons that responded only to the lever press events. We observed a similar multi-peak heatmap when simply time-locking the simulated GCaMP6f fluorescence, but this multi-peak effect is eliminated by the use of our encoding model, which recovers the true relationship between GCaMP6f fluorescence and behavior in the simulated data (Supplementary Figure 2.10).

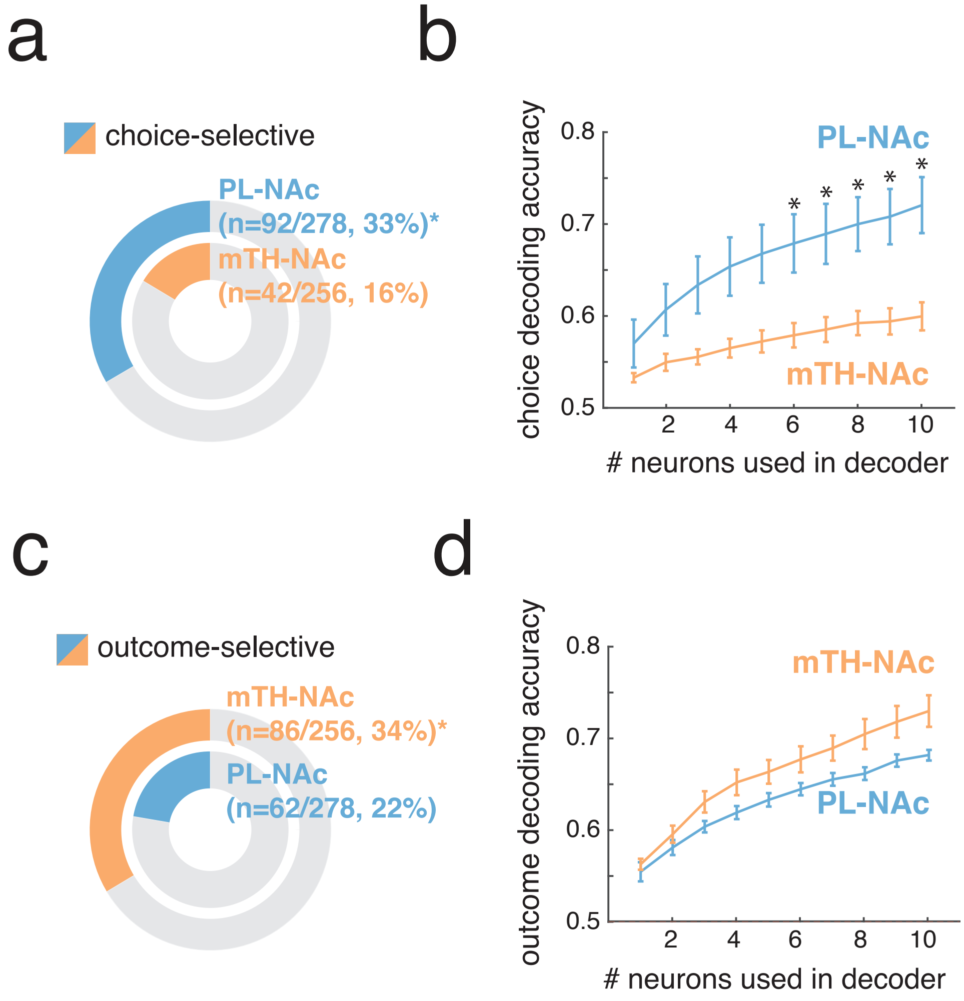


Figure 2.3. **PL-NAc preferentially represents choice but not outcome relative to mTH-NAc.** (a) Fraction of choice-selective neurons in PL-NAc (n=92 out of 278 neurons, 7 mice) and mTH-NAc (n=42 out of 256 neurons, 9 mice). A significantly larger fraction of PL-NAc neurons were choice-selective compared to mTH-NAc neurons (P=9.9x10-6: two-proportion Z-test). (b) Choice decoding accuracy using randomly selected subsets of simultaneously imaged neurons around the lever press. The PL-NAc population more accurately decoded the choice of the trial compared with mTH-NAc (* indicates P<0.05, unpaired two-tailed t-test, n=9 PL-NAc and 6 mTH-NAc mice, peak decoding accuracy of 72±3% for PL-NAc and 60±2% for mTH-NAc). (c) Fraction of outcome-selective neurons in mTH-NAc (n=86 out of 256 neurons, 9 mice)

and PL-NAc (n=62 out of 278 neurons, 7 mice). A significantly larger fraction of mTH-NAc neurons were outcome-selective compared to PL-NAc neurons (P=0.004: two-proportion Z-test). (d) Outcome decoding accuracy using neural activity after the time of the CS from randomly selected, simultaneously imaged neurons in mTH-NAc (orange, peak decoding accuracy: 73±2%) and PL-NAc (blue, peak decoding accuracy: 68±1%). P>0.05, unpaired two-tailed t-test. Data in b,d are represented as mean ± SEM across mice; n=6 PL-NAc mice and 9 mTH-NAc mice. In a,c, * indicates P<0.05, two-proportion Z-test.

This encoding model was used to identify neurons in the PL-NAc and mTH-NAc populations that were significantly modulated by each event in our task (by comparing the encoding model with and without each task event, see Methods). We found that a substantial fraction of both PL-NAc and mTH-NAc neurons were modulated by at least one task event (Figure 2.2g). Of these neurons that were selective to at least one task event, the selectivity for actions versus sensory stimuli differed between the two populations (Figure 2.2h). In particular, more PL-NAc neurons were modulated by at least one action event (nose poke, ipsilateral lever press, contralateral lever press and reward consumption). By contrast, a significantly larger fraction of mTH-NAc neurons were modulated by at least one stimulus cue (levers out, CS+ and CS-).

2.2.3. PL-NAc neurons preferentially encode choice relative to mTH-NAc neurons

This preferential representation of actions in PL-NAc relative to mTH-NAc suggests that lever choice (contralateral versus ipsilateral to the recording site) could also be preferentially encoded in PL-NAc. Indeed, a significantly larger fraction of neurons were choice-selective in PL-NAc compared with mTH-NAc (Figure 2.3a; significant choice-selectivity was determined with a nested comparison of the encoding model with and without choice information, see Methods). A logistic regression population decoder supported this observation of preferential choice-selectivity in PL-NAc relative to mTH-NAc (Figure 2.3b).

In contrast to the preferential representation of choice in PL-NAc compared to mTH-NAc, a larger fraction of neurons in mTH-NAc encoded outcome (CS identity or reward consumption) compared to PL-NAc (Figure 2.3c). However, while outcome decoding accuracy in mTH-NAc was slightly higher relative to PL-NAc, this difference was not statistically significant (Figure 2.3d). These results suggest that, unlike the preferential choice representation observed in PL-NAc over mTH-NAc, outcome was more similarly represented between these two populations. This is presumably due to the fact that both CS+ and reward consumption responses contribute to outcome representation, and although more neurons encoded CS+ in mTH-NAc, the opposite was true for reward consumption (Figure 2.2g). We found no obvious relationship between the strength of either choice or outcome decoding and recording location in either PL-NAc or mTH-NAc (Supplementary Figure 2.11).

2.2.4. PL-NAc neurons display choice-selective sequences that persist into the next trial

We next examined the temporal organization of choice-selective activity in PL-NAc neurons. Across the population, choice-selective PL-NAc neurons displayed sequential activity with respect to the lever press that persisted for >4s after the press (Figure 2.4a-c; see Supplementary Figure 2.12 for sequences without peak-normalization). These sequences were visualized by time-locking the GCaMP6f fluorescence of choice-selective neurons with respect to the lever press, rather than with the encoding model from the earlier figures. The robustness of these sequences was confirmed using a cross-validation procedure, in which the order of peak activity across the PL-NAc choice-selective population was first established using half the trials (Figure 2.4b, 'train'), and then the population heatmap was plotted using the same established ordering and activity from the other half of trials (Figure 2.4c, 'test'). To quantify the consistency of these sequences, we correlated the neurons' time of peak activity in the 'training' and 'test' data and observed a strong correlation (Figure 2.4d). Additionally, the ridge-to-background ratio, a metric used to confirm the presence of sequences [88,89,90] was significantly higher when calculated using the PL-NAc choice-selective sequences compared with sequences generated using shuffled data (Supplementary Figure 2.13a-c).

In contrast, choice-selective sequential activity in the mTH-NAc population was significantly less consistent than in PL-NAc (Supplementary Figure 2.14a-d). Additionally, while the ridge-to-background ratio of the sequences generated using mTH-NAc activity was significantly higher than that using shuffled data, this ratio was also significantly lower than that obtained from PL-NAc sequences (Supplementary Figure 2.13d-f). The ridge-to-background ratio of both the PL-NAc and mTH-NAc sequences did not significantly change across either a block or recording session (Supplementary Figure 2.15a-d).

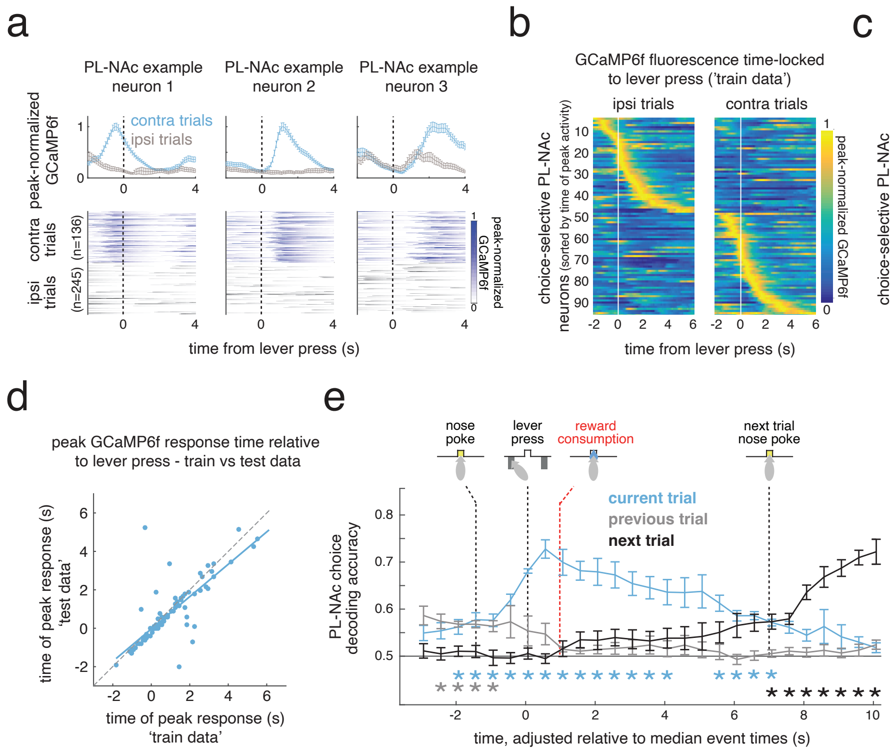


Figure 2.4. **Choice-selective sequences in PL-NAc persist into the subsequent trial.** (a) Top, average peak-normalized GCaMP6f fluorescence of three simultaneously imaged PL-NAc choice-selective neurons. Data are represented as mean ± SEM across trials. Bottom, heatmaps of GCaMP6f fluorescence across trials aligned to ipsilateral (blue) and contralateral (grey) press. (b,c) Heatmaps showing sequential activation of choice-selective PL-NAc neurons (n=92/278 neurons from 7 mice). Each row is a neuron's average GCaMP6f fluorescence time-locked to the ipsilateral (left column) and contralateral (right column) lever press, normalized by its peak average fluorescence. In b ('train data'), heatmap is average fluorescence from half of trials and ordered by the time of peak activity. In c ('test data'), the peak-normalized, time-locked GCaMP6f fluorescence from the other half of trials was plotted in the order from 'train data' in b. (d) Correlation between time of peak activity using the 'train' and 'test' trials for choice-selective PL-NAc neurons in response to a contralateral or ipsilateral lever press ($R2 = 0.80$, $P = 5.3 \times 10^{-22}$, n = 92 neurons).

(e) Average decoding accuracy of choice on the current (blue), previous (grey) and next (black) trial as a function of time-adjusted GCaMP6f fluorescence throughout the current trial from 10 simultaneously imaged PL-NAc neurons. Data are represented as mean ± SEM across mice. Red dashed line indicates median onset of reward consumption. * indicates $P<0.01$, two-tailed, one-sample t-test across mice comparing decoding accuracy to chance, $n = 6$ mice.

A striking feature of these choice-selective sequences in PL-NAc was that they persisted for seconds after the choice, potentially providing a neural 'bridge' between choice and outcome. To further quantify the timescale of choice encoding, both within and across trials, we used activity from simultaneously imaged neurons at each timepoint in the trial to predict the mouse's choice (with a decoder based on a logistic regression using random combinations of 10 simultaneously imaged neurons to predict choice). Choice on the current trial could be decoded above chance for 7s after the lever press, spanning the entire trial (including the time of reward delivery and consumption), as well as the beginning of the next trial (Figure 2.4e). Choice on the previous or subsequent trial was not represented as strongly as current trial choice (Figure 2.4e; in all cases we corrected for cross-trial choice correlations with a weighted decoder, see Methods) and choice from two trials back could not be decoded above chance at any time point (Supplementary Figure 2.15e). We also examined the temporal extent of choice decoding in the mTH-NAc population (Supplementary Figure 2.14e). Similar to PL-NAc, we observed that decoding persisted up to the start of the next trial. However, the peak decoding accuracy across all time points in the trial was lower in mTH-NAc (60±0.1%) compared with PL-NAc (73±0.2%).

2.2.5. Synaptic plasticity or neural dynamics models incorporating choice-selective sequences in PL-NAc neurons can reproduce behavioral and neural recordings

We next used computational modeling to explain how a biologically realistic circuit incorporating the observed choice-selective sequences in PL-NAc neurons could solve the probabilistic reversal task. We constructed two models of the observed trial-by-trial changes in choice probabilities, one based on synaptic plasticity, and one based on slow neural dynamics. Each model sought to explain two features of our data: first, how choices made at an earlier time (around the time of the nose poke, when choice-selective activity appears, Figure 2.4b,c) could be reinforced by rewards that occur at a later time, and, second, how this reinforcement could persist across multiple trials as suggested by our choice regressions (Figure 2.1e).

Synaptic plasticity model. The synaptic plasticity model mathematically implemented a temporal difference (TD) reinforcement learning algorithm by combining the recorded choice-selective sequential activity of PL-NAc neurons with the known connectivity of downstream structures (Figure 2.5a,b). The goal of TD learning is to learn to predict the sum of future rewards, or "value" [**91**, **92**, **93**, **94**]. When this sum of expected future rewards changes, such as when an unexpected reward is received or an unexpected predictor of reward is experienced, a TD reward prediction error (RPE) occurs and adjusts the weights of reward-predicting inputs to reduce this error. The error signal in the TD algorithm closely resembles the RPE signal observed in ventral tegmental area (VTA) dopamine neurons [**26**, **95**, **96**], but how this signal is computed remains an open question.

In our model, the PL-NAc sequences (Figure 2.5c) enabled the calculation of the RPE in dopamine neurons which, in turn, reinforced those PL-NAc inputs that lead to better-than-predicted rewards. In more detail, the model took as inputs experimental, single-trial recordings of choice-selective, sequentially active PL neurons (Figure 2.5a, left, see Methods). These inputs represented temporal basis functions $f_i(t)$ for computing the estimated value of making a left or right choice. These basis functions are weighted in the NAc by the strength w_i of the PL-NAc synaptic connection and summed together to create a (sign-inverted) representation of the estimated value, at time t, of making a left choice, $V_L(t)$, or right choice, $V_R(t)$. To create the RPE observed in DA neurons requires that the DA neuron population receive a fast, positive value signal $V(t)$ and a delayed negative value signal $V(t - \Delta)$, as well as a direct reward signal $r(t)$ (Figure 2.5b). In Figure 2.5a, the summation of NAc inputs and sign-inversion occurs in the ventral pallidum (VP) [**97**, **98**], so that the fast value signal is due to direct VP to DA input. The delayed negative value signal to the DA population is due to a slower, disynaptic pathway that converges first upon the VTA γ-aminobutyric acid (GABA) neurons, so that these neurons encode a value signal as observed experimentally [**99**]. The temporal discounting factor γ is implemented through different strengths of the two pathways to the VTA DA neurons (Figure 2.5b). Other mathematically equivalent circuit architectures, including those involving other structures such as the lateral habenula [100], are given in Supplementary Figure 2.16.

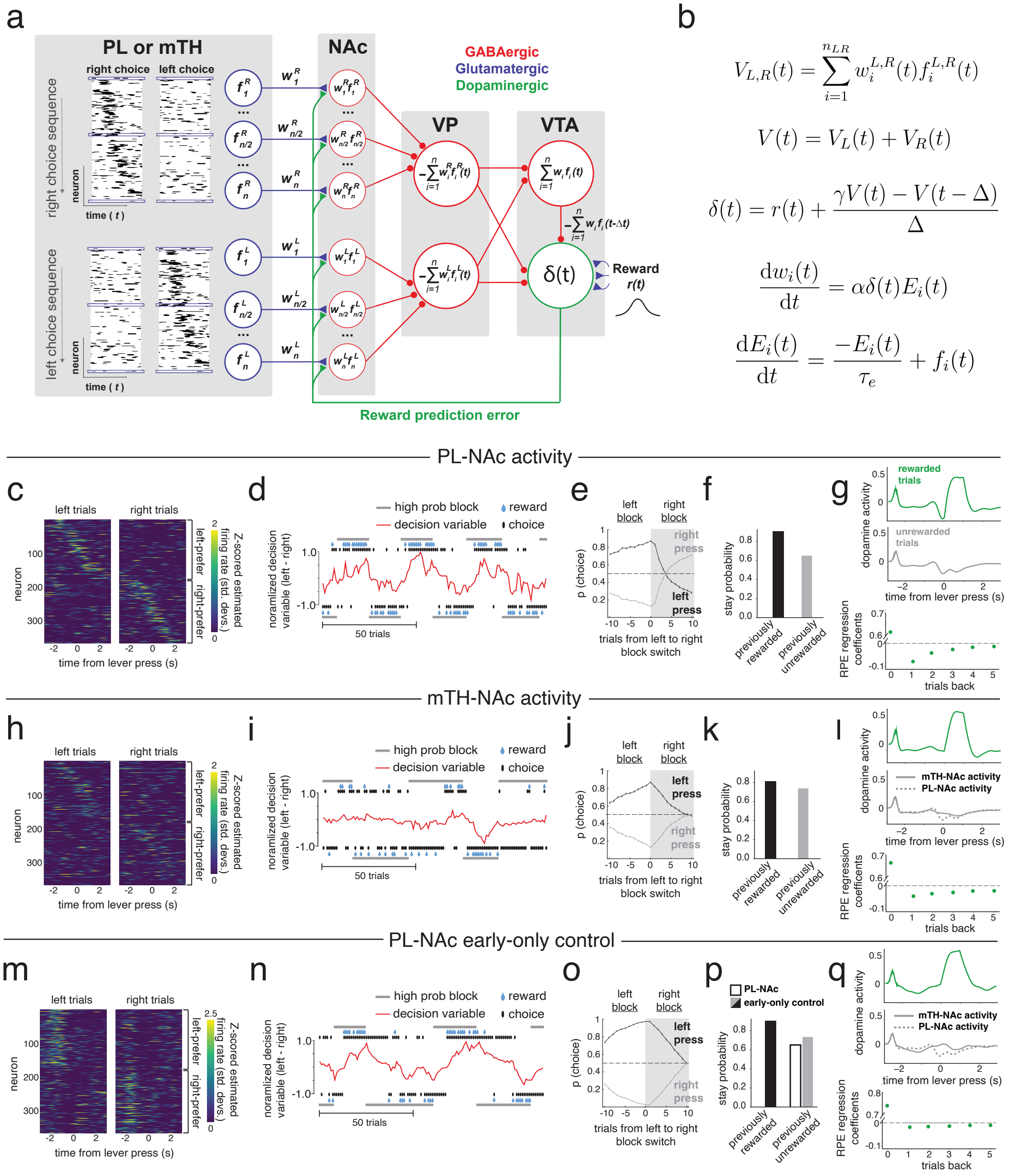


Figure 2.5. **Choice-selective sequences recorded in PL-NAc, combined with known downstream connectivity, can implement a temporal difference (TD) learning model based on synaptic plasticity.** (a) Schematic of circuit architecture used in the model. Model implementation used single-trial recorded PL-NAc or mTH-NAc responses as input. See Results and Methods for model details and Supplementary Figure 2.16 for alternative, mathematically equivalent circuit architectures. (b) Model equations. V: value; V_L, V_R: weighted sum of the n_L left-choice or n_R right-choice preferring NAc neuron activities f_i^L and f_i^R, respectively, with weights w_i^L or w_i^R. α: learning rate; τ_e: decay time constant for the PL-NAc synaptic eligibility trace $E(t)$; Δ: delay of the pathway through the VTA GABA interneuron; γ: discounting of value during time Δ. (c) Heatmap of single trial PL-NAc estimated firing rates input to the model. (d) Behavior of the synaptic plasticity model for 120 example trials. The decision variable (red trace) and the choice of the model (black dots) follow the identity of the higher probability lever. (e) Probability the model chooses left (black) and right (grey) following a left-to-right block reversal. (f) Stay probability of the synaptic plasticity model following rewarded and unrewarded trials. (g) Top, simulated VTA dopamine neuron activity averaged across rewarded (green) and unrewarded (grey) trials. Bottom, coefficients from a linear regression that uses outcome of the current and previous five trials to predict dopamine neuron activity following outcome feedback (Methods). (h-l) Same as c-g except using the estimated firing rates from mTH-NAc single-trial activity. The mTH-NAc model input generates worse performance than using PL-NAc input, with less and slower modulation of the decision variables, and weaker modulation of DA activity by previous trial outcomes. Dashed line in l shows results from PL-NAc model (same data as panels g). (m) Control model including only early-firing neurons active at the onset of the sequence, when the model makes the choice. (n-q) Same as d-g, except results from using the early-only control model. Open bar and dashed line in p,q show results from PL-NAc model (same data as panels f,g).

Learning is achieved through DA-dependent modification of the PL-NAc synaptic strengths. We assume that PL-NAc neuronal activity leads to an exponentially decaying synaptic "eligibility trace" [93, 101]. The correlation of this presynaptically driven eligibility trace with DA input then drives learning (Figure 2.5b). Altogether, this circuit architecture (as well as those shown in Supplementary Figure 2.16) realizes a TD learning algorithm for generating value representations in the NAc, providing a substrate for the selection of proper choice based on previous trial outcomes.

The synaptic plasticity model was able to correctly perform the task and recapitulate the mice's behavior. It achieved a comparable rate of reward (47.2% for the model, 47.6% for the mice) and exhibited similar alternation of choice following block reversals (Figure 2.5d,e; compare to Figure 2.1b,c; choice was based upon a probabilistic readout, at the start of the sequence, of the difference between right and left values plus a stay-with-previous choice bias (Methods) and similarly higher stay probability following rewarded relative to unrewarded trials (Figure 2.5f; compare to Figure 2.1d).

Model neuron responses resembled those previously observed experimentally. The RPE signal within a trial showed characteristic positive response to rewarded outcomes and negative response to unrewarded outcomes (Figure 2.5g; compare to Supplementary Figure 2.17a,b) and had similar dependence upon previous trial outcomes (Figure 2.5g, multiple linear regression similar to [26, 102]; Supplementary Figure 2.17c-d). The VTA GABA interneuron had a sustained value signal, due to the converging input of the transient, sequential value signals from NAc/VP (Supplementary Figure 2.18), replicating the sustained value signal in VTA GABA interneurons observed in monosynaptic inputs to VTA dopamine neurons [99]. Alternatively, the VP neurons shown in Figure 2.5a could project to a second set of VP neurons that functionally take the place of the VTA GABA interneurons (Supplementary Figure 2.16a,c,f), leading to sustained positive value encoding VP neurons as observed in VTA-projecting VP neurons [103].

We next ran the same model using single-trial activity from choice-selective mTH-NAc neurons (Figure 2.5h). In line with the less consistent sequential choice-selective activity in mTH-NAc relative to PL-NAc (Figure 2.4; Supplementary Figure 2.14), the correct value after a block switch was learned much more slowly within the NAc and VTA GABA neurons (Supplementary Figure 2.18c,d), leading to correspondingly slow changes in choice probability (Figure 2.5i,j). As a result, choice probabilities were often out of sync with the current block, leading to overall reward rate near chance levels (38.7% reward rate, chance rate of 40%). Stay probabilities were inappropriately high following unrewarded trials (Figure 2.5k), reflecting reduced formation of an RPE and thus less negative modulation of dopamine signal at the time of expected reward (Figure 2.5l).

The choice-selective sequences in PL-NAc neurons were critical to model performance, as they allowed proper formation of an RPE at the time of reward receipt. This was verified by generating a control model that only included early-firing PL-NAc neurons (neurons active at the onset of the sequence when the model makes its choice) (Figure 2.5m). This "early-only control" model failed to quickly modulate lever value following block reversals (∼10 trials to reverse following a block switch rather than ∼3 trials for the full PL-NAc data; Figure 2.5n-p). The inferior performance of this control model (model reward rate: 43.9%) reflected two factors. First, the early-only control model was unable to generate a well-timed RPE signal due to the absence of significant PL-NAc input activity at the time of reward. As a result, on unrewarded trials, there was almost no negative reward-predictive dip in DA activity at the time of reward omission, unlike for the model with the full PL-NAc input activity (Figure 2.5q). This lack of learning from unrewarded trials is evident in the stay probability plot (Figure 2.5p), which shows less modulation by unrewarded trials when controlling (by adjusting the model's action-selection parameters) for the stay probability following rewarded trials. Second, unlike the sequential model, the RPE in the early-only control model could not propagate backwards across successive trials, so single-trial learning (enabled by the eligibility trace) was the only mechanism available to bridge the gap in time between the firing of the early-firing decision neurons and an RPE occuring at the time of reward.

Neural dynamics model. The synaptic plasticity model described above requires fast, dopamine-mediated synaptic plasticity, on the time scale of a trial, to mediate behavior. Whether plasticity operates in the NAc on this timescale is unclear. We thus developed an alternative model (Figure 2.6a; Methods) in which the across-trial updating of values and corresponding selection of actions is accomplished through the dynamics of a recurrent neural network rather than the dynamics of synaptic plasticity [79, 104, 105, 106, 107]. The initial learning of the neural network's synaptic weights is based on a reinforcement learning algorithm, which models slow initial task acquisition, but during task performance, synaptic weights remain fixed and the DA RPE serves only to alter neural dynamics.

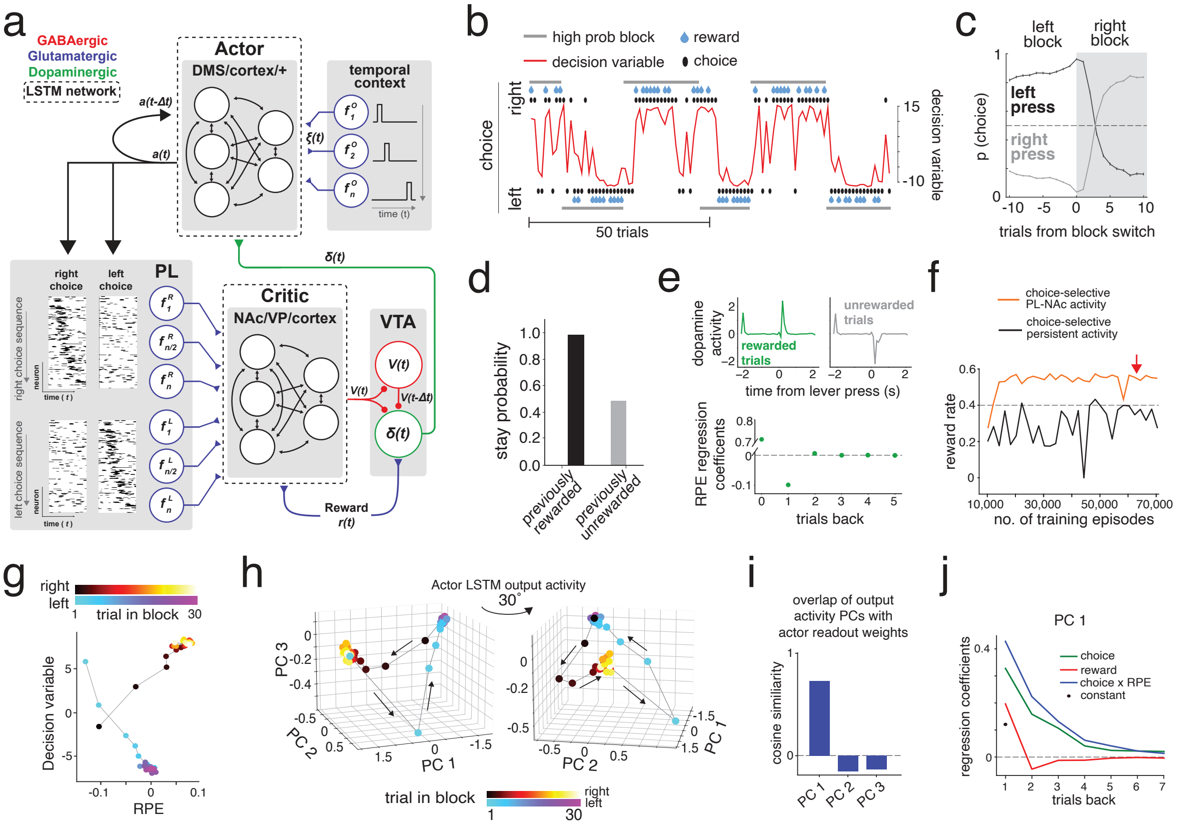


Figure 2.6. **Neural dynamics model, with recorded choice-selective PL-NAc activity input to the critic, performs the task similarly to synaptic plasticity model.** (a) Model schematic. See results and methods for details. (b-e) Example behavior and dopamine activity from the neural dynamics model. Figure panel descriptions same as those for the synaptic plasticity model (Figure 2.5d-g). (f) Reward rate as a function of the number of training episodes for the model with recorded PL-NAc input to the critic (orange) and for a model with persistent choice-selective input to the critic (black). Red arrow indicates the training duration used to generate all other figure panels. Grey dashed line indicates chance reward rate of 0.4. (g) Relationship between the decision variable used to select the choice on the next trial and the calculated RPE across right and left blocks. The RPE shown is an average of 0-2s after lever press, averaged across blocks. The decision variable is also averaged across blocks. (h) Evolution of the principal components of the output of the actor LSTM units across trials within a right and left block. The displayed activity is from the first time point in each trial (when the choice is made), averaged across blocks. The first three components accounted for 70.9%, 16.6%, and 6.4% of the total variance at this time point, respectively. (i)

Cosine of the angle between the actor network's readout weight vector and the vectors corresponding to the first three principal components. Network activity in the PC1 direction (but not PC2 or PC3) aligns with the network readout weights. (j) Coefficients from a linear regression that uses choice on the previous trial (green), average RPE from 0-2 s after the lever press (red), and 'Choice x RPE' interaction (blue) from the previous 7 trials to predict the amplitude of activity in PC1 on the current trial.

Similar to the synaptic plasticity model, single-trial, experimentally recorded PL-NAc activity was input to a (now recurrent) neural network that modeled NAc and other associated brain regions (the "critic network") to calculate value. RPE was calculated in the DA neurons from the value signal using the same circuit architecture as the synaptic plasticity model. However, rather than reweighting PL-NAc synapses on the timescale of trials, the RPE was input to a second recurrent neural network that modeled dorsomedial striatum (DMS) and other associated brain regions (the "actor network"; [108, 109, 110, 111, 112]). This actor network used the RPE input from the previous timestep, the action from the previous timestep and a "temporal context" sequence that may arise from hippocampus or other cortical areas [88, 113, 114] to generate a decision variable corresponding to the probability of selecting one of three choices (left, right, or no action) at any time. Selection of the left or right choice then triggered the onset of the corresponding PL-NAc activity sequence.

The neural dynamics model appropriately modulated choice following a reversal in the identity of the high probability lever (Figure 2.6b-d) and generated RPE signals in VTA dopamine neurons that resemble previous experimental recordings (Figure 2.6e; Supplementary Figure 2.17). By contrast, when we replaced the choice-selective sequences to the NAc by choice-selective persistent activity, the model failed to train within the same number of training episodes (Figure 2.6f). This suggests that temporal structure in this input is beneficial for efficient task learning.

To reveal how the model appropriately modulates its choices, we analyzed the evolution of the actor network's activity across trials (Figure 2.6g-j). We found that the actor network's activity at the time of decision was low-dimensional, with the first three principal components explaining ∼94% of the variance. Given the symmetry in the block structure, the average RPE signal as a function of trial number is similar for the left and right block. However, the model should make opposite choices for left and right blocks, meaning that the actor network needs to respond oppositely to similar RPE inputs. Consistent with this, the decision variable for a given RPE was approximately opposite for left versus right blocks (Figure 2.6g). At a block reversal, for example from a left block to a right block, the network activity rapidly transitioned from the approximately steady-state representation of the left block (cluster of blue-purple points in Figure 2.6h) to the approximately steady-state representation of the right block (cluster of red-yellow points). Furthermore, the model learned to align the first principal component of activity along the direction of the network readout weights that determine the actor's choice $a(t)$ (Figure 2.6i). Thus, the actor learned to generate an explicit representation of the decision variable in the first principal component of its activity.

To solve the reversal learning task, the network needs to use its past history of choices and rewards to accumulate evidence for whether the current block is a left block or a right block. Rewarded left-side choices, or unrewarded right-side choices, represent evidence that the current block is a left block, while the converse represents evidence for a right block. In the synaptic plasticity model (Figure 2.5), new evidence (not accounted for by previous expectations) is accumulated in the PL-NAc synaptic weights as the product of the eligibility trace (which, due to the choice-selectivity of the PL-NAc activity, represents the current choice) and the RPE. To analyze whether the actor network uses a similar accumulation of evidence to solve the task, we linearly regressed the first principal component of actor activity (PC1, which correlated strongly with the decision variable as described above) against the past history of choices and RPEs, which serve as inputs to the network, as well as the product of these ('Choice x RPE'). PC1 most strongly depended upon the 'Choice x RPE' predictor, with coefficients that decayed on a timescale of approximately 3 trials, suggesting that the actor used a leaky accumulation of evidence over this timescale to solve the task (Figure 2.6j, blue trace). In addition, like the mice and the synaptic plasticity model, the neural dynamics model tended to stay with its previous choices as evident from the positive coefficients for the previous choice regressors in Figure 2.6j (green trace). Thus, both the synaptic plasticity model and the neural dynamics model follow the same principle of accumulating evidence across trials to perform fast reversal learning in addition to having a tendency to repeat their previous choices.

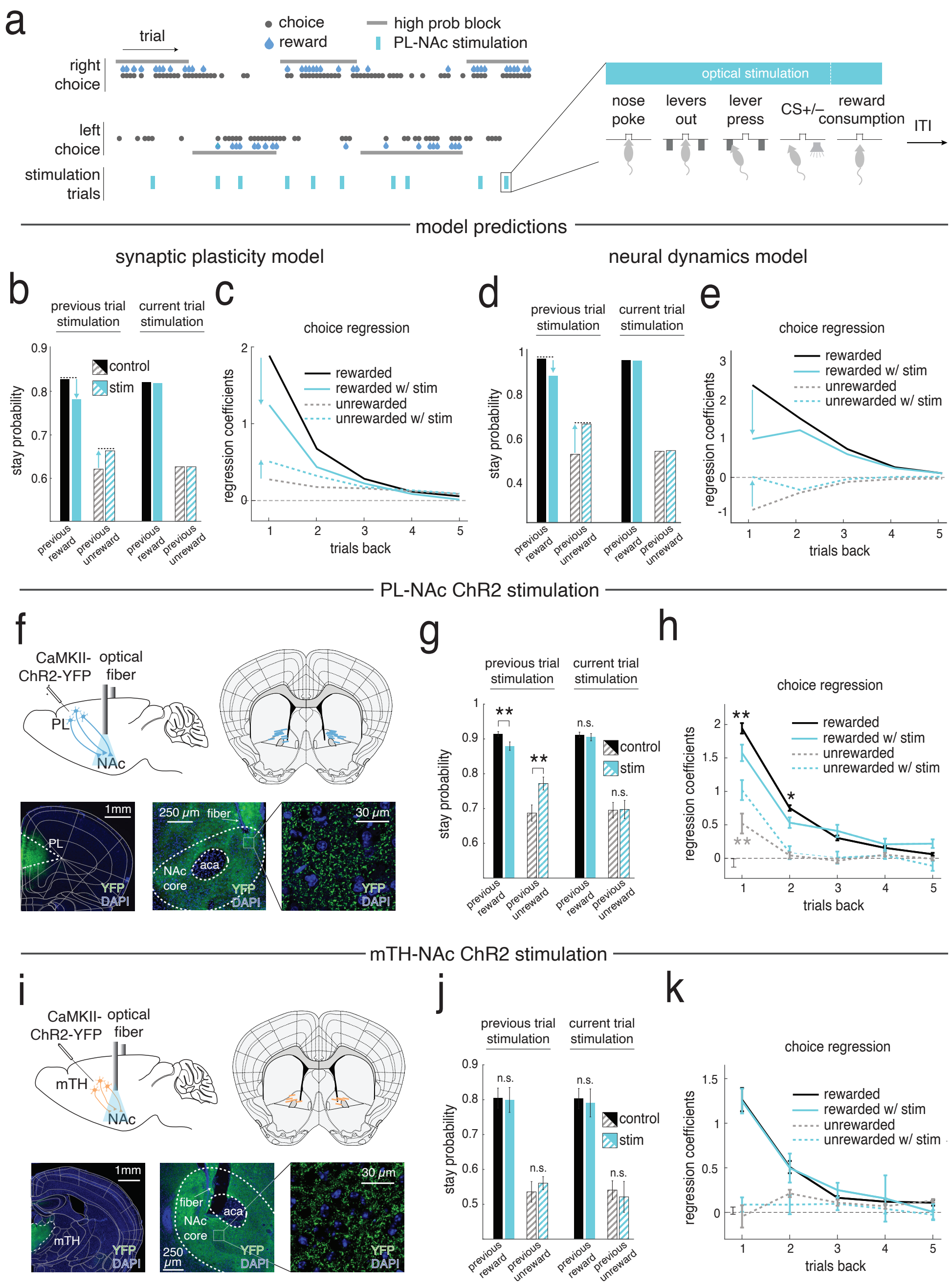


Figure 2.7. **Stimulation of PL-NAc neurons disrupts the influence of previous trial outcomes on subsequent choice in both the models and mice.** (a) In the mice and models, PL-NAc neurons were stimulated for the whole trial on a random 10% of trials, disrupting the endogenous choice-selective

sequential activity (see Methods and Supplementary Figure 2.20). (b) Effect of stimulating the PL-NAc input on the previous (left) or current (right) trial in the synaptic plasticity model. (c) Logistic choice regression showing dependence of the current choice on previously rewarded and unrewarded choices, with and without stimulation. Higher coefficients indicate a higher probability of staying with the previously chosen lever. (d-e) Same as b-c for the neural dynamics model. (f) Top left, schematic illustrating injection site in the PL (black needle) and optical fiber implant in the NAc core. Top right, location of optical fiber tips of PL-NAc ChR2 cohort (n=14 mice) Bottom left, coronal section showing ChR2-YFP expression in PL. Bottom middle and right, ChR2-YFP expression in PL terminals in the NAc-core. (g) Similar to the models, PL-NAc ChR2 stimulation on the previous trial significantly reduced the mice's stay probability following a rewarded trial (P = 0.002) while increasing stay probability following an unrewarded trial (P = 0.0005). Stimulation on the current trial had no significant effect on stay probability following rewarded (P = 0.62) or unrewarded (P=0.91) trials. All comparisons: paired, two-tailed t-tests, n=14 mice. (h) PL-NAc ChR2 stimulation decreased the weight of rewarded choices one- and two-trials back (P=0.002: one-trial back; P=0.023: two-trials back) and increased the weight of unrewarded choices one-trial back ($P=5.4\times10^{-6}$). (i-k) Same as f-h for mTH-NAc ChR2 stimulation (n=8 mice). mTH-NAc stimulation had no significant effect on stay probability following either rewarded (P=0.85) or unrewarded choices (P=0.40) on the previous (j, paired t-test, n = 8 mice) or multiple trials back (k, P>0.05 for all trials back, one-sample t-tests). Current trial stimulation also had no effect following either rewarded (P=0.59) or unrewarded (P=0.50) choices. For all panels, ** indicates P<0.005 and * indicates P<0.05 for one-sample, two-tailed t-tests.

2.2.6. Stimulation of PL-NAc (but not mTH-NAc) neurons decreases the effect of previous trial outcomes on subsequent choice in both the models and the mice

We next generated experimentally testable predictions from our models by examining the effect of disruption of the PL-NAc inputs on behavioral performance. To do so, we simulated optogenetic-like neural stimulation of this projection by replacing the PL-NAc sequential activity in the model with constant, choice-independent activity across 70% of the population on a subset of trials Figure 2.7a). For both models, this generated a decrease in the probability of staying with the previously chosen lever following rewarded trials and an increase following unrewarded trials relative to unstimulated trials (Figure 2.7b,d). In other words, the effect of previous outcome on choice was reduced when PL-NAc activity was disrupted. This effect persists for multiple trials, as revealed by a logistic regression of current-trial choice on the history of previous rewarded and unrewarded choices with and without stimulation (Figure 2.7c,e; note that the negative coefficients for unrewarded trials in the neural dynamics model reflect that, unlike the synaptic plasticity model, this model does not include an explicit stay-with-previous choice bias). This reduced effect of outcome on choice arises because the stimulation disrupts the calculation of value. In the synaptic plasticity model, the stimulation of both left- and right-preferring PL-NAc neurons has two effects: first, it disrupts the RPE calculation by the circuit; second, it leads to dopamine indiscriminately adjusting the synaptic weights (i.e., value) of both the right and left PL-NAc synapses following rewarded or unrewarded outcomes. These weight changes then persist for multiple trials, leading to decreased performance in subsequent trials. In the neural dynamics model, stimulation reduces behavioral performance on subsequent trials by disrupting the RPE signal that is transmitted to the actor, and this effect lasts for multiple trials because the actor network temporally accumulates RPE signals across multiple trials (Figure 2.6j). In both models, the choice behavior on the current trial is unaffected because choice is determined at the beginning of the trial, before the weights are updated (Figure 2.7b,d).

We tested these model predictions experimentally by performing an analogous optogenetic manipulation in mice (Figure 2.7f). In close agreement with our models, mice significantly decreased their stay probability following a rewarded trial that was paired with stimulation and significantly increased their stay probability following an unrewarded trial paired with stimulation (Figure 2.7g). Similar to the models, the effect of stimulation on the mouse's choice persisted for multiple trials. Mice had a significant decrease in their stay probability following PL-NAc stimulation on rewarded choices one and two trials back (Figure 2.7h). Also similar to the model, stimulation on the current trial had no significant effect on choice following either rewarded or unrewarded trials (Figure 2.7g).

In contrast to PL-NAc stimulation, but consistent with the relatively weak choice encoding in mTH-NAc compared to PL-NAc (Figure 2.3a,b) and weak trial-by-trial learning in our synaptic plasticity model (Figure 2.5h-k), mTH-NAc stimulation (Figure 2.7i) had no significant effect on the mice's stay probability on the subsequent trial, following either rewarded or unrewarded stimulation trials (Figure 2.7j). Similarly, inclusion of mTH-NAc stimulation in our choice regression model revealed no significant effect of stimulation on rewarded or unrewarded choices (Figure 2.7k). Additionally, there was no effect on the mice's stay probability for current trial stimulation (Figure 2.7j).

For both PL-NAc and mTH-NAc stimulation, we observed an increase in the probability of mice abandoning the trials with stimulation compared to those trials without (P=0.0006 for PL-NAc, P=0.032 for mTH-NAc: paired, two-tailed t-test comparing percentage of abandoned trials on stimulated versus non-stimulated trials; 12.2±2.5% and 22.1±7.9% abandoned for PL-NAc and mTH-NAc stimulated trials, respectively; 0.9±0.2% and 6.4±3.1% for PL-NAc and mTH-NAc non-stimulated trials, respectively). Relatedly, we also found an increase in the latency to initiate a trial following either PL-NAc or mTH-NAc stimulation (Supplementary Figure 2.19a-c). Together, these results suggest that this manipulation had some influence on the mouse's motivation to perform the task. However, unlike the stronger effect of PL-NAc versus mTH-NAc stimulation on subsequent choice behavior, this trial-abandonment effect was stronger for mTH-NAc than PL-NAc.

To control for non-specific effects of optogenetic stimulation, we ran a control cohort of mice that received identical stimulation but did not express the opsin (Supplementary Figure 2.19e,f). Stimulation had no significant effect on the mice's choice behavior (Supplementary Figure 2.19d,g,h) or probability of abandoning trials on stimulation versus control trials (P=0.38: paired, two-tailed t-test comparing percentage of abandoned trials on stimulated versus non-stimulated trials; 0.4±0.08% for stimulated trials, 0.4±0.01% for non-stimulated trials).

2.3. Discussion

This work provides both experimental and computational insights into how the NAc and associated regions could contribute to reinforcement learning. Experimentally, we found that mTH-NAc neurons were preferentially modulated by a reward-predictive cue, while PL-NAc neurons more strongly encoded actions (e.g. nose poke, lever press). In addition, PL-NAc neurons display choice-selective sequential activity which persists for several seconds after the lever press action, beyond the time the mice receive reward feedback. Computationally, we demonstrate that the choice-selective and sequential nature of PL-NAc activity can contribute to performance of a choice task by implementing a circuit-based version of reinforcement learning based on either synaptic plasticity or neural dynamics. Furthermore, PL-NAc perturbations affect future but not current choice in both the models and in mice, consistent with perturbation of the critic not the actor.

Relationship to previous neural recordings in the NAc and associated regions

To our knowledge, a direct comparison, at cellular resolution, of activity across multiple glutamatergic inputs to the NAc has not previously been conducted. The preferential representations of actions relative to sensory stimuli in PL-NAc is somewhat surprising, given that previous studies have focused on sensory representations in this projection [19], and also given that the NAc is heavily implicated in Pavlovian conditioning [5,7,11,115,116,117].

On the other hand, there is extensive previous evidence of action correlates in PFC [118,119,120,121,122], and NAc is implicated in operant conditioning in addition to Pavlovian conditioning [108,123,124,125,126,127,128]. Our finding of sustained choice-encoding in PL-NAc neurons is in agreement with previous work recording from medial prefrontal cortex (mPFC) neurons during a different reinforcement learning task [129,130]. Additionally, other papers have reported choice-selective sequences in other regions of cortex, as well as in the hippocampus [89,131,132]. In fact, given previous reports of choice-selective (or outcome-selective) sequences in multiple brain regions and species [133,134,135,136,137,138], the relative absence of sequences in mTH-NAc neurons may be more surprising than the presence in PL-NAc.

Our observation of prolonged representation of the CS+ in mTH-NAc (Figure 2.2d,f) is in line with previous observations of pronounced and prolonged encoding of task-related stimuli in the primate thalamus during a Pavlovian conditioning task [139]. Together with our data, this suggests that the thalamus is contributing information about task-relevant stimuli to the striatum, which could bridge the gap between a CS and US in a Pavlovian trace conditioning task [16,140,141,142].

Implementation of reinforcement learning in models based on synaptic plasticity or neural dynamics

We presented two different classes of models that could solve the reversal learning task when provided with the choice-selective sequences observed in PL-NAc neurons as inputs. In our synaptic plasticity model, we show how these sequences may contribute to a neural implementation of TD learning by providing a temporal basis set that bridges the gap in time between actions and outcomes and enables the calculation of RPE in dopamine neurons. Other forms of neural dynamics, such as constant or slowly decaying persistent activity, can also maintain values across a delay period. However, creating a temporally precise RPE from such persistent activity is challenging if the persistent activity does not have sharp temporal features. Likewise, synaptic eligibility traces are another useful mechanism for bridging gaps in time, enabling earlier inputs to be reinforced by an RPE, but they do not provide the active input required to create the RPE itself.

A limitation of the synaptic plasticity model for producing the rapid reversals of behavior at block switches is that it requires a dopamine-dependent synaptic plasticity mechanism that operates on the timescale of trials (Figure 2.5). Whether dopamine-mediated synaptic plasticity operates on such fast timescales is not clear. Furthermore, model-free TD learning cannot take advantage of additional task-structure information such as the reward probabilities within a block [143, 144] but see Supplementary Figure 2.21 for challenges in identifying this ability within tasks like ours). These observations motivated the neural dynamics model in which, following initial slow-timescale learning of synaptic weights, the plasticity was turned off and trial-by-trial modulation of behavior was mediated by dopamine-dependent neural dynamics instead of synaptic plasticity (Figure 2.6; see related work by [79, 104, 105, 106, 107, 145, 146, 147, 148, 149]. Because the recurrent "critic" network dynamics can be trained to construct a temporally rich representation, the neural dynamics model has less need for precise temporal sequences in the PL-NAc inputs. However, we found that strictly eliminating the temporal structure of the PL-NAc input while preserving the choice-selectivity made training of the network less efficient (Figure 2.6f), suggesting that having temporal structure in PL-NAc inputs facilitates the calculation of value.

Previous work in biological TD learning has used sequentially active neurons as the basis for learning in the context of sequential behaviors [150, 151] and learning the timing of a CS-US relationship [152, 153, 154, 155]. Likewise, our neural dynamics model was inspired by a previous meta-reinforcement learning model that was used to solve a reversal learning task [79]. Here we extend these ideas in multiple important ways:

First, we link these theoretical ideas directly to data, by demonstrating that choice-selective sequential activity in the NAc is provided primarily by PL-NAc (as opposed to mTH-NAc) input neurons, and that perturbation of the PL-NAc (but not mTH-NAc) projection disrupts action-outcome pairing consistent with model predictions. Thus, our models provide a mechanistic explanation of a puzzling experimental finding: that optogenetic manipulation of PL-NAc neurons affects subsequent choices but not the choice on the stimulation trial itself and that this stimulation creates oppositely directed effects following rewarded versus unrewarded trials.

Second, both of our models replicate numerous experimental findings in the circuitry downstream of PL-NAc. Each calculates an RPE signal in dopamine neurons [26, 102], generates conjunctive encoding of actions and outcomes [127, 156] and calculates chosen value signals [109]. Additionally, both models generate encoding of value by GABA interneurons [99, 103], which produces the temporally delayed, sign inverted signals required for the calculation of a temporally differenced RPE (Figure 2.5a; see [74, 152, 153, 157, 158, 159, 160, 161, 162]). Consistent with our models, electrical stimulation of VP generates both immediate inhibition of dopamine neurons, and delayed excitation [163]. Conceptually, the proposed temporal differencing by the VTA GABA interneuron is attractive in that it could provide a generalizable mechanism for calculating RPE: it could extend to any pathway that projects both to the dopamine and GABA neurons in the VTA [164], and that also receives a dopaminergic input that can modify synaptic weights.

Third, we showed that the fundamental operating principle of both models was similar: each temporally accumulates the correlation of previous choices with reward to determine the current-trial choice probability. In the synaptic plasticity model, this accumulation is done in the PL-NAc synaptic weights (Figure 2.5b). In the neural dynamics model, the accumulation is done in the low-dimensional neural dynamics of the actor network (Figure 2.6j). Future experiments that exploit these differences will need to be designed and executed to determine whether the brain more closely resembles the synaptic plasticity or neural dynamics model.

Limitations of the Study

A limitation of this study is that we could not artificially recapitulate sequential firing to directly test its role in constructing value representations. Additionally, any artificial stimulation can have off target and unintended consequences. Thus, further work directly investigating the causal role of PL-NAc sequences in reinforcement learning is needed. Neither of our models account for the influence of glutamatergic inputs to NAc from regions other than prelimbic cortex and the medial thalamus. In addition, our neural dynamics model used LSTM units, which should not be interpreted as single neurons, but might model computations performed by larger populations. Finally, single-photon imaging limits the ability to resolve single z-planes during imaging and, thus, can make single neuron identification difficult. Future studies confirming our studies with multi-photon imaging in head-restrained animals may be helpful.

Author contributions

NFP, JC, LMH, AZ, and MM performed the experiments under the supervision of IBW; NFP, AB, JC and BE analyzed the behavioral and neural data; AB performed the modeling work under the supervision of MSG; NFP, AB, JC, MSG and IBW wrote the paper.

Acknowledgements

This work was supported by grants from NIH R01 DA047869 (IBW), U19 NS104648-01 (MSG, IBW), F32 MH112320-02 (JC), ARO W911NF1710554 (IBW), Brain Research Foundation (IBW), Simons Collaboration on the Global Brain (MSG, IBW), and the New York Stem Cell Foundation (IBW). IBW is a NYSCF—Robertson Investigator.

2.4. Methods

2.4.1. Experimental model and subject details

Mice

46 male C57BL/6J mice from The Jackson Laboratory (strain 000664) were used for these experiments. Prior to surgery, mice were group-housed with 3-5 mice/cage. All mice were >6 weeks of age prior to surgery and/or behavioral training. To prevent mice from damaging the implant of cagemates, all mice used in imaging experiments were single housed post-surgery. All mice were kept on a 12-h on/ 12-h off light schedule. All experiments and surgeries were performed during the light off time. All experimental procedures and animal care was performed in accordance with the guidelines set forth by the National Institutes of Health and were approved by the Princeton University Institutional Animal Care and Use Committee.

Probabilistic reversal learning task

Beginning three days prior to the first day of training, mice were placed on water restriction and given per diem water to maintain >80% original body weight throughout training. Mice performed the task in a 21 x 18 cm operant behavior box (MED associates, ENV-307W). A shaping protocol of three stages was used to enable training and discourage a bias from forming to the right or left lever. In all stages of training, the start of a trial was indicated by illumination of a central nose poke port. After completing a nose poke, the mouse was presented with both the right and left lever after a temporal delay drawn from a random distribution from 0 to 1s in 100ms intervals. The probability of reward of these two levers varied based on the stage of training (see below for details). After the mouse successfully pressed one of the two levers, both retracted and, after a temporal delay drawn from the same uniform distribution, the mice were presented with one of two auditory cues for 500ms indicating whether the mouse was rewarded (CS+, 5 kHz pure tone) or not rewarded (CS–, white noise). Concurrent with the CS+ presentation, the mouse was presented with 6μl of 10% sucrose reward in a dish located equidistantly between the two levers, just interior to the central nose poke. The start time of reward consumption was defined as the moment the mouse first made contact with the central reward port spout following the delivery of the reward. The end of the reward consumption period (i.e., reward exit) was defined as the first moment at which the mouse was disengaged from the reward port for >100ms. In all stages of training, trials were separated by a 2s intertrial interval, which began either at the end of CS on unrewarded trials or at the end of reward consumption on rewarded trials.

In the first stage of training ("100-100 debias"), during a two-hour session, mice could make a central nose poke and be presented with both the right and left levers, each with a 100% probability of reward. However, to ensure that mice did not form a bias during this stage, after five successive presses of either lever the mouse was required to press the opposite lever to receive a reward. In this case, a single successful switch to the opposite lever returned both levers to a rewarded state. Once a mouse received >100 rewards in a single session they were moved to the second stage ("100-0") where only one of the two levers would result in a reward. The identity of the rewarded lever reversed after 10 rewarded trials plus a random number of trials drawn from the geometric distribution:

$$P(k) = (1-p)^{k-1} \tag{2.1}$$

where $P(k)$ is the probability of a block reversal trials into a block and is the success probability of a reversal for each trial, which in our case was 0.4. After 3 successive days of receiving >100 total rewards, the mice were moved to the final stage of training ("70-10"), during which on any given trial pressing one lever had a 70% probability of leading to reward (high-probability lever) while pressing the opposite lever had only a 10% reward probability (low-probability lever). The identity of the higher probability lever reversed using the same geometric distribution as the 100-0 training stage. On average, there were 23.23 ± 7.93 trials per block and 9.67 ± 3.66 blocks per session (mean ± std. dev.). In this final stage, the mice were required to press either lever within 10s of their presentation; otherwise, the trial was considered an 'abandoned trial' and the levers retracted. All experimental data shown was collected while mice performed this final "70-10" stage.

Cellular-resolution calcium imaging

To selectively image from neurons which project to the NAc, we utilized a combinatorial virus strategy to image cortical and thalamic neurons which send projections to the NAc. 16 mice (7 PL-NAc, 9 mTH-NAc) previously trained on the probabilistic reversal learning task were unilaterally injected with 500nl of a retrogradely transporting virus to express Cre-recombinase (CAV2-cre, IGMM vector core, France, injected at ∼2.5 × 1012 parts/ml or retroAAV-EF1a-Cre-WPRE-hGHpA, PNI vector core, injected at ∼6.0 x 1013) in either the right or left NAc core (1.2 mm A/P, ± 1.0 mm M/L, -4.7 D/V) along with 600nl of a virus to express GCaMP6f in a Cre-dependent manner (AAV2/5-CAG-Flex -GCaMP6f-WPRE-SV40, UPenn vector core, injected at 1.27 x 1013 parts/ml) in either the mTH (-0.3 & -0.8 A/P, ± 0.4 M/L, -3.7 D/V) or PL (1.5 & 2.0 A/P, ± 0.4 M/L, -2.5 D/V) of the same hemisphere. 154 of 278 (55%, n=5 mice) PL-NAc neurons and 95 out of 256 (37%, n=5 mice) mTH-NAc neurons were labeled using the CAV2-Cre virus, the remainder were labeled using the retroAAV-Cre virus. In this same surgery, mice were implanted with a 500 μm diameter gradient refractive index (GRIN) lens (GLP-0561, Inscopix) in the same region as the GCaMP6f injection – either the PL (1.7 A/P, ± 0.4 M/L, -2.35 D/V) or mTH (-0.5 A/P, ± 0.3 M/L, -3.6 D/V). 2-3 weeks after this initial surgery, mice were implanted with a base plate attached to a miniature, head-mountable, one-photon microscope (nVISTA HD v2, Inscopix) above the top of the implanted lens at a distance which focused the field of view. All coordinates are relative to bregma using *Paxinos and Franklin's the Mouse Brain in Stereotaxic Coordinates, 2nd edition* (Paxinos and Franklin, 2004). GRIN lens location was imaged using the Nanozoomer S60 Digital Slide Scanner (Hamamatsu) (location of implants shown in Supplementary Figure 2.8). The subsequent image of the coronal section determined to be the center of the lens implant was then aligned to the Allen Brain Atlas (Allen Institute, brain-map.org) using the *Wholebrain* software package (wholebrainsoftware.org; [**165**]).

Post-surgery, mice with visible calcium transients were then retrained on the task while habituating to carrying a dummy microscope attached to the implanted baseplate. After the mice acclimated to the dummy microscope, they performed the task while images of the recording field of view were acquired at 10 Hz using the Mosaic acquisition software (Inscopix). To synchronize imaging data with behavioral events, pulses from the microscope and behavioral acquisition software were recorded using either a data acquisition card (USB-201, Measurement computing) or, when LED tracking (see below for details) was performed, an RZ5D BioAmp processor from Tucker-Davis Technologies. Acquired videos were then pre-processed using the Mosaic software and spatially downsampled by a factor of 4. Subsequent down-sampled videos then went through two rounds of motion-correction. First, rigid motion in the video was corrected using the translational motion correction algorithm based on [166] included in the Mosaic software (Inscopix, motion correction parameters: translation only, reference image: the mean image, speed/accuracy balance: 0.1, subtract spatial mean [r = 20 pixels], invert, and apply spatial mean [r = 5 pixels]). The video then went through multiple rounds of non-rigid motion correction using the NormCore motion correction algorithm [167] NormCore parameters: gSig=7, gSiz=17, grid size and grid overlap ranged from 12-36 and 8-16 pixels, respectively, based on the individual motion of each video. Videos underwent multiple (no greater than 3) iterations of NormCore until non-rigid motion was no longer visible). Following motion correction, the CNMFe algorithm [168] was used to extract the fluorescence traces (referred to as 'GCaMP6f' throughout the text) as well as an estimated firing rate of each neuron (CNMFe parameters: spatial downsample factor=1, temporal downsample=1, gaussian kernel width=4, maximum neuron diameter=20, tau decay=1, tau rise=0.1). Only those neurons with an estimated firing rate of four transients/ minute or higher were considered 'task-active' and included in this paper – 278/330 (84%; each mouse contributed 49,57,67,12,6,27,60 neurons, respectively) of neurons recorded from PL-NAc passed this threshold while 256/328 (78%; each mouse contributed 17,28,20,46,47,40,13,13,32 neurons, respectively) passed in mTH-NAc. Across all figures, to normalize the neural activity across different neurons and between mice, we Z-scored each GCaMP6f recording trace using the mean and standard deviation calculated using the entire recording session.

Optogenetic stimulation of PL-NAc neurons

22 male C57BL/6J mice were bilaterally injected in either the PL (n=14 mice, M–L ± 0.4, A–P 2.0 and D–V −2.5 mm) or mTH (n=8 mice, M–L ± 0.3, A–P -0.7 and D–V −3.6 mm) with 600nl AAV2/5-CaMKIIa-hChR2-EYFP (UPenn vector core, injected 0.6 μl per hemisphere of titer of 9.6×10^{13} pp per ml). Optical fibers (300 μm core diameter, 0.37 NA) delivering 1–3 mW of 447 nm laser light (measured at the fiber tip) were implanted bilaterally above the NAc Core at a 10 degree angle (M–L ± 1.1, A–P 1.4 and D–V −4.2 mm). An additional cohort of control mice (n=8)

were implanted with optical fibers in the NAc without injection of ChR2 and underwent the same stimulation protocol outlined below (Supplementary Figure 2.19e-h). Mice were anesthetized for implant surgeries with isoflurane (3–4% induction and 1–2% maintenance). Mice were given 5 days of recovery after the surgical procedure before behavioral testing.

During behavioral sessions, 5 ms pulses of 1-3 mW, 447 nm blue light was delivered at 20 Hz on a randomly selected 10% of trials beginning when the mouse entered the central nose poke. Light stimulation on unrewarded trials ended 1s after the end of the CS– presentation. On rewarded trials, light administration ended either 1s after CS+ presentation ('cohort 1') or at the end of reward consumption, as measured by the mouse not engaging the reward port for 100ms ('cohort 2'). See Supplementary Figure 2.20 for a schematic of stimulation times as well as the behavior of the two cohorts. Mice alternated between sessions with and without stimulation – sessions without stimulation were excluded from analysis. Anatomical targeting was confirmed as successful in all mice through histology after the experiment, and therefore no mice were excluded from this data set.

To quantify the effect of laser stimulation on latency times shown in Supplementary Figure 2.19a-d, we ran a mixed effects linear model using the fitglme package in MATLAB. In this model, the median latency to initiate a trial of a mouse, defined as the time between illumination of the central nose poke (i.e., trial start) and the mouse initiating a trial via nose poke, was predicted using i) opsin identity (PL-NAc CaMKII-ChR2, mTH-NAc CaMKII-ChR2 or no-opsin controls), ii) laser stimulation on the current trial, iii) laser stimulation on the previous trial, iv) the interaction between opsin identity and laser stimulation on the current trial and v) the interaction between opsin and laser stimulation on the previous trial. To account for individual variation between mice, a random effect of mouse ID was included.

2.4.2. Quantification and statistical analysis

Logistic choice regression

For the logistic choice regressions shown in Figure 2.1e and Supplementary Figure 2.9a, we modeled the choice of the mouse on trial i based on lever choice and reward outcome information from the previous n trials using the following logistic regression model:

$$\log \frac{C(i)}{1 - C(i)} = \beta_0 + \sum_{j=1}^{n} \beta_j^R R(i-j) + \sum_{j=1}^{n} \beta_j^U U(i-j) + error \tag{2.2}$$

Where $C(i)$ is the probability of choosing the right lever on trial i, and $R(i-j)$ and $U(i-j)$ are the choice of the mouse j trials back from the i^{th} trial for either rewarded or unrewarded trials, respectively. $R(i-j)$ was defined as +1 when the j^{th} trial back was both rewarded and a right press, -1 when the j^{th} trial back was rewarded and a left press and 0 when it was unrewarded. Similarly, was defined as +1 when the trial back was both unrewarded and a right press, -1 when the trial back was unrewarded and a left press and 0 when it was rewarded. The calculated regression coefficients, β_j^R and β_j^U, reflect the strength of the relationship between the identity of the chosen lever on a previously rewarded or unrewarded trial, respectively, and the lever chosen on the current trial.

To examine the effect of optogenetic stimulation from multiple trials back on the mouse's choice (Figure 2.7c,e,h,k; Supplementary Figure 2.19h & Supplementary Figure 2.20c−d), we expanded our behavioral logistic regression model to include the identity of those trials with optical stimulation, as well as the interaction between rewarded and unrewarded choice predictors and stimulation:

$$\begin{gathered}\log \frac{C(i)}{1 - C(i)} = \beta_0 + \sum_{j=1}^{n} \beta_j^R R(i-j) + \sum_{j=1}^{n} \beta_j^U U(i-j) + ... \\ \sum_{j=1}^{n} \beta_j^{LR} L(i-j)R(i-j) + \sum_{j=1}^{n} \beta_j^{LU} L(i-j)U(i-j) + \sum_{j=1}^{n} \beta_j^{L} L(i-j) + error\end{gathered} \tag{2.3}$$

where $L(i)$ represents optical stimulation on the i^{th} trial (1 for optical stimulation, 0 for control trials), β_j^L represents the coefficient corresponding to the effect of stimulation on choice j trials back, β_j^{LR} and β_j^{LU} represents the coefficients corresponding to the interaction between rewarded choice x optical stimulation and unrewarded choice x stimulation, respectively.

To visualize the relative influence of stimulation on the mice's choices compared with unstimulated trials, in Figure 2.7c,e,h,k; Supplementary Figure 2.19h & Supplementary Figure 2.20c-d, the solid blue trace represents the sum of the rewarded choice coefficients (represented by the black trace) and rewarded choice x stimulation coefficients $(\beta_j^R + \beta_j^{LR})$. Similarly, the dashed blue trace represents the sum of the unrewarded choice coefficients (grey trace) and unrewarded choice x stimulation coefficients $(\beta_j^U + \beta_j^{LU})$. For all choice regressions, the coefficients for each mouse were fit using the *glmfit* function in MATLAB and error bars represent mean ± SEM across mice.

Encoding model to generate response kernels for behavioral events

To determine the response of each neuron attributable to each of the events in our task, we used a multiple linear encoding model with lasso regularization to generate a response kernel for each behavioral event (example kernels shown in Figure 2.2b). In this model, the dependent variable was the GCaMP6f trace of each neuron recorded during a behavioral session and the independent variables were the times of each behavioral event ('nose poke', 'levers out', 'ipsilateral lever press', 'contralateral lever press', 'CS+', 'CS–' and 'reward consumption) convolved with a 25 degrees-of-freedom spline basis set that spanned -2 to 6s before and after the time of action events ('nose poke', 'ipsilateral press', 'contralateral press' and 'reward consumption') and 0 to 8s from stimulus events ('levers out', 'CS+' and 'CS–'). To generate this kernel, we used the following linear regression with lasso regularization using the lasso function in MATLAB:

$$\min_{\beta_0,\beta_{jk}} \left(\sum_{t=1}^{T} \left(F(t) - \sum_{k=1}^{K} \sum_{j=1}^{N_{sp}} \beta_{jk} X_{jk}(t) - \beta_0 \right)^2 + \lambda \sum_{k=1}^{K} \sum_{j=1}^{N_{sp}} |\beta_{jk}| \right) \tag{2.4}$$

where $F(t)$ is the Z-scored GCaMP6f fluorescence of a given neuron at time t, T is the total time of recording, K is the total number of behavioral events used in the model, N_{sp} is the degrees-of-freedom for the spline basis set (25 in all cases, splines generated using the FDAfuns MATLAB package), β_{jk} is the regression coefficient for the j^{th} spline basis function and k^{th} behavioral event,

β_0 is the intercept term and λ is the lasso penalty coefficient. The value of lambda was chosen for each neuron that minimized the mean squared error of the model, as determined by 5-fold cross validation. The predictors in our model, X_{jk}, were generated by convolving the behavioral events with a spline basis set, to enable temporally delayed versions of the events to predict neural activity:

$$X_{jk}(t) = \sum_{i=1}^{N=81} S_j(i) e_k(t-i) \tag{2.5}$$

where $S_j(i)$ is the j^{th} spline basis function at time point i with a length of 81 time bins (time window of -2 to 6s for action events or 0 to 8s for stimulus events sampled at 10 Hz) and e_k is a binary vector of length T representing the time of each behavioral event k (1 at each time point where a behavioral event was recorded using the MED associates and TDT software, 0 at all other timepoints).

Using the regression coefficients, β_{jk}, generated from the above model, we then calculated a 'response kernel' for each behavioral event:

$$kernel_k(t) = \sum_{j=1}^{N_{sp}} \beta_{jk} S_j(t) \tag{2.6}$$

This kernel represents the (linear) response of a neuron to each behavioral event, while accounting for the linear component of the response of this neuron to the other events in the task.

Quantification of neural modulation to behavioral events

To identify neurons that were significantly modulated by each of the behavioral events in our task (fractions shown in Figure 2.2g-h), we used the encoding model described above, but without the lasso regularization:

$$F(t) = \beta_0 + \sum_{k=1}^{K} \sum_{j=1}^{N_{sp}} \beta_{jk} X_{jk}(t) \tag{2.7}$$

As above, $F(t)$ is the Z-scored GCaMP6f fluorescence of a given neuron at time t, K is the total number of behavioral events used in the model, N_{sp} is the degrees-of-freedom for the spline basis set (25 in all cases), β_{jk} is the regression coefficient for the j^{th} spline basis function and k^{th} behavioral event and β_0 is the intercept term. To determine the relative contribution of each behavioral event when predicting the activity of a neuron, we compared the full version of this model to a reduced model with the X and β terms associated with the behavioral event in question excluded. For each behavioral event, we first generated an F-statistic by comparing the fit of a full model containing all event predictors with that of a reduced model that lacks the predictors associated with the event in question. We then calculated this same statistic on 500 instances of shuffled data, where shuffling was performed by circularly shifting the GCaMP6f fluorescence by a random integer. We then compared the F-statistic from the real data to the shuffled distribution to determine whether the removal of an event as a predictor compromised the model significantly more than expected by chance. If the resulting P-value was less than the significance threshold of P=0.01, after accounting for multiple comparison testing of each of the behavioral events by Bonferroni correction, then the event was considered significantly encoded by that neuron.

To determine whether a neuron was significantly selective to the choice or outcome of a trial ('choice-selective' and 'outcome-selective', fractions of neurons from each population shown in Figure 2.3a,c), we utilized a nested model comparison test similar to that used to determine significant modulation of behavioral events above, where the full model used the following behavioral events as predictors: 'nose poke', 'levers out', 'all lever press', 'ipsilateral lever press', 'all CS', 'CS+' and 'reward consumption'. For choice-selectivity, an F-statistic was computed for a reduced model lacking the 'ipsilateral lever press' predictors and significance was determined by comparing this value with a null distribution generated using shuffled data as described above. For outcome-selectivity, the reduced model used to test for significance lacked the predictors associated with both the 'CS+' and 'reward consumption' events.

By separating the lever press and outcome-related events into predictors that were either blind to the choice or outcome of the trial ('all lever press' and 'all CS', respectively) and those which included choice or outcome information ('ipsilateral lever press' or 'CS+' and 'reward consumption', respectively) we were able to determine whether the model was significantly impacted by the removal of either choice or outcome information. Therefore, neurons with significant encoding of the 'ipsilateral lever press' event (using the same P-value threshold determined by the shuffled distribution of F-statistics) were considered choice-selective, while those with significant encoding of the 'CS+/reward consumption' events were considered outcome-selective.

Neural decoders

Choice decoder In Figure 2.3b, we quantified how well simultaneously imaged populations of 1 to 10 PL-NAc or mTH-NAc neurons could be used to decode choice using a logistic regression:

$$log\left(\frac{C(i)}{1-C(i)}\right) = \beta_0 + \sum_{j=1}^{n} \beta_j X_j(i) + \epsilon \tag{2.8}$$

where C(i) is the probability the mouse made an ipsilateral choice on trial i, β_0 is the offset term, n is the number of neurons (between 1 and 10), β_j is the regression weight for each neuron, $X_j(i)$ is the mean z-scored GCaMP6f fluorescence from -2s to 6s around the lever press on trial i and ϵ is the error term.

Given that the mice's choices were correlated across neighboring trials, we weighted the logistic regression based on the frequency of each trial type combination. This was to ensure that choice decoding of a given trial was a reflection of the identity of the lever press on the current trial as opposed to that of the previous or future trial. Thus, we classified each trial as one of eight 'press sequence types' based on the following 'previous-current-future' press sequences: ipsi-ipsi-ipsi, ipsi-ipsi-contra, ipsi-contra-contra, ipsi-contra-ipsi, contra-contra-contra, contra-contra-ipsi, contra-ipsi-ipsi, contra-ipsi-contra. We then used this classification to equalize the effects of press-sequence type on our decoder by generating weights corresponding to the inverse of the frequency of the press sequence type of that trial. These weights were then used as an input to the *fitglm* function in MATLAB, which was used to fit a weighted version of the logistic regression model above (Equation 2.8).

Decoder performance was evaluated with 5-fold cross-validation by calculating the proportion of correctly classified held-out trials. Predicted ipsilateral press probabilities greater than or equal to

0.5 were decoded as an ipsilateral choice and values less than 0.5 were decoded as a contralateral choice. This was repeated with 100 combinations of randomly selected, simultaneously imaged neurons from each mouse. Reported decoding accuracy is the average accuracy across the 100 runs and 5 combinations of train-test data for each mouse. Note that only 6/7 mice in the PL-NAc cohort were used in the decoder analyses as one mouse had fewer than 10 simultaneously imaged neurons.

Outcome decoder For the outcome decoder in Figure 2.3d, we used the same weighted logistic regression used for choice decoding, except the dependent variable was the outcome of the trial (+1 for a reward, 0 for no reward) and the predictors were the average GCaMP6f fluorescence during the intertrial interval (ITI) of each trial. The ITI was defined as the time between CS presentation and either 1s before the next trial's nose poke or 8s after the CS, whichever occurred first. This was used in order to avoid including any neural activity attributable to the next trial's nose poke in our analysis.

To correct for outcome correlations between neighboring trials, we performed a similar weighting of predictors as performed in the choice decoder above using the following eight outcome sequence types: 'reward- reward- reward', 'reward- reward- unreward', 'reward- unreward- unreward', 'reward- unreward- reward', 'unreward- unreward- unreward', 'unreward- unreward- reward', 'unreward- reward- reward', 'unreward- reward- unreward.'

Time-course choice decoder To determine how well activity from PL-NAc and mTH-NAc neurons was able to predict the mouse's choice as a function of time throughout the trial (Figure 2.4e, Supplementary Figure 2.14e & Supplementary Figure 2.15e), we trained separate logistic regressions on 500ms bins throughout the trial, using the GCaMP6f fluorescence of 10 simultaneously imaged neurons.

Because of the variability in task timing imposed by the jitter and variability of the mice's actions, we linearly interpolated the GCaMP6f fluorescence trace of each trial to a uniform length, $t_{adjusted}$, relative to behavioral events in our task. Specifically, for each trial, T, we divided time into the following four epochs: (i) 2s before nose poke, (ii) time from the nose poke to the lever press, (iii) time from the lever press to the nose poke of the subsequent trial, $T+1$ and (iv) the 3s following the next trial nosepoke. For epochs ii and iii, $t_{adjusted}$ was determined by interpolating the GCaMP6f fluorescence trace from each trial to a uniform length defined as the median time between the flanking events across all trials. Thus, $t_{adjusted}$ within each epoch for each trial, T, was defined as:

$$T_{adjusted}(t) \equiv \begin{cases} t & , t_{np}^{T} - 2 \leq t < t_{np}^{T} \\ 2 + \frac{(t - t_{np}^{T})}{(t_{lp}^{T} - t_{np}^{T})} \widetilde{ep_{ii}} & , t_{np}^{T} \leq t < t_{lp}^{T} \\ 2 + \widetilde{ep_{ii}} + \frac{(t - t_{lp}^{T})}{(t_{np}^{T+1} - t_{lp}^{T})} \widetilde{ep_{iii}} & , t_{lp}^{T} \leq t < t_{np}^{T+1} \\ t & , t_{np}^{T+1} \leq t < t_{np}^{T+1} + 3 \end{cases} \tag{2.9}$$

where t_{np}^{T}, and t_{lp}^{T} are the times of the nose poke and lever press on the current trial, t_{np}^{T+1} is the time of the nose poke of the subsequent trial and $\widetilde{ep_{ii}}$ and $\widetilde{ep_{iii}}$ are the median times across trials of epoch ii and iii.

The resulting time-adjusted GCaMP6f traces were divided into 500ms bins. For each bin, we fit the weighted logistic regression described above to predict choice on the current, previous or future trial from the activity of 10 simultaneously imaged neurons. Predictors were weighted based on press sequence type as described above. Decoding accuracy was assessed as described above using 100 combinations of 10 randomly selected neurons and 5-fold cross-validation. To determine if decoding was significantly above chance, which is 0.5, for each timepoint we performed a two-tailed, one-sample t-test.

Statistics

All t-tests reported in the results and as specified in each figure legend were performed using either the *ttest* or *ttest2* function in MATLAB. In all cases, t-tests were two-tailed. In cases where multiple comparisons were performed, we applied a Bonferroni correction to determine the significance threshold. Two-proportion Z-tests (used to compare fractions of significantly modulated/selective neurons, Figures 2h & 3a,c) and Fisher's Z (used to compare correlation coefficients, Figure 2.4d & Supplementary Figure 2.14d) were performed using Vassarstats.net. Asterisks indicating significance thresholds are referenced in respective figure legends.

For all t-tests in this paper, data distributions were assumed to be normal, but this was not formally tested. No statistical methods were used to predetermine sample sizes, but our sample sizes were similar to those generally employed in the field.

2.4.3. Synaptic plasticity model

To computationally model how the brain could solve the reversal learning task using fast dopamine-mediated synaptic plasticity, we generated a biological instantiation of the TD algorithm for reinforcement learning [93] by combining the recorded PL-NAc activity with known circuit connectivity in the NAc and associated regions [18,19,169,170]. The goal of the model is to solve the "temporal credit assignment problem" by learning the value of each choice at the onset of the choice-selective PL-NAc sequence, when we assume the mouse makes its decision and which is well before the time of reward.

Synaptic plasticity model description

The value function Our implementation of the TD algorithm seeks to learn an estimate, at any given time, of the total discounted sum of expected future rewards, known as the value function $V(t)$. To do this, we assume that the value function over time is decomposed into a weighted sum of temporal basis functions $f_i^R(t)$ and $f_i^L(t)$ [93] corresponding to the right-choice and left-choice preferring neurons:

$$\begin{aligned} V_R(t) &= \sum_{i=1}^{n_R} w_i^R(t) f_i^R(t) \\ V_L(t) &= \sum_{i=1}^{n_L} w_i^L(t) f_i^L(t) \end{aligned} \tag{2.10}$$

with the total value being given by the sum over both the left and right neurons as

$$V(t) = V_R(t) + V_L(t) \tag{2.11}$$

Here, $V_R(t)$ and $V_L(t)$ are the components of the value functions encoded by the right- and left-preferring neurons respectively, n_R and n_L are the number of right- and left-preferring choice-selective neurons respectively, and $w_i^{R,L}$ are the weights between the i^{th} PL neuron and the NAc, which multiply the corresponding basis functions. Thus, each term in $V_R(t)$ or $V_L(t)$ above corresponds to the activity of one of the striatal neurons in the model (Figure 2.5a). Note that, in our model, the total value $V(t)$ sums the values associated with the left and right actions and is thus not associated with a particular action. At any given time on a given trial, the choice-selective activity inherent to the recorded PL-NAc neurons results predominantly in activation of the sequence corresponding to the chosen lever compared to the unchosen lever (see Figure 2.5c), so that a single sequence, corresponding to the chosen action, gets reinforced.

The reward prediction error (RPE) TD learning updates the value function iteratively by computing errors in the predicted value function and using these to update the weights w_i. The RPE at each moment of time is calculated from the change in the estimated value function over a time step of size dt as follows

$$\text{RPE} = \delta(t)\text{d}t = r(t)\text{d}t + e^{\frac{-\text{d}t}{\tau}} V(t) - V(t - \text{d}t) \tag{2.12}$$

where $\delta(t)$ is the reward prediction error per unit time. Here, the first two terms represent the estimated value at time t, which equals the sum of the total reward received at time t and the (discounted) expectation of rewards, i.e. value, at all times into the future. This is compared to the previous time step's estimated value $V(t - dt)$. The coefficient $e^{\frac{-\text{d}t}{\tau}}$ represents the temporal discounting of rewards incurred over the time step dt. Here τ denotes the timescale of temporal discounting and was chosen to be 0.7s.

To translate this continuous time representation of RPE signals to our biological circuit model, we assume that the RPE $\delta(t)$ is carried by dopamine neurons [95, 171]. These dopamine neurons receive three inputs corresponding to the three terms on the right side of the above equation: a reward signal originating from outside the VTA, a discounted estimate of the value function $V(t)$ that, in Figure 2.5a, represents input from the striatum via the ventral pallidum [103, 163] and an oppositely signed, delayed copy of the value function $V(t-\Delta)$ that converges upon the VTA interneurons [99].

Because the analytical formulation of TD learning in continuous time is defined in terms of the infinitesimal time step dt, but a realistic circuit implementation needs to be characterized by a finite delay time for the disynaptic pathway through the VTA interneurons, we rewrite the above equation approximately for small, but finite delay Δ as:

$$\delta(t)\mathrm{d}t = r(t)\mathrm{d}t + \frac{\gamma V(t) - V(t-\Delta)}{\Delta}\mathrm{d}t \tag{2.13}$$

where we have defined $\gamma = e^{\frac{-\Delta}{\tau}}$ as the discount factor corresponding to one interneuron time delay and, in all simulations, we chose a delay time $\Delta = 0.01s$. Note that the discount factor is biologically implemented in different strengths of the weights of the VP inputs to the GABA interneuron and dopaminergic neuron in the VTA.

The proposed circuit architecture of Figure 2.5a can be rearranged into several other, mathematically equivalent architectures (Supplementary Figure 2.16). These architectures are not mutually exclusive, so other more complicated architectures could be generated by superpositions of these architectures.

The eligibility trace The RPE at each time step $\delta(t)\mathrm{d}t$ was used to update the weights of the recently activated synapses, where the "eligibility" $E_i(t)$ of a synapse for updating depends upon an exponentially weighted average of its recent past activity [93, 101]:

$$E_i(t) = \int_{-\infty}^{t} e^{\frac{s-t}{\tau_e}} f_i(s)\,\mathrm{d}s \tag{2.14}$$

which can be rewritten as

$$\frac{\mathrm{d}E_i(t)}{\mathrm{d}t} = -\frac{E_i(t)}{\tau_e} + f_i(t) \tag{2.15}$$

or, in the limit $dt << 1$,

$$E_i(t) \approx e^{-\frac{\mathrm{d}t}{\tau_e}} E_i(t - \mathrm{d}t) + f_i(t)\mathrm{d}t \tag{2.16}$$

where τ_e defines the time constant of the decay of the eligibility trace, which was chosen to be $0.8s$ consistent with [101, 172].

Weight Updates The weight of each PL-NAc synapse, w_i, is updated according to the product of its eligibility $E_i(t)$ and the RPE rate $\delta(t)$ at that time using the following update rule [93, 101] :

$$\begin{aligned} \frac{\mathrm{d}\hat{w}_i(t)}{\mathrm{d}t} &= \alpha\delta(t)E_i(t) \\ w_i(t) &= max[0, \hat{w}_i(t)] \end{aligned} \tag{2.17}$$

where $\alpha = 0.009(spikes/s)^{-1}$ was the learning rate. Note that the units of α derive from the units of weight being value $\cdot$ $(\text{spikes/s})^{-1}$. The PL-NAc weights used in the model are thresholded to be non-negative so that the weights obey Dale's principle.

Action Selection In the model, the decision to go left or right is determined by "probing" the relative values of the left versus right actions just prior to the start of the choice-selective sequence. To implement this, we assumed that the choice was read out in a noisy, probabilistic manner from the activity of the cluster of neurons that responded at the time choice-selectivity robustly appeared, when we assume the decision is made. This corresponded to the first 17 neurons in each (left or right) PL population prior to the start of the sequential activity. This was accomplished by providing a 50 ms long, noisy probe input to each of these PL neurons and reading out the summed activity of the left and the summed activity of the right striatal populations. The difference between these summed activities was then put through a softmax function (given below) to produce the probabilistic decision.

To describe this decision process quantitatively, we define the probability of making a leftward or rightward choice in terms of underlying decision variables d_{left} and d_{right} corresponding to the summed activity of the first 17 striatal neurons in each population:

$$\begin{aligned} d_{left} &= \mathbb{E}_t\left[\sum_{i=1}^{17} w_i^{left} n_i^{left}(t)\right] \\ d_{right} &= \mathbb{E}_t\left[\sum_{i=1}^{17} w_i^{right} n_i^{right}(t)\right] \end{aligned} \tag{2.18}$$

where $\mathbb{E}_t[.]$ denotes time-averaging over the 50 ms probe period and $n_i^{left}(t)$ and $n_i^{right}(t)$ denote the non-negative stochastic probe input, which was chosen independently for each neuron and each time step from a normal distribution (truncated at zero to enforce non-negativity) with mean prior to truncation equal to 0.05 s^{-1} (5% of peak activity) and a standard deviation of $0.0025/\sqrt{\mathrm{d}t}$ s^{-1}. Note that the weights $w_i^{left/right}$ used here correspond to the weights from the end of the previous trial, which we assume are the same as the weights at the beginning of the next trial. The probability of choosing the left or the right lever for a given trial n is modeled as a softmax function of these decision variables plus a "stay with the previous choice" term that models the tendency of mice in our study to return to the previously chosen lever irrespective of reward (Figure 2.1d), given by the softmax distribution

$$\begin{aligned} Prob(left) &= \frac{exp(\beta_{value} d_{left} + \beta_{stay} I_{left})}{exp(\beta_{value} d_{left} + \beta_{stay} I_{left}) + exp(\beta_{value} d_{right} + \beta_{stay} I_{right})} \\ Prob(right) &= \frac{exp(\beta_{value} d_{right} + \beta_{stay} I_{right})}{exp(\beta_{value} d_{left} + \beta_{stay} I_{left}) + exp(\beta_{value} d_{right} + \beta_{stay} I_{right})} \end{aligned} \tag{2.19}$$

where $I_{left/right}$ is 1 if that action (i.e. left or right) was chosen on the previous trial and 0 otherwise, and $\beta_{value} = 7000$ and $\beta_{stay} = 0.15$ are free parameters that define the width of the softmax distribution and the relative weighting of the value-driven versus stay contributions to the choice.

Synaptic plasticity model implementation

Block structure for the model Block reversals were determined using the same criteria as in the probabilistic reversal learning task performed by the mice – the identity of the rewarded lever reversed after 10 rewarded trials plus a random number of trials drawn from the geometric distribution given by Equation 2.1. The model used p=0.4 as in the reversal learning experiments. Given the variation in performance across the models that use PL-NAc, mTH-NAc or early-only activity as input (see Figure 2.5), the average block length for each model varied as well (because block reversals depended upon the number of rewarded trials). The average block length for the single-trial PL-NAc model, single-trial mTH-NAc model and early-only control were 23.0±7.6, 28.1±8.8 and 25.1±6.3 trials (mean ± std. dev.), respectively. The PL-NAc model produced a similar block length as that of behaving mice (23.2±7.9 trials, mean ± std. dev.). Because a block reversal in our task is dependent on the mice receiving a set number of rewards, the choices just prior to a block reversal are more likely to align with the identity of the block and result in reward (see Figure 2.5e,j,o). Thus, the increase in choice probability observed on trials close to the block reversal is an artifact of this reversal rule and not reflective of the model learning choice values.

PL-NAc inputs to the neural circuit model To generate the temporal basis functions $f_i(t)$ (example activity shown in Figure 2.5c), we used the choice-selective sequential activity recorded from the PL-NAc neurons shown in Figure 2.4b-c. Spiking activity was inferred from calcium fluorescence using the CNMFe algorithm [168] and choice-selectivity was determined using the nested comparison model used to generate Figure 2.3a (see **Quantification of neural modulation to behavioral events** above for details). Model firing rates were generated by Z-scoring the inferred spiking activity of each choice-selective PL-NAc neuron. The resulting model firing rates were interpolated using the interp function from Python's numpy package to match the time step, $dt = 0.01s$, and smoothed using a Gaussian kernel with zero mean and a standard deviation of 0.2s using the *gaussian_filter1d* function from the ndimage module in Python's SciPy package.

To generate a large population of model input neurons on each trial, we created a population of 368 choice-selective "pseudoneurons" on each trial. This was done as follows: for each simulated trial, we created 4 copies (pseudoneurons) of each of the 92 recorded choice-selective PL-NAc neurons using that neuron's inferred spiking activity from 4 different randomly selected trials. The pool of experimentally recorded trials from which pseudoneuron activities were chosen was balanced to have equal numbers of stay and switch trials. This was done because the choices of the mice were strongly positively correlated from trial to trial (i.e., had more stay than switch trials), which (if left uncorrected) potentially could lead to biases in model performance if activity late in a trial was reflective of choice on the next, rather than the present trial. To avoid choice bias in the model, we combined the activity of left- and right-choice-preferring recorded neurons when creating the pool of pseudoneurons. We then randomly selected 184 left-choice-preferring and 184 right-choice-preferring model neurons from this pool of pseudoneurons. An identical procedure, using the 92 most choice-selective mTH-NAc neurons, was followed to create the model mTH-NAc neurons. The identity of these 92 neurons was determined by ranking each neuron's choice-selectivity using the p-value calculated to determine choice-selectivity (see **Quantification of neural modulation to behavioral events** above for details).

To generate the early-only control activity (example activity shown in Figure 2.5m), similar to the PL-NAc activity, we created a population of 368 pseudoneurons on each trial that were divided into 184 left-choice-preferring and 184 right-choice-preferring pseudoneurons. However, in this case, we only used the early-firing neurons (neurons active at the onset of the sequence) of the PL-NAc population to create the pseudoneurons. Thus, for this control simulation, all neurons contribute to the decision as they are all active at the onset of the sequence when the model makes its choice. More specifically, to create a pool of pseudoneurons, we created multiple copies of each of the first 17 neurons of the left-choice-preferring and right-choice-preferring PL-NAc population, where each copy corresponds to the activity of the neuron on a different randomly selected trial. We then randomly selected 184 left-choice-preferring and 184 right-choice-preferring model neurons from this pool of pseudoneurons. We used a smaller learning rate $\alpha = 0.003(spikes/s)^{-1}$ in this case in order to prevent the PL-NAc synaptic weights from exhibiting unstable growth. We also adjust $\beta_{value} = 1000$ in order to match the stay probability following rewarded trials to that of the model with recorded PL-NAc input (Figure 2.5p).

To mimic the PL-NAc activity during the optogenetic stimulation of PL-NAc neurons (Figure 2.7b-c), we set $f_i^{R,L}(t)$ equal to 0.2 for a randomly selected 70% of PL neurons, at all times t, from the time of the simulated nosepoke to 2s after the reward presentation. These 'stimulation trials' occurred on a random 10% of trials. 70% of PL neurons were activated to mimic the incomplete penetrance of ChR2 viral expression.

Reward input to the neural circuit model The reward input $r(t)$ to the dopamine neurons was modeled by a truncated Gaussian temporal profile centered at the time of the peak reward:

$$r(t) = R(i)\frac{1}{\sqrt{2\pi\sigma_r^2}}e^{-\frac{(t-\mu_r)^2}{2\sigma_r^2}} \tag{2.20}$$

where $R(i)$ is 1 if trial i was rewarded and 0 otherwise, μ_r is the time of peak reward and σ_r defines the width of the Gaussian (0.3s in all cases, width chosen to approximate distribution of dopamine activity in response to reward stimuli observed in previous studies such as [95, 173]). For each trial, a value of μ_r was randomly drawn from a uniform distribution spanning 0.2-1.2s from the time of the lever press. This distribution was chosen to reflect the 1s jitter between lever press and reward used in our behavioral task (see Methods above) as well as the observed delay between reward presentation and peak dopamine release in a variety of studies [26, 99, 173, 174]. To ensure that no residual reward response occurred before the time of the lever press, $r_a(t)$ was set to 0 for any time t that was 0.2s before the time of the peak reward, μ_r.

Initial weights The performance of the model does not depend on the choice of the initial weights as the model learns the correct weights by the end of the first block irrespective of the chosen initial weights. We chose the initial weights to be zero.

Weight and eligibility update implementation We assumed that the weight and eligibility trace updates start at the time of the simulated nose poke. The nose poke time, relative to the time of the lever press, varies due to a variable delay between the nose poke and the lever presentation as well as variation in time between lever presentation and lever press. To account for this, the weight and eligibility trace updates are initiated at time $t = t_{start}$, where t_{start} was drawn from a Gaussian distribution with a mean at –2.5s, and a variance of 0.2s, which was approximately both the time of the nose poke and the time at which choice-selective sequences initiated in the experimental recordings. The eligibility trace is reset to zero at the beginning of each trial. We stopped updating the weights at the end of the trial, defined as 3s after the time of lever press. The eligibility traces were updated according to Equation 2.16. The weights were updated by integrating Equation 2.17 with a first-order forward Euler routine. In all simulations, we used a simulation time step $\mathrm{d}t = 0.01s$.

2.4.4. Neural dynamics model

To computationally model how the brain could solve the reversal learning task without fast dopamine-mediated synaptic plasticity, we used an actor-critic network based on the meta-RL framework introduced by [79]. The model actor and critic networks are recurrent neural networks of Long Short-Term Memory (LSTM) units whose weights are learned slowly during the training phase of the task. The weights are then frozen during the testing phase so that fast reversal learning occurs only through the activation dynamics of the recurrent actor-critic network. Like the synaptic plasticity model, we input recorded PL-NAc activity to a value-generating "critic" network (conceived of as NAc, VP, and associated cortical regions) to generate appropriate reward prediction error signals in dopamine neurons. Unlike the synaptic plasticity model, the reward prediction error signals in this model are sent to an explicit actor network (conceived of as DMS and associated cortical regions), where they act as an input to help generate appropriate action signals based on reward history.

Neural dynamics model description

LSTM The model comprises two separate fully connected, gated recurrent neural networks of LSTM units, one each for the actor and critic network. An LSTM unit works by keeping track of a 'long-term memory' state ('memory state' $\boldsymbol{c}(t)$, also known as cell state) and a 'short-term memory' state ('output state' $\boldsymbol{h}(t)$, also known as hidden state) at all times. To regulate the information to be kept or discarded in the memory and output states, LSTMs use three types of gates: the input gate $\boldsymbol{i}(t)$ regulates what information is input to the network, the forget gate $\boldsymbol{\phi}(t)$ regulates what information to forget from the previous memory state, and the output gate $\boldsymbol{o}(t)$ (not to be confused with the output state $\boldsymbol{h}(t)$) regulates the output of the network. More precisely, the dynamics of an LSTM is defined by the following equations:

$$
\begin{aligned}
\boldsymbol{\phi}(t) &= \sigma(\boldsymbol{W}_\phi \boldsymbol{x}(t) + \boldsymbol{U}_\phi \boldsymbol{h}(t-\Delta t) + \boldsymbol{b}_\phi) \\
\boldsymbol{i}(t) &= \sigma(\boldsymbol{W}_i \boldsymbol{x}(t) + \boldsymbol{U}_i \boldsymbol{h}(t-\Delta t) + \boldsymbol{b}_i) \\
\boldsymbol{o}(t) &= \sigma(\boldsymbol{W}_o \boldsymbol{x}(t) + \boldsymbol{U}_o \boldsymbol{h}(t-\Delta t) + \boldsymbol{b}_o) \\
\boldsymbol{c}(t) &= \boldsymbol{\phi}(t) \odot \boldsymbol{c}(t-\Delta t) + \boldsymbol{i}(t) \odot tanh(\boldsymbol{W}_c \boldsymbol{x}(t) + \boldsymbol{U}_c \boldsymbol{h}(t-\Delta t) + \boldsymbol{b}_c) \\
\boldsymbol{h}(t) &= \boldsymbol{o}(t) \odot tanh(\boldsymbol{c}(t))
\end{aligned}
\tag{2.21}
$$

where $\boldsymbol{x}(t)$ is the vector of external inputs to the LSTM network at time step t, $\boldsymbol{W}_q$ and $\boldsymbol{U}_q$ are the weight matrices of the input and recurrent connections, respectively, where the subscript q denotes the state or gate being updated, $\boldsymbol{b}_q$ are the bias vectors, $\odot$ denotes element-wise multiplication and σ denotes the softmax function.

Critic network As in the synaptic plasticity model, the goal of the critic is to learn the value (discounted sum of future rewards) of a given choice at any time in a trial. The learned value signal can then be used to generate the RPE signals that are sent to the actor. The critic is modeled as a network of LSTM units that linearly project through trainable weights to a value readout neuron that represents the estimated value $V(t)$ at time step t. The critic takes as input the reward received $r(t)$ and the experimentally recorded PL-NAc choice-selective sequential input $C(t)$. The PL-NAc input provides the critic with a representation of the chosen side on the current trial as well as the time during the trial. This allows the critic to output an appropriately timed value signal (and consequently an appropriately timed RPE signal) corresponding to the chosen side. The reward input acts as a feedback signal to the critic that provides information about the correctness of the chosen action.

To map the critic to a biological neural circuit, we hypothesize that NAc, together with VP and associated cortical areas, form the critic recurrent neural network (Figure 2.6a; [108, 109, 110, 111, 112]). The choice-selective sequential input $C(t)$ to the critic is provided by the recorded choice-selective sequential activity in PL-NAc neurons (Figure 2.6a).

The reward prediction error (RPE) As in the synaptic plasticity model (Figure 2.5a), the RPE $\delta(t)$ is computed in the VTA DA neurons based on the value signal from the critic network (Figure 2.6a).

$$\delta(t) = r(t) + \gamma V(t) - V(t - \Delta t) \tag{2.22}$$

Unlike the synaptic plasticity model, the RPE signal is conveyed by the VTA dopamine neurons to the actor network. Note that the delay of the negative value signal equals one time step $\Delta t = 0.1s$ in this model, rather than the smaller delay $\Delta = 0.01s$ for the synaptic plasticity model. This is because the neural dynamics model used a larger time step for simulations due to limitations in computational power.

Actor network In contrast to the synaptic plasticity model, in which actions were directly readout from the activity of the value neurons early in the trial, we consider an explicit actor network that generates actions. The actor is modeled as a network of LSTM units that compute the policy, i.e., the probability of choosing an action $a(t)$ at time step t given the current state of the network. The policy is represented by three policy readout neurons, corresponding to choosing left, right or 'do nothing', whose activities are given by a (trainable) linear readout of the activities of the actor LSTM units. The actor receives three inputs: (i) an efference copy of the action taken at the previous time step $a(t - \Delta t)$, (ii) a 'temporal context' input $\xi(t)$, encoded as a vector of all 0s except for a value of 1 in the entry corresponding to the current time point t, that provides the actor with a representation of the time within the trial, and (iii) the reward prediction error at the current time step $\delta(t)$.

To map the actor to a biological neural circuit, we hypothesize that the DMS and associated cortical areas form the actor recurrent neural network (Figure 2.6a; [108, 110, 175, 176]). The temporal sequence input $\xi(t)$ to the actor is assumed to originate in the hippocampus or other cortical areas (Figure 2.6a; [113, 177, 178, 179]).

Training algorithm To train the recurrent weights of the network, which are then held fixed during task performance, we implement the Advantage Actor-Critic algorithm [**27**] on a slightly modified version of the reversal learning task (see "Block structure for training" section below). In brief, the weights of the neural network are updated via gradient descent and backpropagation through time. The loss function for the critic network, $\mathcal{L}_{critic}$, defines the error in the estimated value function. The synaptic weight parameters θ_v of the critic network are updated through gradient descent on the critic loss function $\mathcal{L}_{critic}$:

$$\begin{aligned} \Delta\theta_v &= -\alpha\nabla\mathcal{L}_{critic} \\ \nabla\mathcal{L}_{critic} &= -\beta_v\delta_t(s_t;\theta_v)\frac{\partial V}{\partial\theta_v} \end{aligned} \tag{2.23}$$

where α is the learning rate, s_t is the state at time step t, V denotes the value function and β_v is the scaling factor of the critic loss term. $\delta_t(s_t;\theta_v)$ is the k-step return temporal difference error (not to be confused with the RPE input to the actor defined in Equation 2.22) defined as follows:

$$\delta_t(s_t;\theta_v) = R_t - V(s_t;\theta_v)$$

where R_t is the discounted k-step bootstrapped return at time t

$$R_t = \sum_{i=0}^{k-1}\left(r_{t+i}\prod_{j=0}^{i}\gamma_{t+j}\right) + V(s_{t+k};\theta_v)\prod_{j=0}^{k}\gamma_{t+j}$$

where r_t is the reward received at time step t, γ_t is the discount factor at time step t (defined below), and k is the number of time steps until the end of an episode.

The loss function for the actor network, $\mathcal{L}_{actor}$, is given by a weighted sum of two terms: a policy gradient loss term, which enables the actor network to learn a policy $\pi(a_t|s_t)$ that approximately maximizes the estimated sum of future rewards $V(s_t)$, and an entropy regularization term that maximizes the entropy of the policy π to encourage the actor net.work to explore by avoiding premature convergence to suboptimal policies. The gradient of the actor loss function $\mathcal{L}_{actor}$ with respect to the synaptic weight parameters of the actor network, θ, is given by

$$\begin{aligned}\Delta\theta &= -\alpha\nabla\mathcal{L}_{actor} \\ \nabla\mathcal{L}_{actor} &= -\frac{\partial \log \pi(a_t|s_t;\theta)}{\partial\theta}\delta_t(s_t;\theta_v) - \beta_e\frac{\partial H(s_t;\theta)}{\partial\theta}\end{aligned} \tag{2.24}$$

where a_t is the action at time step t, π is the policy, β_e is the scaling factor of the entropy regularization term and $H(s_t;\theta)$ is the entropy of the policy π

$$H(s_t;\theta) = -\sum_{a\in A}\pi(a|s_t;\theta)\log\pi(a|s_t;\theta)$$

where A denotes the space of all possible actions.

Neural dynamics model implementation

LSTM Both the actor and critic LSTM networks consisted of 128 units each and were implemented using TensorFlow's Keras API. The weight matrices $\boldsymbol{U}_q$ were initialized using Keras's '*glorot_uniform*' initializer, the weight matrices $\boldsymbol{W}_q$ were initialized using Keras's 'orthogonal' initializer and the biases $\boldsymbol{b}$ were initialized to 0. The output and memory states for both LSTM networks were initialized to zero at the beginning of each training or testing episode.

PL-NAc inputs to the critic Input to the critic was identical to the smoothed, single-trial input used for the synaptic plasticity model described above, except i) activity was not interpolated because each time step in this model was equivalent to the sampling rate of the collected data (10 Hz), and ii) we chose to input only the activity from 2s before to 2s after the lever press (as compared to 3s after the lever press for the synaptic plasticity model) in order to reduce the computational complexity of the training process. To reduce episode length, and therefore training time, we also excluded those neurons whose peak activity occurred more than 2s after the lever press, reducing the final number of 'pseudoneurons' used as input to 306 (compared with 368 for the synaptic plasticity model).

Optogenetic-like stimulation of the PL-NAc population (Figure 2.7d-e) was performed in a similar manner to the synaptic plasticity model, with activity set to 0.15 for a randomly selected 70% of neurons for the duration of the trial.

Trial structure Each trial was 4s long starting at 2s before lever press and ending at 2s after lever press. At any given time, the model has three different choices: choose left, choose right or do nothing. Similar to the synaptic plasticity model, the model makes its decision to choose left or right at the start of a trial, which then leads to the start of the corresponding choice-selective sequential activity. However, unlike the synaptic plasticity model, the model can also choose 'do nothing' at the first time step, in which case an activity pattern of all zeros is input to the critic for the rest of the trial. For all other time steps, the correct response for the model is to 'do nothing'. Choosing 'do nothing' on the first time step or choosing something other than 'do nothing' on the subsequent time steps results in a reward $r(t)$ of -1 at that time. If a left or right choice is made on the first time step, then the current trial is rewarded based on the reward probabilities of the current block (Figure 2.1a) and the reward input $r(t)$ to the critic is modeled by a truncated Gaussian temporal profile centered at the time of the peak reward (Equation 2.20) with the same parameters as in the synaptic plasticity model.

Block structure for training We used a slightly modified version of the reversal learning task performed by the mice in which the block reversal probabilities were altered in order to make the block reversals unpredictable. This was done to discourage the model from learning the expected times of block reversals based on the number of rewarded trials in a block and to instead mimic the results of our behavioral regressions (Figure 2.1e) suggesting that the mice use only the previous $\sim$4 trials to make a choice. To make the block reversals unpredictable, the identity of the high-probability lever reversed after a random number of trials drawn from a geometric distribution (Equation 2.1) with p=0.9.

Training Each training episode was chosen to be 15 trials long and the model was trained for 62000 episodes. For this model, we used a time step $\Delta t = 0.1s$. The values of the training hyperparameters were as follows: the scaling factor of the critic loss term $\beta_v = 0.05$, the scaling factor of the entropy regularization term $\beta_e = 0.05$, the learning rate $\alpha = 0.01s^{-1}$ ($\alpha = 0.001$ per time step), and the timescale of temporal discounting within a trial $\tau = 2.45s$, leading to a discount factor $\gamma = e^{-\Delta t/\tau} = 0.96$ for all times except for the last time step of a trial when the discount factor was 0 to denote the end of a trial. The network's weights and biases were trained using the RMSprop gradient descent optimization algorithm [180] and backpropagation through time, which involved unrolling the LSTM network over an episode (630 time steps).

Block structure for testing Block reversal probabilities for the testing phase were the same as in the probabilistic reversal learning task performed by the mice. The average block length for the PL-NAc neural dynamics model was 19.3±5.0 trials (mean±std. dev.).

Testing The model's performance (Figures 6b-j) was evaluated in a testing phase during which all network weights were held fixed so that reversal learning was accomplished solely through the neural dynamics of the LSTM networks. The network weights used in the testing phase were the weights learned at the end of the training phase. A testing episode was chosen to be 1500 trials long and the model was run for 120 episodes.

Actor network analysis For Figures 6g-j, we tested the model's performance on a slightly modified version of the reversal learning task in which, after training, block lengths were fixed at 30 trials. This facilitated the calculation and interpretation of the block-averaged activity on a given trial of a block. Dimensionality reduction of the actor network activity (Figure 2.6h) was performed using the PCA function from the decomposition module in Python's scikit-learn package.

Replacing sequential input to the critic with persistent input. In Figure 2.6f, we analyzed how model performance changed when the temporal structure provided by the choice-selective sequential inputs to the critic were replaced during training by persistent choice-selective input. The persistent choice-selective input was generated by setting the activity of all the left-choice selective neurons to 1 and all the right-choice selective neurons to 0 for all time points on left-choice trials and vice versa on right-choice trials.

2.4.5. Cross-trial analysis of RPE in dopamine neurons

To generate the regression coefficients in Figure 2.5g,l,q, Figure 2.6e and Supplementary Figure 2.17c,d, we performed a linear regression analysis adapted from [102], which uses the mouse's reward outcome history from the current and previous 5 trials to predict the average dopamine response to reward feedback on a given trial, i:

$$D(i) = \beta_0 + \sum_{j=0}^{5} \beta_j \hat{R}(i-j) + error \tag{2.25}$$

where $D(i)$ is the average dopamine activity from 0.2 to 1.2s following reward feedback on trial i, $\hat{R}(i-j)$ is the reward outcome j trials back from trial i (1 if j trials back is rewarded and 0 if unrewarded) and β_j are the calculated regression coefficients that represent the effect of reward outcome j trials back on the strength of the average dopamine activity, $D(i)$. For the regression coefficients generated from recorded dopamine activity (Supplementary Figure 2.17c,d) we used the Z-scored GCaMP6f fluorescence from VTA-NAc terminal recordings of 11 mice performing the same probabilistic reversal learning task described in this paper (for details see [26]). The regression coefficients for the experimental data as well as the synaptic plasticity and neural dynamics model simulations were fit using the *LinearRegression* function from the $linear_model$ module in Python's scikit-learn package.

2.4.6. Simulation of model-free versus model-based task performance

In order to identify possible RPE signatures that distinguish ideal observer ("model-based") versus Q-learning ("model-free") behavior in this task (Supplementary Figure 2.21), we simulated choices using the two models. Based on the dopaminergic signature of block reversal inference reported in [181], we first confirmed that our ideal observer and Q-learning models gave rise to distinct dopamine signatures when performing the task used in [181]. In that task, reward probabilities were 100% and 0% for the "high probability" and "low probability" choices, respectively, and the reward probabilities reversed with a 5% probability on each trial. Next, we applied the same framework to our task, to determine if we could observe similar distinctions between the models. In this case, the reward probabilities were 70% and 10%, as in the task studied in this paper, and blocks reversed with a 5% probability on each trial, which resulted in block lengths comparable to those observed in our experiments.

Ideal observer model The ideal observer model was provided with knowledge of the reward probabilities associated with each block and the probability of block reversal on each trial. The 5% block reversal probability on each trial can be written in terms of the block state transition probabilities as

$$T_{ij} = P\left(s(t) = s_j | s(t-1) = s_i\right) = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix} \tag{2.26}$$

where T_{ij} is defined as the transition probability between block state s_i on trial t and block state s_j on trial $t+1$. Here, 'block state' refers to whether the current block has a higher probability of left or right choices being rewarded. The reward probabilities for each block were as follows

$$R_{ik} = P\left(r(t) = 1 | s(t) = s_i, c(t) = c_k\right) = \begin{cases} \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}, & \text{Bromberg-Martin Task} \\ \begin{bmatrix} 0.7 & 0.1 \\ 0.1 & 0.7 \end{bmatrix}, & \text{Our Task} \end{cases} \tag{2.27}$$

where R_{ik} is defined as the probability of reward for block state s_i and choice c_k.

On each trial, the ideal observer model selects the choice with the highest expectation of reward based on its belief about the current block state given the choice and reward history. The expectation of reward $\rho_l(t+1)$ for choice l on trial $t+1$, given the entire reward history $r(1:t)$ and choice history $c(1:t)$ up until trial t is given by

$$\begin{aligned}\rho_l(t+1) &= \sum_{i=1}^{2} R_{il} P(s(t+1)=s_i|r(1:t),c(1:t)) \\ &= \sum_{i=1}^{2}\sum_{j=1}^{2} R_{il} P(s(t+1)=s_i|s(t)=s_j) P(s(t)=s_j|r(1:t),c(1:t)) \\ &= \sum_{i=1}^{2}\sum_{j=1}^{2} R_{il} T_{ji} P(s(t)=s_j|r(1:t),c(1:t))\end{aligned} \tag{2.28}$$

where l can be either 1 (left choice) or 2 (right choice) and $P\left(s(t)=s_j|r(1:t),c(1:t)\right)$ is the probability of being in block state s_j on trial t given the entire reward and choice history up to and including trial t. Equation 2.28 tells us that estimating the block state probability $P\left(s(t)=s_j|r(1:t),c(1:t)\right)$ will provide us with an estimate of the expected reward for a given choice on trial $t+1$ as R_{il} and T_{ji} are already known. Using Bayes' theorem, we can estimate the block state probability as

$$P\left(s(t)=s_j|r(1:t),c(1:t)\right) = \frac{P\left(r(t)|r(1:t-1),c(1:t),s(t)=s_j\right) \times P\left(s(t)=s_j|r(1:t-1),c(1:t)\right)}{\sum_{j=1}^{2} P\left(r(t)|r(1:t-1),c(1:t),s(t)=s_j\right) \times P\left(s(t)=s_j|r(1:t-1),c(1:t)\right)} \tag{2.29}$$

The first term in the numerator of the right hand side of Equation 2.29, $P\left(r(t)|r(1:t-1),c(1:t),s(t)=s_j\right)$, is the probability of receiving reward $r(t)$ (1 if rewarded and 0 if unrewarded) on trial t given the current choice $c(t)=c_k$, the block state s_j, and the reward history $r(1:t-1)$ and the choice history $c(1:t-1)$ up to trial $t-1$. Because the past history of rewards and choices does not affect the reward probability once the block state is known, this can be rewritten as

$$\begin{aligned}P\left(r(t)|r(1:t-1),c(1:t),s(t)=s_j\right) &= P\left(r(t)|c(t)=c_k,s(t)=s_j\right) \\ &= R_{jk}^{r(t)}(1-R_{jk})^{1-r(t)}\end{aligned} \tag{2.30}$$

The second term in the numerator of the right hand side of Equation 2.29, $P\left(s(t)=s_j|r(1:t-1),c(1:t)\right)$, is the probability that the current block state is $P\left(s(t)=s_j|r(1:t-1),c(1:t)\right)$ given the reward and choice history. This can be rewritten as

$$\begin{aligned}
&P\left(s(t)=s_j|r(1:t-1),c(1:t)\right)\\
&=P\left(s(t)=s_j|r(1:t-1),c(1:t-1)\right)\\
&=\sum_{m=1}^{2}P\left(s(t)=s_j|s(t-1)=s_m\right)\times P\left(s(t-1)=s_m|r(1:t-1),c(1:t-1)\right)\\
&=\sum_{m=1}^{2}T_{mj}P\left(s(t-1)=s_m|r(1:t-1),c(1:t-1)\right)
\end{aligned}\tag{2.31}$$

In the second line above, the dependence on c(t) has been removed because the choice on the current trial, in the absence of reward information on the current trial, does not provide any additional information about the current state beyond that provided by the past reward and choice history. Combining Equations 2.29-2.31, the block state probability on the current trial t can be written in terms of the known reward probabilities, known state transition probabilities and the previous block state probability as

$$P\left(s(t)=s_j|r(1:t),c(1:t)\right)=\frac{\sum_{m=1}^{2}R_{jk}^{r(t)}(1-R_{jk})^{1-r(t)}T_{mj}P\left(s(t-1)=s_m|r(1:t-1),c(1:t-1)\right)}{\sum_{j=1}^{2}\sum_{m=1}^{2}R_{jk}^{r(t)}(1-R_{jk})^{1-r(t)}T_{mj}P\left(s(t-1)=s_m|r(1:t-1),c(1:t-1)\right)}\tag{2.32}$$

The above equation allows us to estimate the current trial block state probability $P\left(s(t)=s_j|r(1:t),c(1:t)\right)$ recursively, since it can be expressed in terms of the previous trial block state probability $P\left(s(t-1)=s_m|r(1:t-1\right.$ and other known constant terms. This combined with the known reward and block transition probabilities allows the model to select the optimal choice according to Equation 2.28.

Q-Learning model To simulate trial-by-trial, model-free performance of the tasks, we used a Q-learning model in which the value of the chosen action is updated on each trial as follows:

$$\begin{gathered}
Q_{right}(t+1)=\begin{cases}Q_{right}(t)+\alpha(r(t)-Q_{right}(t)), & \text{if } c(t)=right\\ Q_{right}(t), & \text{if } c(t)=left\end{cases}\\
Q_{left}(t+1)=\begin{cases}Q_{left}(t), & \text{if } c(t)=right\\ Q_{left}(t)+\alpha(r(t)-Q_{left}(t)), & \text{if } c(t)=left\end{cases}
\end{gathered}\tag{2.33}$$

where Q_{right} is the value for the right choice and Q_{left} is the value for the left choice. t is the current trial and α is the learning rate, which was set to 0.612 per trial. $r(t)$ is the outcome of trial t (1 for reward, 0 for no reward). Q-values for each choice were initialized to 0. The outcome $r(t)$ was determined based on the reward probability for choice $c(t)$ given the block. Choice was simulated using a softmax equation such that the probability of choosing right or left is given by,

$$\begin{aligned} P(c(t) = right) &= \frac{exp(\beta_{value} Q_{right}(t) + \beta_{stay} I_{right}(t))}{exp(\beta_{value} Q_{left}(t) + \beta_{stay} I_{left}(t)) + exp(\beta_{value} Q_{right}(t) + \beta_{stay} I_{right}(t))} \\ P(c(t) = left) &= \frac{exp(\beta_{value} Q_{left}(t) + \beta_{stay} I_{left}(t))}{exp(\beta_{value} Q_{left}(t) + \beta_{stay} I_{left}(t)) + exp(\beta_{value} Q_{right}(t) + \beta_{stay} I_{right}(t))} \end{aligned} \tag{2.34}$$

where β_{value} is the inverse temperature parameter, which was set to 0.99. β_{stay} is a parameter accounting for how likely mice were to repeat their previous choice, which was set to 0.95. $I_{left/right}$ is 1 if that action (i.e. left or right) was chosen on the previous trial and 0 otherwise. Parameters for the Q-learning model were fit in [182] to the behavior of mice in which dopamine neuron activity was recorded in [26].

Comparison of RPE at block reversals RPE for both the ideal-observer model and the Q-learning model (Supplementary Figure 2.21) was defined as the difference between the experienced reward $r(t)$ and the expected reward for the chosen action ($\rho_{chosen}(t)$ for the ideal-observer model or $Q_{chosen}(t)$ for the Q-learning model) as follows:

$$\begin{aligned} RPE_{IdealObserver} &= r(t) - \rho_{chosen}(t) \\ RPE_{Q-learning} &= r(t) - Q_{chosen}(t) \end{aligned} \tag{2.35}$$

To identify RPE signatures of model free versus model based performance of the two tasks, we compared the RPE from the ideal-observer model and the Q-learning model on trials around block reversals. Specifically, we compared the RPE from the two models on the first trial of a block with the RPE on the second trial of a block when the choice on trial 1 was different from the choice on trial 2. This means that any changes in RPE from trial 1 to trial 2 were inferred because the new action-outcome relationship for the choice made on trial 2 had not been explicitly experienced in the new block.

2.5. Supplementary Materials

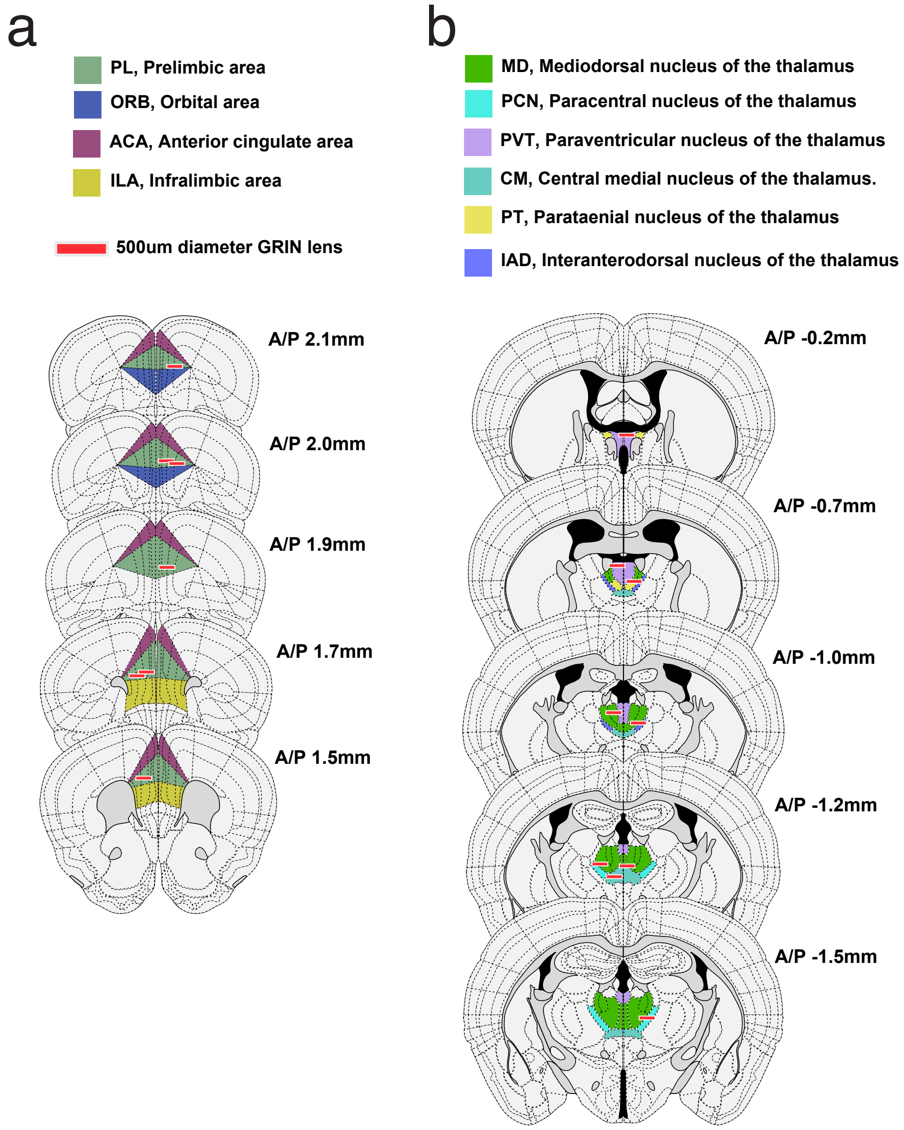

a
PL, Prelimbic area
ORB, Orbital area
ACA, Anterior cingulate area
ILA, Infralimbic area
500um diameter GRIN lens
A/P 2.1mm
A/P 2.0mm
A/P 1.9mm
A/P 1.7mm
A/P 1.5mm
b
MD, Mediodorsal nucleus of the thalamus
PCN, Paracentral nucleus of the thalamus
PVT, Paraventricular nucleus of the thalamus
CM, Central medial nucleus of the thalamus.
PT, Parataenial nucleus of the thalamus
IAD, Interanterodorsal nucleus of the thalamus
A/P -0.2mm
A/P -0.7mm
A/P -1.0mm
A/P -1.2mm
A/P -1.5mm

Figure 2.8. **Locations of GRIN lens implants. Related to methods.** (a) Schematic of coronal sections along the anterior/posterior axis (A/P, numbers relative to bregma) with recording locations of 7 PL-NAc mice. Red lines indicate bottom of lens implant. (b) Same as a except location of 9 mTH-NAc recordings.

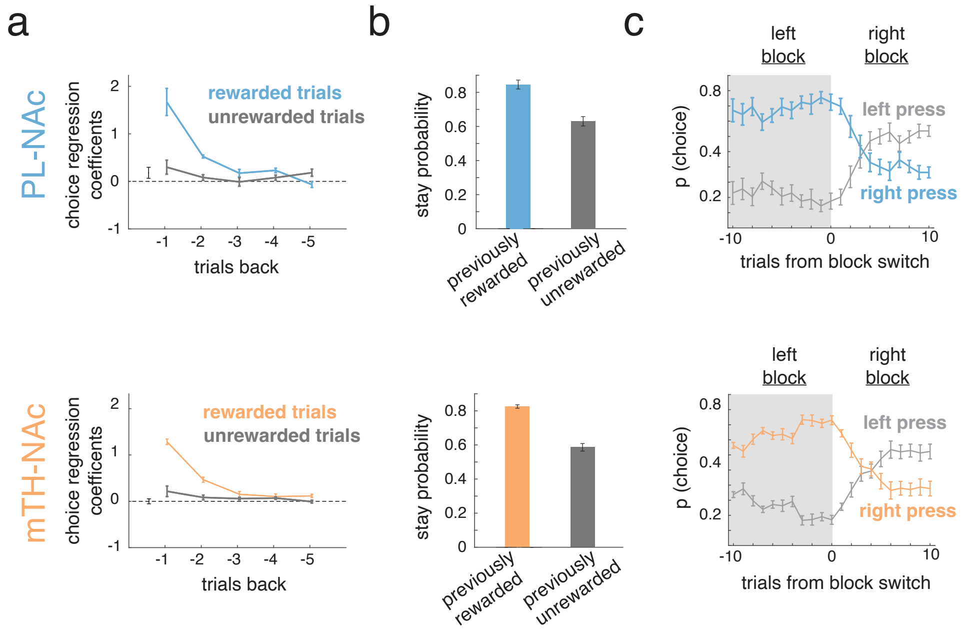


Figure 2.9. **Mice in the PL-NAc and mTH-NAc imaging cohorts have comparable behavior. Related to Figure 2.1 and methods.** (a) Top, coefficients from logistic regression to predict choice (see Figure 2.1) from PL-NAc imaging cohort (n=7 mice). Bottom, same except coefficients from mTH-NAc imaging cohort (n=9 mice). Both cohorts use choice and outcome information from previous trials to predict the current choice. Regression coefficients between the two cohorts are not significantly different for any trials back for either rewarded or unrewarded trials (P>0.01, unpaired, two-tailed t-test of regression coefficients across mice at each trial back, n=7 and 9 mice for PL-NAc and mTH-NAc, respectively). (b) Stay probability following rewarded (blue or orange) and unrewarded (grey) trials for PL-NAc (top) and mTH-NAc (bottom) cohorts. Both cohorts have a significantly higher stay probability following a rewarded trial (PL-NAc: P=0.00008; mTH-NAc: P=0.00003, paired, two-tailed t-test comparing stay probability on rewarded and unrewarded trials across mice, n=7 and 9 mice for PL-NAc and mTH-NAc, respectively). (c) Probability of a left or right lever press following a reversal from a left-preferring to right-preferring block of mice from the PL-NAc (top, n=7 mice) and mTH-NAc (bottom, n=9 mice) cohorts. Both cohorts display a qualitatively similar change in choice behavior following a block reversal. In all panels, data are represented as mean ± SEM across mice.

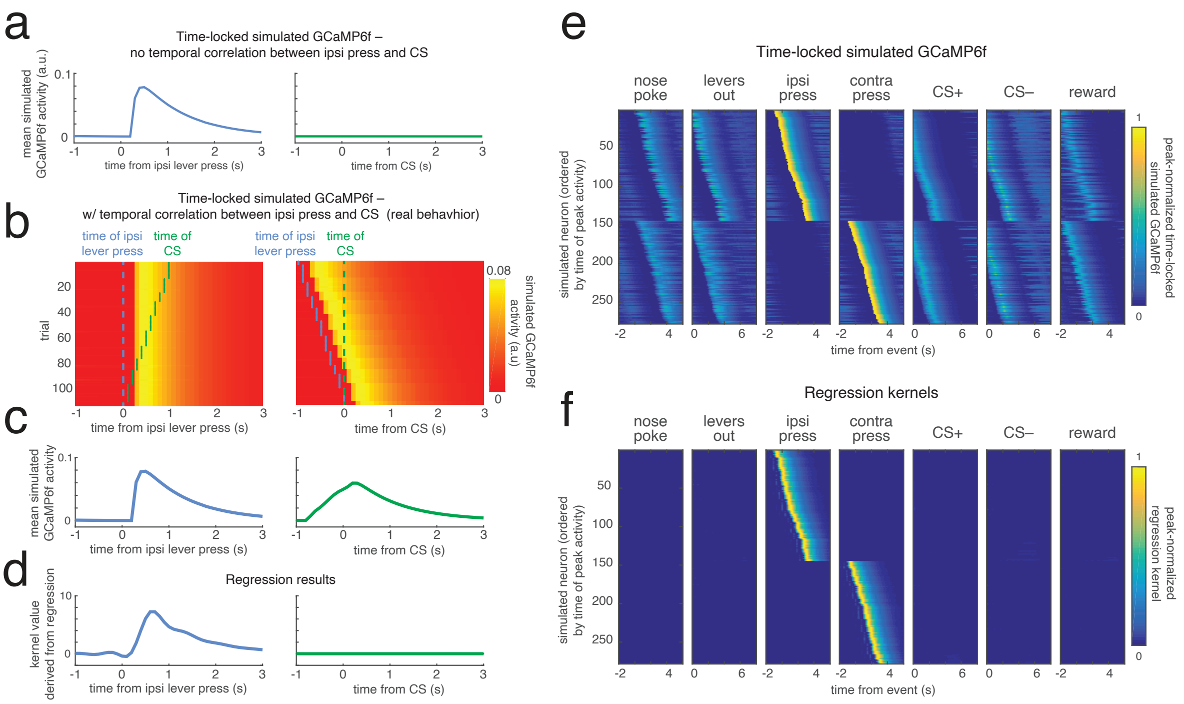


Figure 2.10. **Simulated neural activity to illustrate the ability of the encoding model to successfully relate neural activity to the appropriate behavioral event. Related to Figure 2.2 and methods.** (a) Simulated neuron that is responsive only to the ipsilateral lever press. (b) Trial-by-trial heatmap of a simulated neuron that has increased activity time-locked to the ipsilateral lever press. In the data, there was a correlation between the time of lever press and the time of the CS, which produced a time-locked response to the CS, even though the neuron did not respond to that event. Left, activity heatmap aligned to the time of an ipsilateral lever press (dashed blue line) sorted by the time of the subsequent CS presentation (green dots). Right, activity heatmap is aligned to the time of the CS (dashed green line),ordered by the time of the preceding lever press (blue dots). (c) Average activity across trials of the example simulated neuron in b aligned to the lever press, left, and CS presentation, right. Unlike the idealized case in a, when the timing of task events is maintained from the real behavior, the temporal correlations result in a bump in activity aligned to the CS (right plot). Note that this bump in activity is generated entirely by the correlation in event times, since this simulated neuron only had activity in relation to the lever press (and not the CS presentation). (d) Response kernels for lever press, left, and CS, right, derived from the encoding model used to attribute the neural response of individual task events. The model successfully

recovers the fact that neural activity in this simulated neuron is related to the lever press and not the CS. (e) Heatmap displaying the average activity from a population of 278 simulated neurons that respond to either the ipsilateral or contralateral lever press, but not the other events. Each neuron responds to the lever press, with a randomly assigned response latency from -1 to 3s. While the strongest average time-locked response is to the ipsilateral or contralateral lever presses, there are visible responses to the other task events as a consequence of the correlation between task events resulting from their temporal proximity. (f) Same as e except heatmap displays the response kernels derived from the encoding model. The model successfully discovers the underlying structure of the data (i.e., that responses are driven by the lever press).

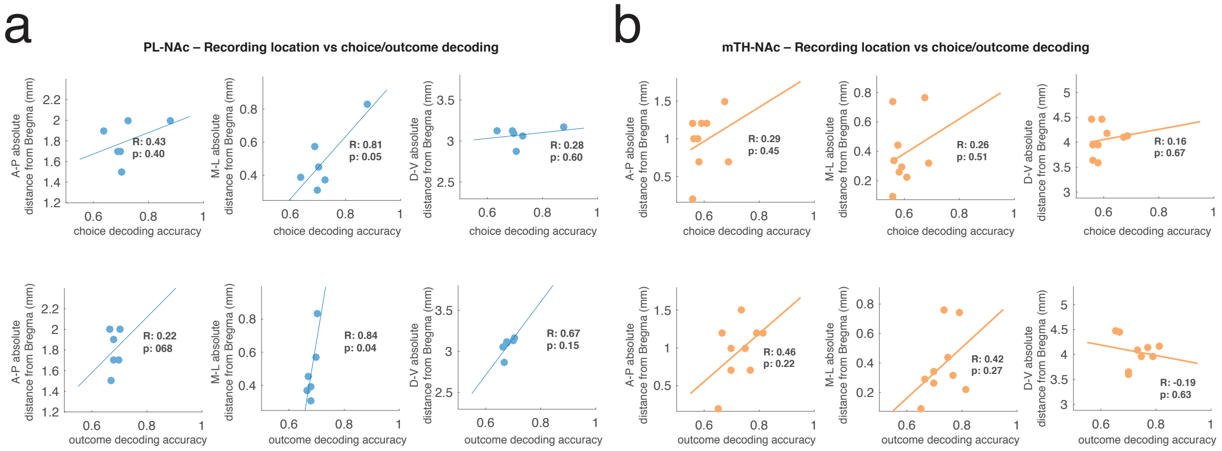


Figure 2.11. **Lack of correlation between recording locations relative to Bregma and choice / outcome decoding. Related to Figure 2.3.** (a) Top, correlations between choice decoding accuracy using recorded PL-NAc activity and, in order, the anterior/posterior (A/P), medial/lateral (M/L) and dorsal/ventral (D/V) recording locations relative to Bregma (see Supplemental Figure 2.1 for schematic of recording locations; recording locations were aligned to the Allen atlas using the Wholebrain software suite (http://www.wholebrainsoftware.org/) of [165]; see Methods for details, n=6 mice). Bottom, same as top except correlation between recording location and outcome decoding accuracy using PL-NAc activity. (b) Same as a except decoding accuracy for choice (top) and outcome (bottom) determined using recorded mTH-NAc activity (n=9 mice). All p-values are calculated from Pearson's correlation coefficient; none are significant at the P<.05 level after correction for multiple (6) hypotheses using Bonferroni correction.

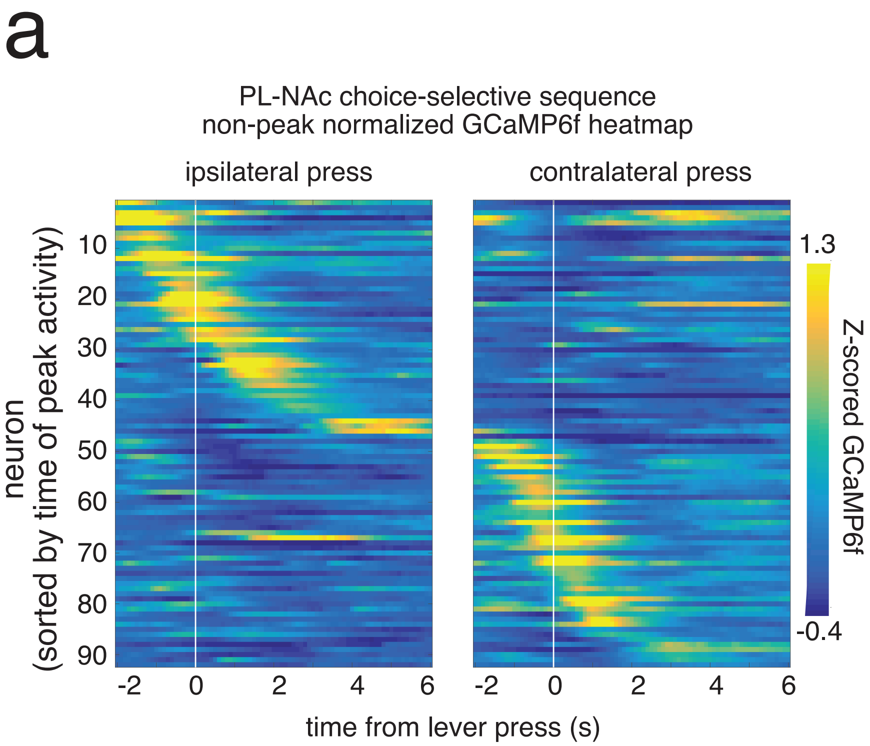


Figure 2.12. **Choice-selective sequences in PL-NAc neurons without peak-normalization. Related to Figure 2.4.** Heatmap demonstrating sequential response of choice-selective PL-NAc neurons to the ipsilateral and contralateral lever press (n=92 neurons from 7 mice). Similar to Figure 2.4b-c, but time-locked, trial-averaged GCaMP6f fluorescence is not normalized by the peak response to the lever press and is taken from all trials.

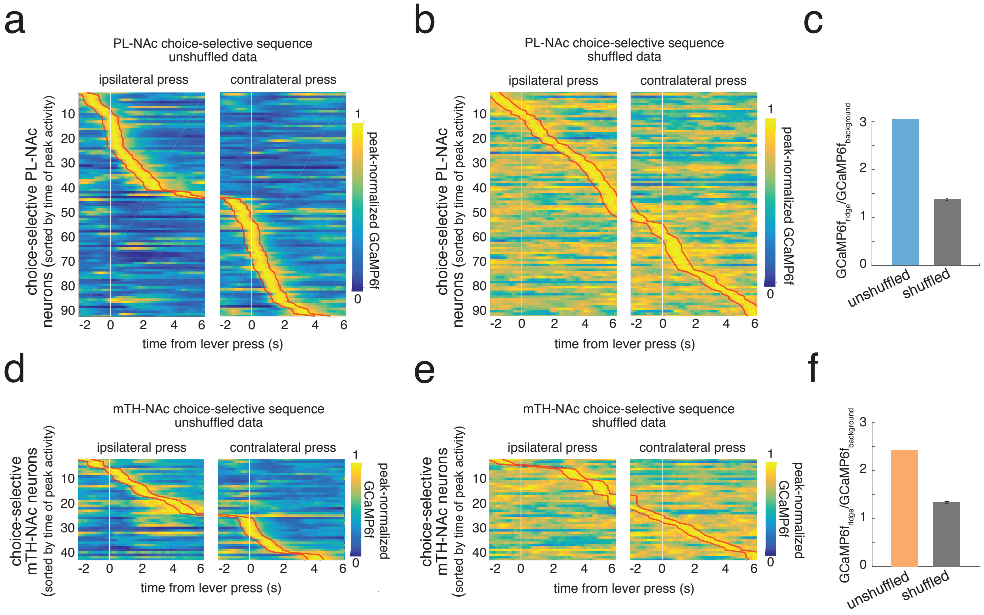


Figure 2.13. **The calculated ridge-to-background ratio of PL-NAc neurons supports the presence of sequences. Related to Figure 2.4.** (a) Sequential activity of PL-NAc choice-selective neurons. Similar to Figure 2.4b-c, the heatmap is ordered by the time of peak activity time-locked to the ipsilateral (left column) and contralateral (right column) lever press of each neuron, but instead of cross-validation, activity is averaged across all trials. Red trace represents the borders of the one-second window around the peak defined as the 'ridge'. Activity at all other surrounding timepoints is considered the 'background'. (b) Same as a for data that is shuffled by temporally shifting the GCaMP6f fluorescence trace across a recording session separately for each neuron by a random number of frames, chosen from a uniform distribution. Ordering by the time of peak activity generates spurious sequential activity across the diagonal in shuffled data. (c) Calculated ridge-to-background ratio of PL-NAc neurons using unshuffled (blue) and shuffled (grey) data. A ratio is calculated for each individual neuron and the average of these ratios across all neurons displayed in the heatmap is shown. The ratio calculated from unshuffled data is significantly larger than that from the shuffled data (P<0.0001, comparison between unshuffled data

and distribution of 500 shuffled iterations). Error bars for shuffled data indicate one standard deviation. (d-f) Same as a-c but ridge-to-background is calculated using mTH-NAc neural recordings. Similar to PL-NAc, the ratio calculated from unshuffled data was significantly larger than that from the shuffled data (P<0.0001). However, when comparing across the populations, the ridge-to-background calculated using PL-NAc neurons (3.06±0.12, mean±sem, n=92 neurons from 7 mice) was significantly larger than that using mTH-NAc (2.42±0.12, mean±sem, n=42 neurons from 9 mice; P=0.004: unpaired, two-tailed t-test comparing ratio between PL-NAc and mTH-NAc neurons).

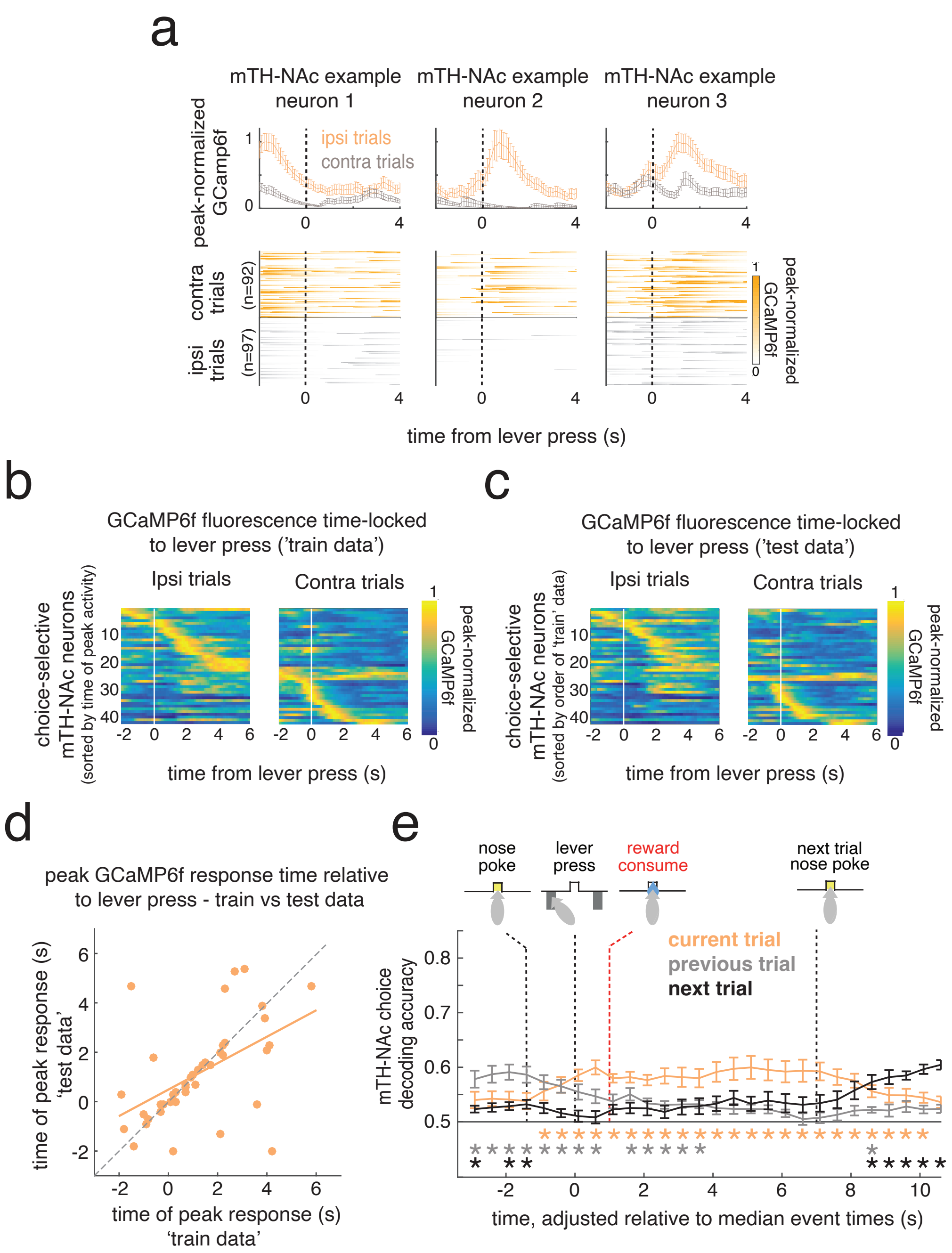


Figure 2.14. **mTH-NAc choice-selective neurons display sequential activity that is less consistent than PL-NAc. Related to Figure 2.4.** (a) Top; average GCaMP6f fluorescence of three simultaneously imaged mTH-NAc choice-selective neurons with different response times relative to the lever press. Error bars are s.e.m across trials. Bottom, heatmaps of GCaMP6f fluorescence response across trials

to ipsilateral (orange) and contralateral (grey) lever presses. (b,c) Heatmaps of choice-selective mTH-NAc neurons' peak-normalized GCaMP6f responses to lever press (n=42/256 neurons from 9 mice). Each row is the average GCaMP6f fluorescence time-locked to the ipsilateral (left column) and contralateral (right column) lever press for a neuron, normalized by the neuron's peak average fluorescence. In b ('train data'), heatmap is generated using a randomly selected half of trials and ordered by the time of each neuron's peak activity. In c ('test data'), the peak-normalized, time-locked GCaMP6f fluorescence from the other half of trials was used while maintaining the order from 'train data' in b. Compare to PL-NAc data in Figure 2.4b-c. (d) Correlation between the time of peak activity using the 'train' (horizontal axis) and 'test' (vertical axis) trials for choice-selective mTH-NAc neurons. While mTH-NAc choice-selective neurons also show significant correlation between 'train' and 'test' trials ($R^2 = 0.51$, P $= 5.5 \times 10^{-4}$, n=42 neurons from 9 mice), this correlation is significantly lower than that of PL-NAc (comparison with data in Figure 2.4d; P=0.005, Z=2.81, Fisher's R-to-Z transformation, comparison of correlation coefficients derived from comparing peak activity between 'test' and 'training' data from PL-NAc versus mTH-NAc). (e) Average choice decoding accuracy of the mice's choice on the current (orange), previous (grey) and next trial (black) as a function of GCaMP6f fluorescence throughout the current trial. GCaMP6f fluorescence is taken from 100 random selections per mouse of 10 simultaneously imaged mTH-NAc neurons (each trial's activity is adjusted in a piecewise linear manner relative to the median time of the nose poke, lever press and next trial nose poke, see Methods for details). Data are represented as mean ± SEM across mice (n=9 mice). Red dashed line indicates median onset of reward consumption. * indicate significant decoding accuracy above chance, P<0.01, two-tailed, one-sample t-test across mice.

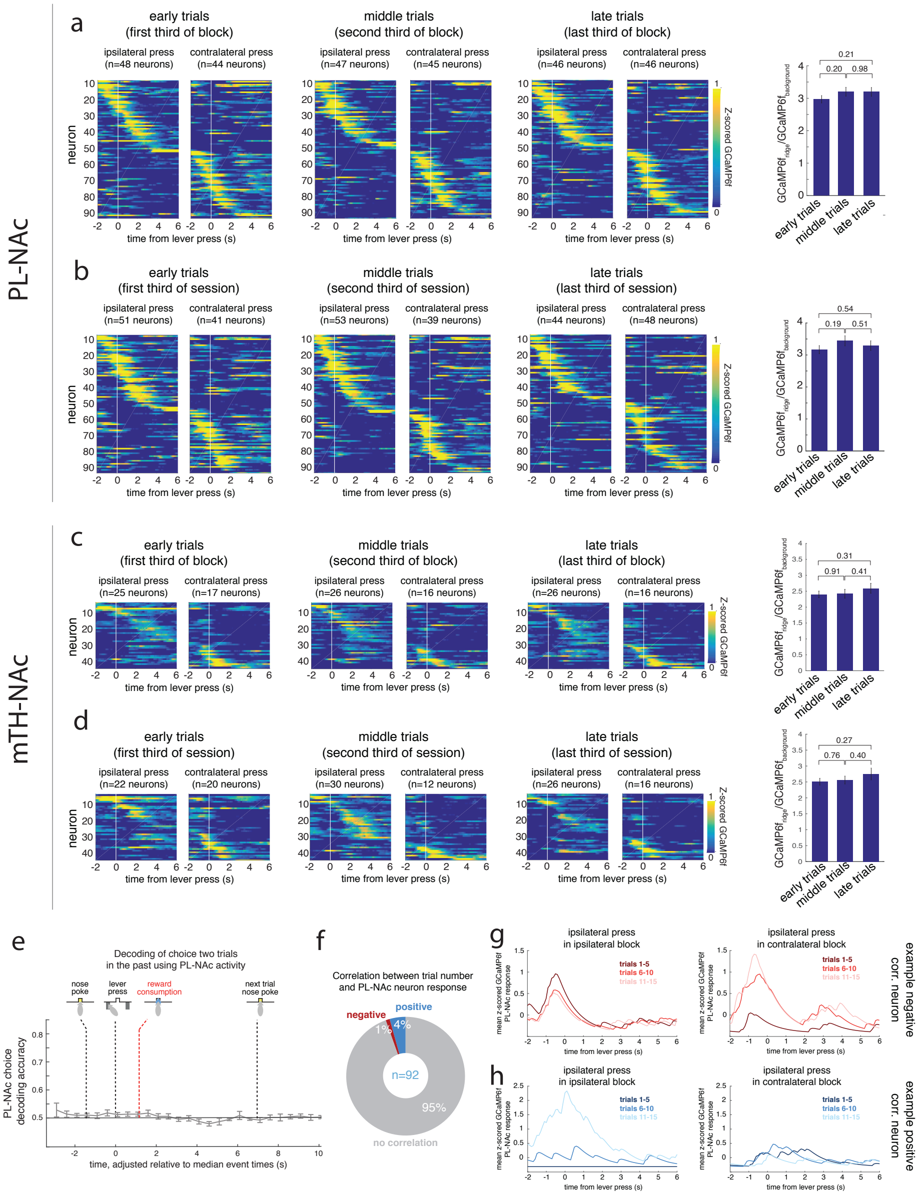


Figure 2.15. **Trial number within a block, and two-trial-back choice, are not strongly encoded by PL-NAc activity. Related to Figure 2.4.** (a) Left, heatmaps of Z-scored GCaMP6f activity from 92 choice-selective PL-NAc neurons averaged across the first, middle and last third of trials of each block. Right,

calculated ridge-to-background ratio derived from average activity from each third of trials within a block. Data are represented as mean ± SEM across neurons (n=92 from 7 mice). No significant changes are observed in the ratios calculated across a block, suggesting that the strength of sequences is not modulated by block trial number (P>0.05, paired, two-tailed t-test). (b) Same as a except data is split into the first, middle and final third of the entire recording session. (c,d) Same as a,b except activity is of mTH-NAc choice-selective neurons. (e) Decoding accuracy for the mice's choice two trials back using activity from 10 simultaneously recorded PL-NAc neurons. Unlike choice decoding on the current and previous trial (Figure 2.4e; blue and black traces, respectively), PL-NAc activity is not able to accurately decode choice from two trials back after correcting for cross-trial choice correlations (see Methods for details) at any time point in the trial (P>0.05 for all time points: one-sample, two-tailed t-test across mice comparing decoding accuracy with chance rate of 0.5). (f) Proportion of PL-NAc choice-selective neurons whose activity is significantly positively (blue, n=4 neurons) or negatively (red, n=1 neuron) correlated with the number of trials into a block (P<0.01). Significance was determined by comparing the calculated correlation coefficient of each neuron to a null distribution of 500 correlation coefficients generated using GCaMP6f signal circularly shifted by a random integer, to control for slow drift in the data. R-values were calculated using the maximum GCaMP6f activity from 2s before to 6s after the time of lever press for the first 15 trials in a block. Using the same criteria, no mTH-NAc neurons were significantly correlated, either positively or negatively with trial number in a block (P>0.05). (g) Left, average activity of the negatively correlated PL-NAc neuron in response to an ipsilateral lever press at various trials in an ipsilateral block, where the ipsilateral lever had a higher probability of reward and, thus, the value of the ipsilateral lever increases as a function of trial number. Right, average activity of the negatively correlated PL-NAc neuron in response to an ipsilateral press in a contralateral block. (h) Same as g except for an example of a positively correlated PL-NAc neuron.

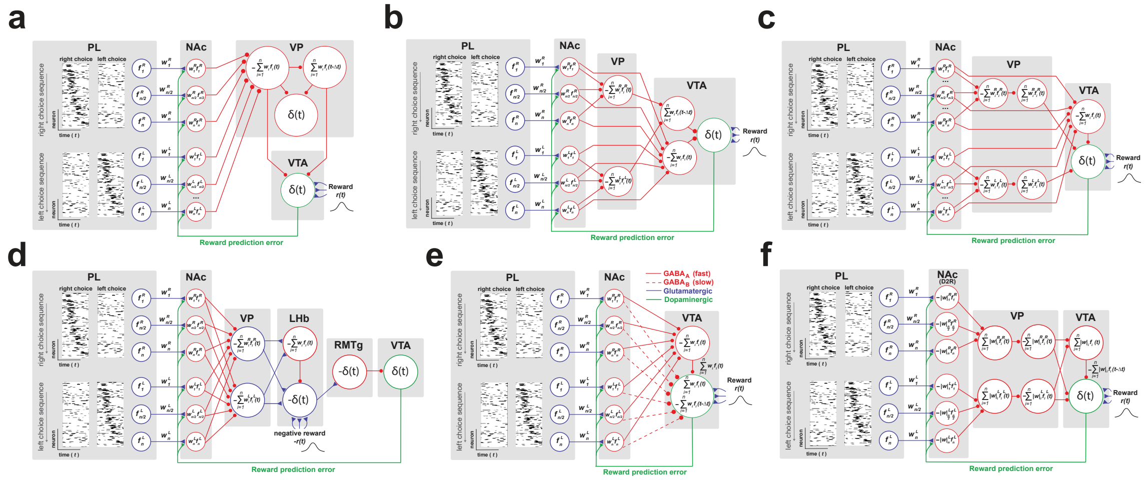


Figure 2.16. **Alternative model architectures used to implement synaptic plasticity model. Related to Figure 2.5.** (a-f) Alternative models constructed using known circuit architecture. All models except a,d generate an RPE signal by providing a 'fast excitatory' and 'slow-inhibitory' value pathway to the VTA dopamine neuron population. Note that all model variants rely on choice-selective sequences in PL-NAc to bridge the beginning of the action sequence and outcomes across time. For all model variants, GABAergic, glutamatergic and dopaminergic projections are denoted as red, blue and green, respectively. Brain region abbreviations are: prelimbic cortex, PL; nucleus accumbens, NAc; ventral pallidum, VP; ventral tegmental area, VTA; lateral habenula, LHb; rostromedial tegmental nucleus, RMTg. (a) In this model, the delay and inversion of the value signal is accomplished through a second VP neuron. These two VP neurons converge onto a third VP interneuron to generate an RPE signal in VP, as has been observed by [183]. (b) In this model, the fast excitatory pathway is generated via a direct projection of NAc neurons onto the VTA GABA neuron while the slow inhibitory pathway passes through the VP before synapsing onto a VTA GABA neuron. (c) Similar to b except that the slow inhibitory pathway contains an additional VP neuron, which accomplishes the sign inversion and delay assigned to a VTA GABA neuron in b. Since the models in b prescribe a role for the observed NAc-D1R projections to VTA GABA neurons, they produce negative value signals in VTA GABA neurons, whereas only positive value signals have been observed experimentally in identified GABA interneurons in the VTA [99]. (d) In this model, a negative reward prediction error is calculated in the LHb using glutamatergic projections from the VP [184], inversion and delay in the value signal from local inhibitory LHb neurons to produce an inverted RPE, which is then transmitted to the

VTA via the RMTg [54, 185]. (e) To account for previous work describing direct projections from NAc D1R neurons to the VTA [164, 170, 186], this alternative model architecture has NAc neurons projecting directly to the VTA, skipping the VP. In this model, the timing difference needed to compute an RPE signal is generated through the activity of fast ionotropic GABA-A receptors (solid red trace), which have been shown to preferentially express in NAc-VTA GABA interneuron projection postsynaptic densities [187], while activity of metabotropic GABA-B receptors (dashed red trace), which are preferentially expressed in the postsynaptic densities of NAc-VTA DA projections [187], generate the slow-inhibitory pathway. Notably, without this differential expression of GABA receptors in the DAergic and GABAergic populations of the VTA, this model architecture would fail to produce an RPE signal, as it would instead generate a fast-inhibitory and slow-excitatory signal in the VTA DA neuron population. (f) Multiple studies have implicated D2-R expressing MSNs as playing a critical role in reversal learning in multiple mammalian species [188, 189, 190, 191, 192, 193]. Thus, in this model we account for the possibility that the reversal behavior in our task is mediated specifically by changes to synaptic weights from PL to D2-R-expressing NAc MSNs. This model assumes the opposite dopamine-mediated plasticity rule (LTD rather than LTP) than the previous models.

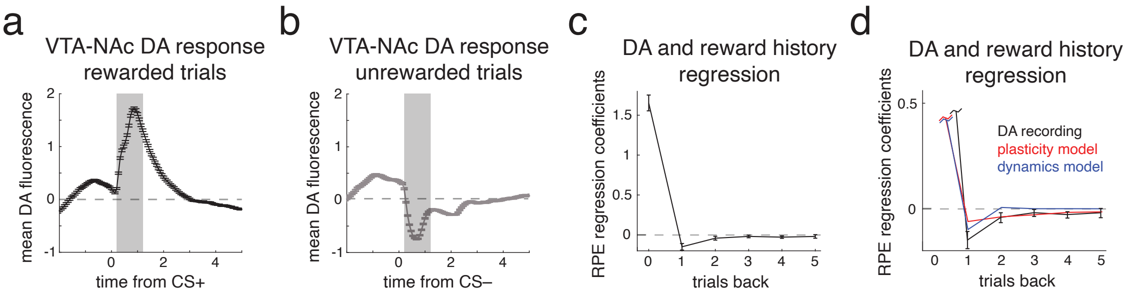


Figure 2.17. **Reward prediction error (RPE) encoding observed in recorded dopamine (DA) activity is similar to that produced by our synaptic plasticity and neural dynamics models. Related to Figures 5 and 6.** (a) Mean bulk GCaMP6f fluorescence from VTA-NAc DA terminals in response to a conditioned stimulus signaling reward (CS+, data taken from ([26], n=11 recording sites). Note that terminal fluorescence recordings are presented here to more accurately reflect the signal that downstream NAc neurons are receiving in our model. (b) Same as a except DA fluorescence in response to the conditioned stimulus signaling an unrewarded trial (CS–). (c) Coefficients from a multiple linear regression in which outcome is predicted using mean DA fluorescence signals from 0.2-1.2s relative to the time of CS presentation across current ("0") and multiple previous trials (see shaded region in a,b), similar to Figure 2.5g,l,q and Figure 2.6e. The positive coefficient for the current trial and negative coefficients for previous trials indicate the encoding of an RPE. (d) Same as c, but also including coefficients from the synaptic plasticity model (red, same coefficients as Figure 2.5g) and the neural dynamics model (black, same coefficients as Figure 2.6e), to allow direct comparison. Error bars in all panels represent s.e.m across 11 recording sites.

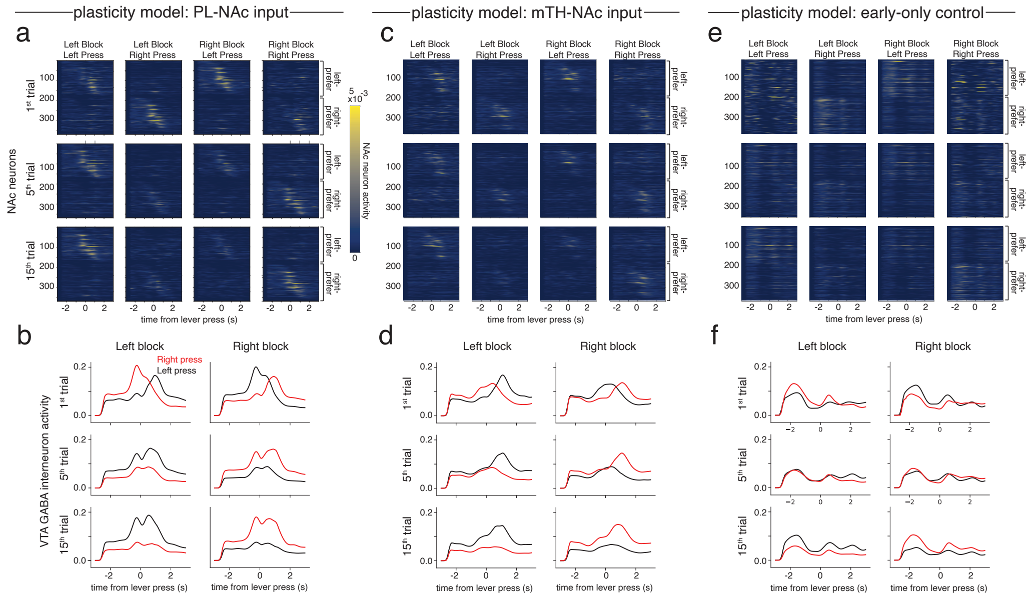
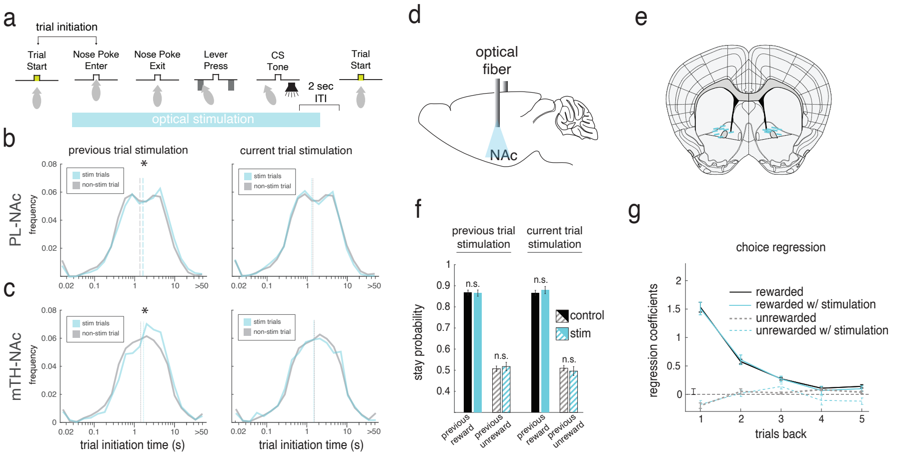


Figure 2.18. **Synaptic plasticity model using sequential PL-NAc but not early-only or mTH-NAc activity correctly modulates activity in NAc projection neurons and VTA GABA interneurons. Related to Figure 2.5.** (a) Heatmaps of average activity relative to the time of the lever press for NAc projection neurons in the PL-NAc model (Figure 2.5c). Top, middle and bottom heatmaps are the average activity across the first, fifth and fifteenth trial of each block, respectively. Each column is the average activity across trials from different block/press combinations. For each subplot, neurons 1-184 are left-preferring and neurons 185-368 are right-preferring. The activity of these left- and right-preferring NAc neurons increases throughout a block of their respective lever preference. In contrast, their activity decreases throughout a block opposite to their lever preference. (b) Average activity of VTA GABA interneuron from synaptic plasticity model using PL-NAc activity as input on left (black) or right (red) trials. Activity is relative to the time of the lever press across the first, fifth and fifteenth trials of a left-preferring (left column) or right-preferring (right column) block. Similar to a, throughout a left block (left column), the activity on left press trials increases from the first to fifteenth trial while the activity on right press trials decreases. The opposite pattern is seen for left and right press trials throughout a right block (right column). (c,d) Same as a,b except NAc and VTA GABA interneuron generated using mTH-NAc as input to the synaptic

plasticity model. (e,f) Same as a,b except NAc and VTA GABA interneuron from the early-only control synaptic plasticity model.



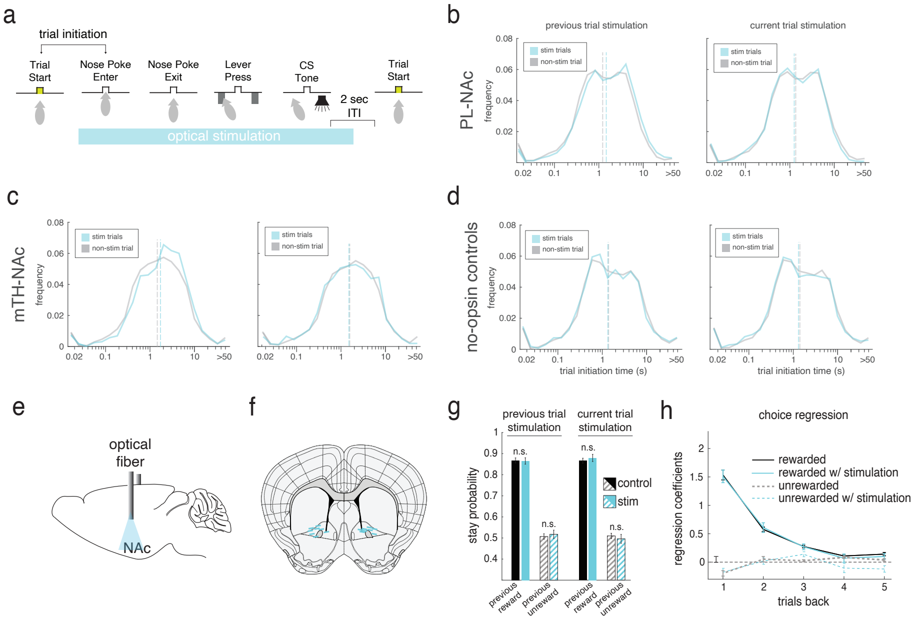


Figure 2.19. **Laser stimulation affects trial initiation times in both PL-NAc and mTH-NAc but does not affect behavior in control mice that do not express opsin. Related to Figure 2.7.** (a) Schematic of trial structure and time of optical stimulation. “trial initiation” time is defined as the latency between the start of the trial and the mouse entering the central nose poke. (b) Left, distribution of trial

initiation times following stimulation trials (blue) and non-stimulation trials (grey) in PL-NAc ChR2 mice. Blue and grey vertical lines indicate median initiation times for stim and non-stim trials, respectively. Right, same only effect of current trial stimulation on trial initiation times. Previous trial stimulation resulted in significantly longer trial initiations in PL-NAc ChR2 mice than no-opsin control mice (P=3.46x10-7, p-value from the 'opsin group X previous trial stimulation' interaction term of a mixed effects model used to predict latency times of PL-NAc and control mice, fit using the fitglme function in MATLAB; see Methods for additional model details). In contrast, current trial stimulation had no significant effect on initiation times (P=0.95, same test as above except p-value is that of the interaction term of 'opsin group X current trial stimulation'), an expected result as the start of stimulation was contingent on the mouse performing a nose poke. (c) mTH-NAc ChR2 mice had significantly longer trial initiation times following optical stimulation than no-opsin control mice ($P=2.34 \times 10^{-4}$, p-value from the 'opsin group X previous trial stimulation' interaction term of a mixed effects model used to predict latency times of mTH-NAc and control mice) but no effect of current trial stimulation was observed (P=0.74, same test as above except the p-value is that from the 'opsin group X current trial stimulation' interaction term). (d) Same as b,c except latencies from no-opsin control cohort. (e) Surgical schematic of no-opsin control cohort. Optical fibers were implanted into the NAc. (f) Optical fiber tip locations of no-opsin control cohort (n=8 mice). (g) Unlike PL-NAc ChR2 expressing mice (Figure 2.7f-h), neither current nor previous trial stimulation changed the stay probability in control mice following rewarded (P=0.52: previous trial stimulation; P=0.24: current trial stimulation; paired t-test) or unrewarded trials (P=0.52: previous trial stimulation; P=0.47: current trial stimulation; paired t-test). (h) Likewise, stimulation from multiple trials back had no effect on choice (P>0.05 for all trials back, t-test across mice's laser x choice interaction term coefficients). Data in g,h are represented as mean ± SEM across mice, n=8.

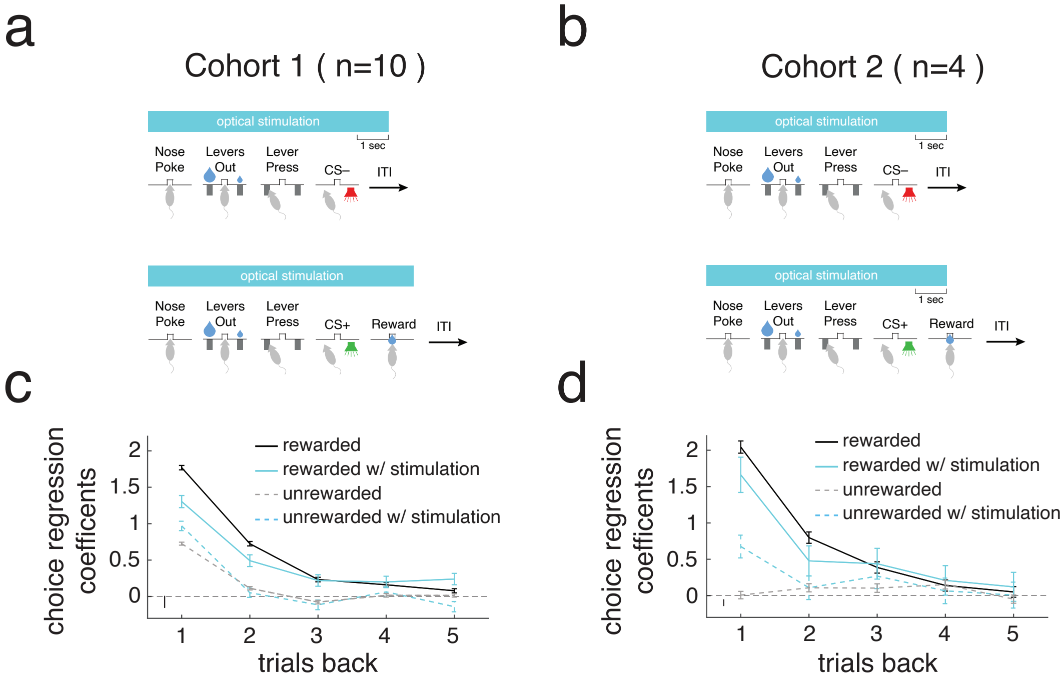


Figure 2.20. **Effect of PL-NAc optogenetic stimulation in two cohorts. Related to Figure 2.7.** (a) Schematic of optical stimulation parameters for cohort 1. On 10% of unrewarded trials, optical stimulation began when the mouse entered the central nosepoke and ended 1s into the intertrial interval (ITI), which began at the end of the 500ms CS– tone. On 10% of rewarded trials, stimulation began with nose poke and ended after the mouse left the reward port. (b) Schematic for cohort 2. Unlike cohort 1, optical stimulation ended on the same timescale on both rewarded and unrewarded trials, 1s after the end of CS presentation. (c) Logistic regression model similar to that in Figure 2.1e demonstrating the effect of PL-NAc stimulation on lever choice in cohort 1 mice (n=10 mice, see Methods for model details). Rewarded trials with stimulation one and two trials back decreased stay probability compared with rewarded trials without stimulation. Stimulation had an opposite effect on unrewarded trials, for which there was an increase in stay probability following stimulation one trial back compared to trials without stimulation. (d) Same as c except data from cohort 2 (n=4 mice). Effect of optical stimulation of PL-NAc neurons was qualitatively similar across the two cohorts. Data in c,d are represented as mean ± SEM across mice.

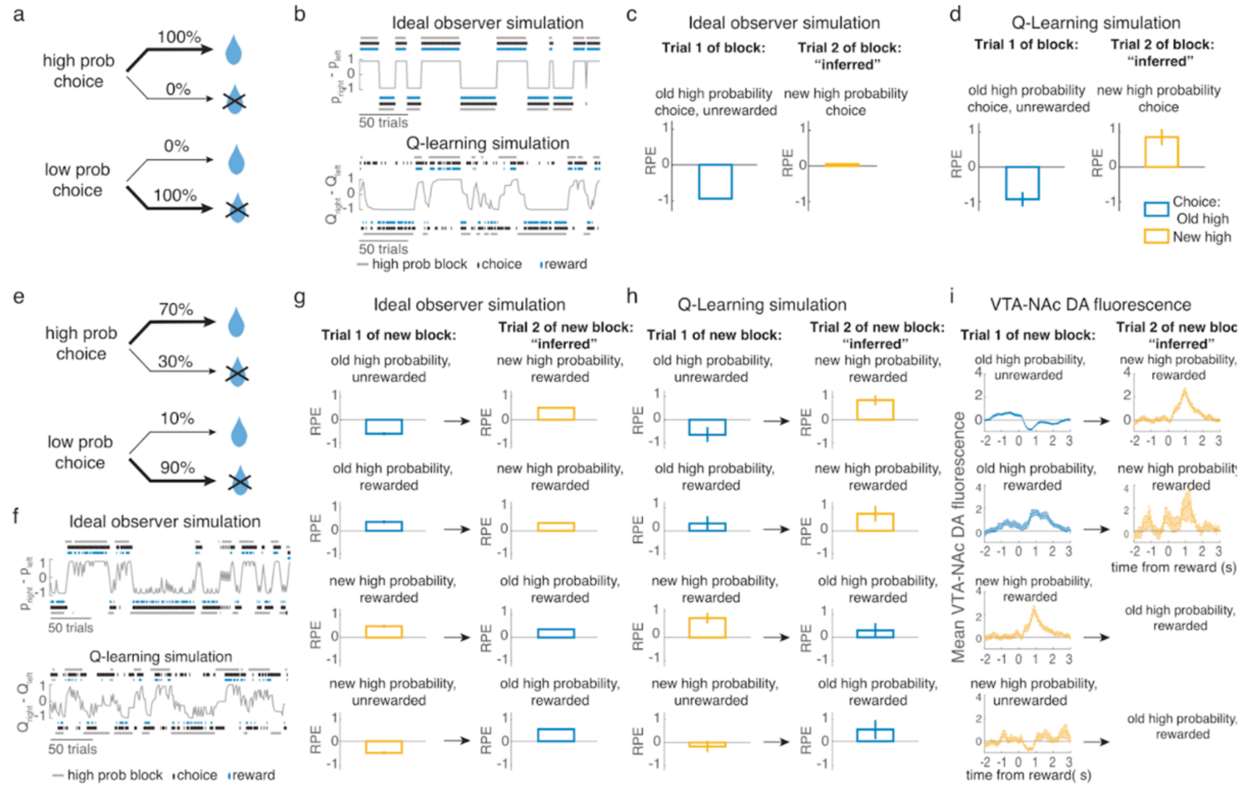


Figure 2.21. **Similar RPE signatures for ideal observer and Q-learning simulation during reversal learning. Related to Figure 2.5 and Methods.** Given previous evidence that dopamine signals can reflect knowledge of task structure [181, 194], we used modeling to gain insight into how clearly RPE in the probabilistic reversal learning task (a bandit task) can indicate the use of model-based inference of block reversals for different reward probability structures. This was done by simulating task performance using an ideal observer model with knowledge of the block structure, and a Q-learning model which did not have information about the block structure (see Methods for more details). (a-d) To confirm that our ideal observer simulation captured a previously reported RPE signature of model-based block reversal inference, we first simulated behavior of 100,000 trials of a task similar to that used in [181]. (a) In this task, one option was rewarded 100% of the time while the other was never rewarded, and the identity of the high probability choice randomly reversed with a probability of 0.05 on each trial. (b) Example performance of the ideal observer simulation (top) and the Q-learning simulation (bottom). Choice is determined by the difference between expected reward for the available actions, ρ, for the ideal observer and the difference between the action values, Q, for the Q-learning simulation (these values are plotted in grey). (c) To evaluate RPE signatures of model-based block reversal inference, we compared the estimated RPE (experienced reward

minus expected reward for the chosen action) on trial 1 and trial 2 of the new block. The RPE on trial 2 was low for the high probability choice in the new block even without direct experience of that action-outcome pairing. This means that the ideal observer infers the block reversal, so the new, not yet experienced reward contingency is expected and the RPE is low. (d) In contrast, because the Q-learning model only updates the value of the chosen action, on trial 2, when the simulation is rewarded for the previously low-probability choice, the reward remains unexpected and the RPE is high. e-i) Simulated performance of 100,000 trials using the reward probabilities from this study. (e) The high probability action was rewarded 70% of the time while the low probability action was rewarded 10% of the time and the blocks reversed according to the same rule as in a. (f) Example performance of the ideal observer (top) and the Q-learning (bottom) simulations in this task. (g) To determine whether there is a strong qualitative RPE signature of block reversal inference in this task, we compared RPE on the 4 possible trial-1 types to RPE on the subsequent rewarded switch trials (i.e. choice on trial 2 was different than trial 1, meaning that any changes in RPE must be inferred). We focus on rewarded trials to aid comparison with reward responses recorded in dopamine terminals during this task [26]. In this case, inference of the block reversal is not obviously reflected in the RPE, since the RPE for a given action on trial 1 and trial 2 are similar (comparing the same color bars for rewarded actions on trials 1 and 2). This is because, even though the ideal observer updates the predicted reward for both the chosen and unchosen actions, when reward delivery is probabilistic, predicted reward remains moderate for both actions and RPE changes only subtly. (h) Same as in g for the Q-learning simulation. As expected, RPE looks very similar on trial 1 and trial 2 for a given rewarded action because the Q-learning simulation does not update the value of the unchosen action on trial 1. (i) Consistent with the results from both simulations, GCaMP6f zscored dF/F from dopaminergic axons in the NAc recorded in [26] is also very similar for a given rewarded action on trial 1 and trial 2. Note that the mice did not make all possible choices in this task, so some trial types are missing. The simulations also rarely made these choices (e.g., switch following a rewarded new high probability choice). Error bars in c-h are SEM across block transitions. Data in i are represented as mean ± SEM across 11 recording sites)

CHAPTER 3

SARS-CoV-2 Omicron Spike Simulations: Broad Antibody Escape, Weakened ACE2 Binding, and Modest Furin Cleavage

This chapter appears as an article [195] published in Microbiology Spectrum 2023. This work was done in collaboration with M. Zaki Jawaid, R. Mahboubi-Ardakani, Richard L. Davis, Daniel L. Cox.

3.1. Introduction

The omicron variant of the SARS-CoV-2 virus was first detected publicly in Nov 2021 [196], and traced back to variants which appeared in mid 2020. Because the variant contains a large number of mutations relative to the original strain, including three relevant regions of the viral surface spike protein (the receptor binding domain (RBD), the furin cleavage domain (FCD), and the n-terminal domain (NTD)), the variant is of great concern. According to current GISAID data, the global infection landscape is almost exclusively dominated by omicron sub-variants, particularly BA1, BA2, and BA2.12.1, with recent emergence of BA4 and BA5 [197].

The fitness of a particular variant depends upon several factors. First, strong binding to surface receptors is of critical importance, and the SARS-CoV-2 RBD binds with high affinity to the angiotensin converting enzyme 2 (ACE2) protein on human cells [198]. This contrasts with likely weaker binding of coronaviruses associated with the common cold such as OC43 which binds more weakly to sialic acid groups on the cell [199]. Second, escaping the background antibody (Ab) spectrum can confer relative fitness over the dominant variant. Third, efficient membrane fusion and transmission is apparently strongly regulated by the FCD, where cleavage can arise both by furin and by transmembrane serine proteases, especially TMPRSS2 [200]. It has been shown, for example, that ferrets inoculated with a WT SARS-CoV-2 with the FCD deleted can become infected but fail to transmit to other ferrets [201]. The delta variant in cultured cells containing endogenous levels of ACE2 and TMPRSS2 has shown significantly enhanced fusion of the viral membrane with the cell membrane [200]. The high viral load of the delta variant has been clearly associated with the mutation P681R of the FCD [202] and has led to the current dominance of SARS-CoV-2 sequences worldwide prior to the omicron emergence [203].

Given the time lag in carrying out protein synthesis, structure determination of bound complexes, determining protein binding affinities, and measuring viral neutralization by Abs for new variants, there is clearly a role for rapid computational studies that can assess the differences of new variants relative to background variants as they arise.

In this paper, we point out here that computational *ab initio* molecular dynamics studies of omicron subvariants RBD-ACE2, RBD-antibody (AB), FCD-Furin, and NTD-antibody are consistent with: 1) robust antibody escape in all regions compared to wild type (WT) and delta, 2) FCD binding to furin intermediate between WT and delta, and 3) weaker binding to the ACE2 than WT or delta. The Ab escape can confer transmissibility advantages for a population with a prevalent delta variant Ab spectrum, but the weaker binding to ACE2 and modest enhancement of furin binding are likely to lead to weaker transmissibility than delta. Due to the high degree of similarity in the RBD and NTD regions of the BA2, BA2.12.1, BA4, and BA5 variants, we present simulation results and subsequent comparisons for WT, delta, BA1, and BA2 variants. For reference, the BA2 RBD is identical to BA2.12.2 RBD with the exception of one mutation (L452Q), the BA4 and BA5 RBD with the exception of residues 486 and 493. The NTD of BA2, BA2.12.2 are identical, while BA4 and BA5 NTD has an additional couple of deletions compared to BA2 NTD. The FCD for all the aforementioned omicron variants are identical.

At the time of writing, the current global infection landscape is dominated by BA2 (24%), BA2.12.2 (13%), BA4 (4%), and BA5(38%) [197]. This work uses ColabFold's [204] implementation of AlphaFold-Multimer [205] to generate structures for FCD-furin binding.

3.2. Results

3.2.1. Binding Strengths: HBond and Binding Free Energy

Before discussing our results, it is important to contextualize what a single HBond difference makes. From earlier work, it has been estimated that a single Hbond in a beta sheet is stabilized by 1.6 kcal/mole [206]. At room temperature ($RT = 0.59$kcal/mole), therefore, using this as a baseline estimate, we would reduce K_D by a factor of $\exp(1.6/.59) \approx 14$ for a single bond, and, e.g., in the case of the 4 Hbond difference for furin binding of the delta FCD over the WT FCD, we would have a reduction of K_D by $\approx 5 \times 10^4$, clearly much stronger binding. We are intending the use of these numbers only for characterizing the significance of the HBond count for energetics and affinity, not to be taken as quantitatively accurate estimates since HBond energetics depend sensitively upon context.

Our main results for interfacial HBonds for the structures of Fig 3.1 are summarized in Fig 3.2. We find somewhat weaker binding to the ACE2 receptor compared to both WT and delta, which should moderate infectivity, and significant antibody escape of the BA1 and BA2 for all three regions (Class I, Class III, and NTD) considered, with the exception of RBD-P4A1 binding for BA2 compared to WT (but still weaker than delta). This escape is measured by the reduction in hydrogen bond count between the antibodies and the spike protein.

For the FCD-Furin binding, six residues fit into the binding pocket, which we argue elsewhere to begin with residue 681 for WT, alpha, and delta [207]. For omicron, we consider the possibility of leading with the N679K mutation or P681H mutation and denote N679K leading furing binding as ”Omicron Alt” in Fig 3.2. The P681H mutation leading is the same as the alpha variant. We see that the expected binding to the FCD is at best the same as the alpha variant, and significantly less than the delta variant.

For the omicron RBD-ACE2 runs, as alluded to above, we carried out additional simulation time for 30 ns vs 10 ns, and we found significantly decreased variability for the last 20 ns. In comparison

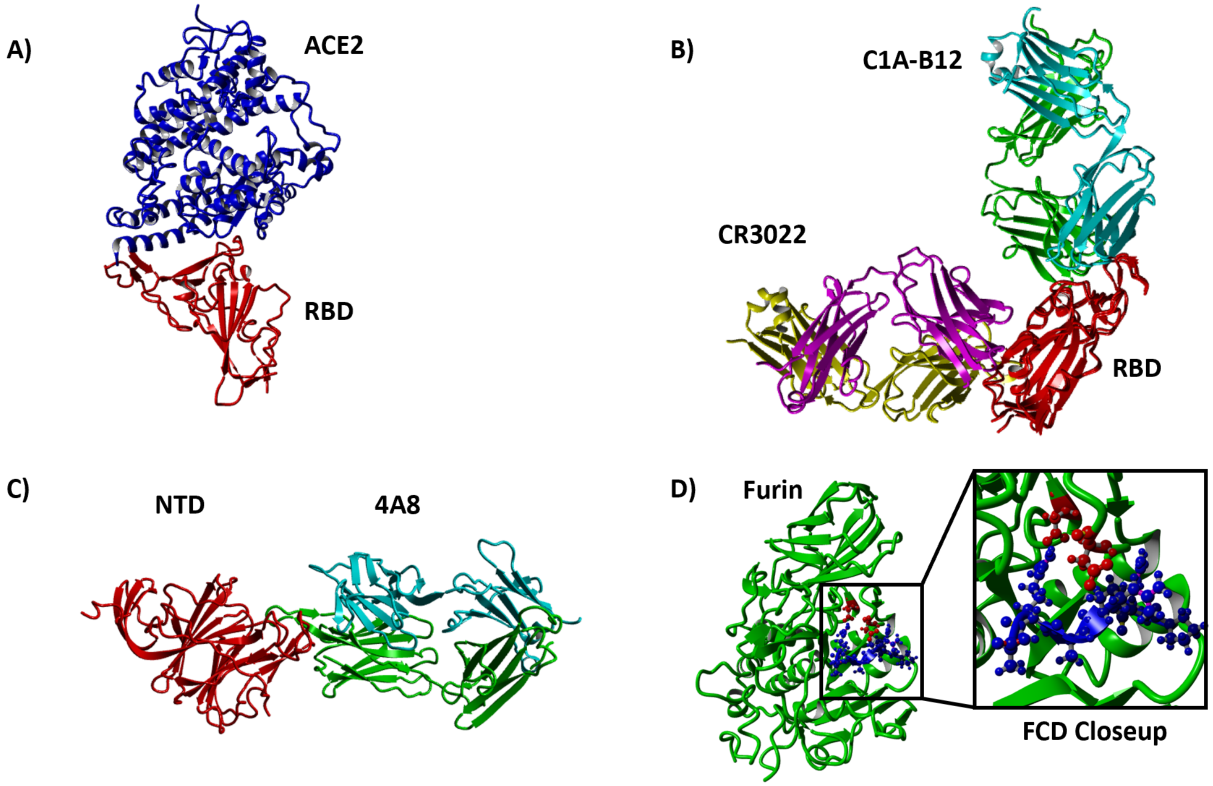


FIGURE 3.1. **Structures of WT spike protein complexes studied** A) ACE2 (red)-RBD (blue) binding (PDB 6m0J). B) Binding of RBD (red) to Class I Ab C1A-B12 (binds in ACE2 interface region; heavy chain green, light chain cyan, PDB 7KFV) and Class III Ab CR3022 (binds away from ACE2; heavy chain magenta, light chain yellow, PDB 6YOR). C) Binding of NTD to 4A8 Ab (heavy chain green, light chain cyan, PDB 7C2L). D) Binding of FCD (blue) to furin (red). Blowup highlighting position of fifth residue R5 (R685 for WT SARS-CoV-2) with proximate aspartic acid residues D151, D199 of the furin enzyme. All AlphaFold PDB files are provided in the Supplementary Material.

with WT, for both the full 30 ns and last 20 ns the p-value is smaller than 0.0001 indicating extreme statistical significance.

For differences between measured Hbond counts, we provide all p-value pairs in Fig 3.2, together with 95% confidence intervals.

The binding energies from the GBSA analysis of molecular dynamics equilibrium conformations are shown in Fig 3.3. The same PDBs are utilized. Evidently the trend of binding energies tracks

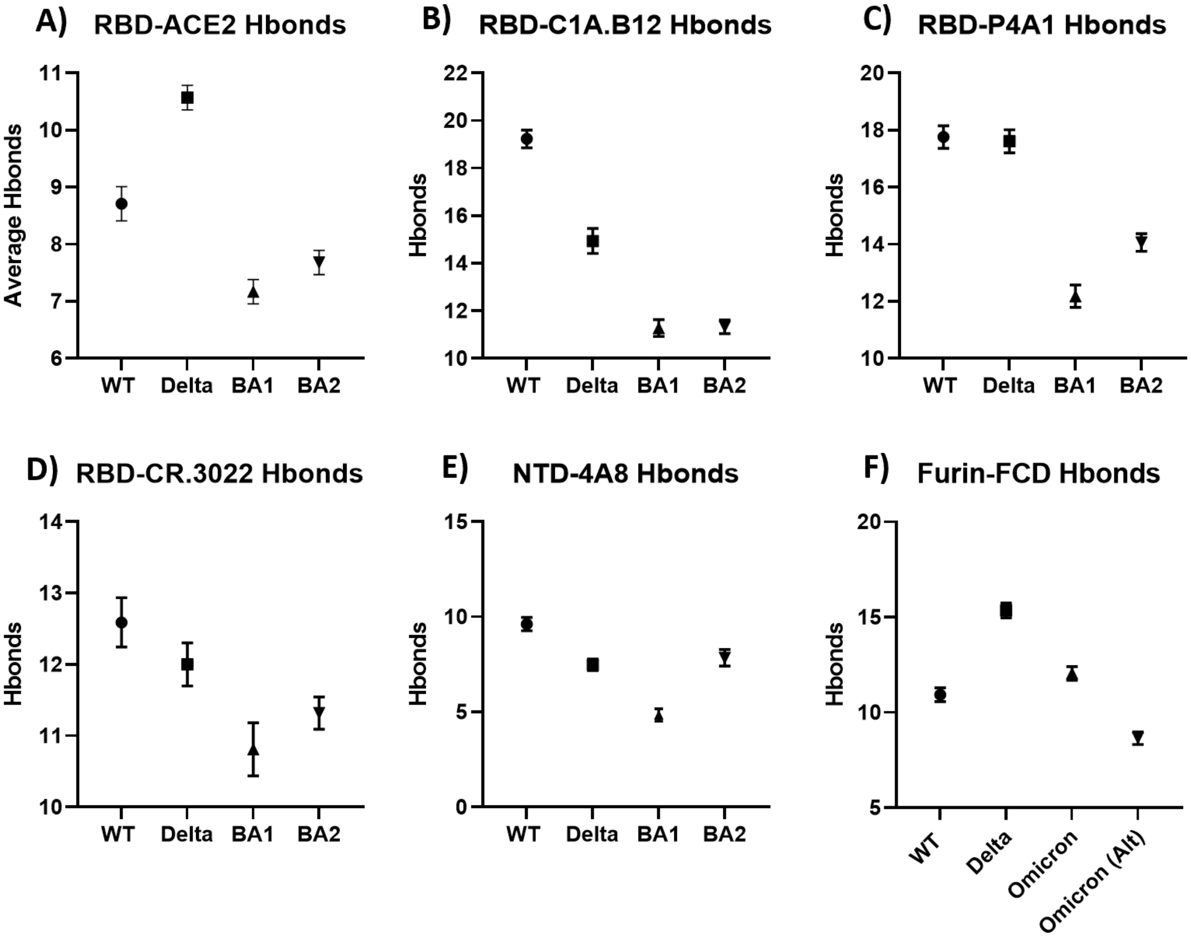


FIGURE 3.2. **Interfacial hydrogen bonds between proteins for WT, delta, BA1, and BA2.** All bars represent 95% CI. A-E) BA1 and BA2 variants participate in significantly fewer interactions than WT and Delta for the simulations shown, with the exception of Delta and BA2 in E). All pairwise p-values in Figs. 2A-E are $p < 0.0001$ (highly significant), with the exception of the aforementioned Delta vs. BA2 pair in E) ($p = 0.19$, not significant), the BA1 vs. BA2 pair in B) ($p = 0.82$, not significant), the WT vs. Delta pair in C) ($p = 0.09$, not significant), and the BA1 vs. BA2 pair in D) ($p = 0.01$, significant). F) FCD-Furin Hbond interactions. All omicron variants participate in slightly higher interactions that WT but less than Delta. We also consider the possibility of the FCD for the omicron variants starting at 679K in Omicron (Alt). All pairwise p-values in F are $p < 0.0001$ (highly significant). All PDB files are referenced in the methods section and provided in the repository referenced in the supplemental material.

well with the easier to estimate interfacial HBond count, with the exceptions of the ACE2-omicron RBD binding.

However, it is very clear that the confidence intervals in Fig 3.3 are relatively larger and overlap more than those from Fig 3.2. The primary reason is in the number of measurements. Because we can draw on large numbers (order hundreds) of simulation snapshots to analyze HBond counts, the error bars are smaller than for GBSA calculations for which time constraints have allowed only 10 snapshots for each interface.

3.2.2. Mutations leading to Ab Escape and weaker ACE2 binding

Fig 3.4 illustrates the key mutations leading to differences in binding for the delta and omicron variants relative to WT.

ACE2 For ACE2 binding, these mutations weaken the ACE2 binding for omicron relative to WT: 1) K417N removes the K417(RBD)-D30(ACE2) salt bridge. 2) Q498R removes hydrogen bonding between the glutamine side chain and K353 of the ACE2 driven by R-K Coulomb repulsion. 3) Y505H removes hydrogen bonding between the Y505 sidechain and the E37 sidechain of ACE2 where the Y505 O acts as a donor. On the other hand, the S477N mutation of omicron enhances bonding relative to wild type, the Q493R mutation enhances the binding to the E35 and D38 acidic residues of ACE2, and the N501Y mutation enhances binding relative to WT. As discussed, the net effect is reduced number of interfacial hydrogen bonds overall. A qualitative picture is provided in Fig 3.4a, while numerical values for detected residue pairs are provided in Supplementary Tables 3.1-3.4.

Class I Abs For Class I antibodies, the following mutations are critical to reducing binding strength of omicron: For binding to P4A1, 1) the Y455 binding to Y33.HC of the Ab heavy chain (HC) is removed. 2) The Q493K, G496S, and Q498R mutations lead to removal of bonds with E101.HC, W32.LC of the Ab light chain, and S67.LC. 3) The Y505H mutation removes bonds to S93.LC. For binding to C1A-B12, 1) the K417N mutation removes a salt bridge to D96.HC, a side chain bond to S98.HC, and weakens a side chain bond to Y52.HC. 2) The mutations Q493R, G496S, and Q498R remove bonds to R100.HC, S30.HC, and S67.HC. 3) The N501Y and Y505H mutations weaken bonds in the 501-505 region to G28.LC, S30.LC, and S93.LC. A complete list of detected residue

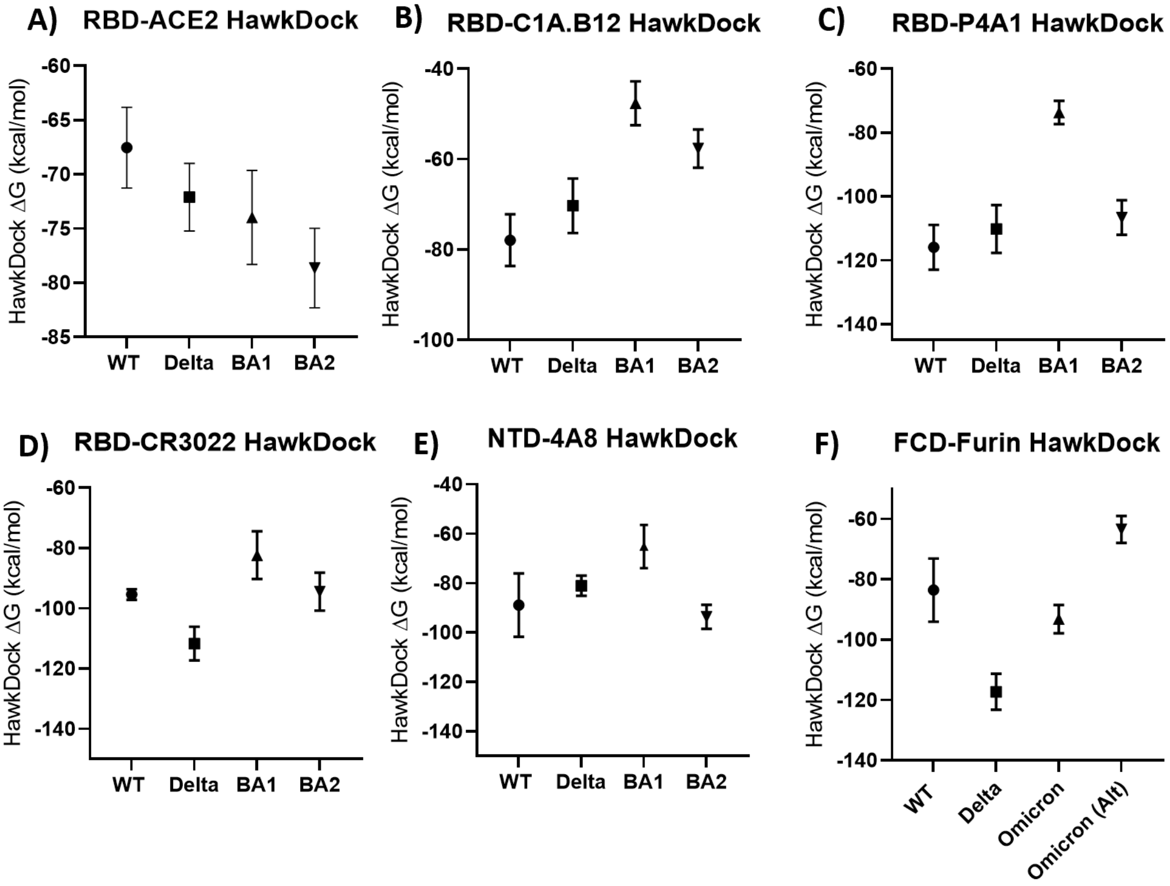


FIGURE 3.3. **GBSA Binding free energy estimate in kcal/mole between proteins for WT, delta, BA1, and BA2.** All bars represent 95% CI. A-F) MM/GBSA binding free energy estimates correlate strongly with the number of Hbonds in Fig 3.2 with the exception of the RBD-ACE2 interactions. All PDB files are referenced in the methods section and provided in the Supplementary Material. p-values for all pairs in A-F are < 0.001 with the following exceptions: A) WT vs. Delta ($p = 0.053$, not significant), WT vs. BA1 ($p = 0.022$, significant), Delta vs. BA1 ($p = 0.44$, not significant), Delta vs. BA2 ($p = 0.007$, significant), and BA1 vs. BA2 ($p = 0.08$, not significant). B) WT vs. Delta ($p = 0.053$, not significant), Delta vs. BA2 ($p = 0.0013$, significant), BA1 vs. BA2 ($p = 0.0024$, significant). C) WT vs. Delta ($p = 0.096$, not significant), WT vs. BA2 ($p = 0.0036$, significant), Delta vs. BA2 ($p =0.23$, not significant) D) WT vs. BA1 ($p = 0.0049$, significant), WT vs. BA2 ($p = 0.75$, not significant), E) WT vs. Delta ($p =0.21$, not significant), WT vs. BA1 ($p = 0.0033$, significant), WT vs. BA2 ($p = 0.44$, not significant), Delta vs. BA1 ($p = 0.0026$, significant)

pairs are provided in Supplementary Tables 3.5-3.12. Fig 3.4b shows binding changes relative to WT for C1A-B12.

Class III Ab For the Class III antibody CR3022, the most noticeable differences compared to WT are 1) the absence of binding at N370 to Y27.HC. This appears to be driven by the hydrophobic substitution S371L, which pulls the asparagine at 370 out of bonding distance from Y27.HC. 2) Weakened bonding of T385 to S100.HC. A complete list of detected residue pairs are provided in Supplementary Tables 3.13-3.16.

NTD Ab For the NTD Ab 4A8, we find that the notable differences of omicron compared to WT are 1) weakened binding at 145-152 presumably due to the deletion at 142-145 relative to WT, and 2) significantly weakened bonding at 246-254 driven by the EPE insertion at 214 and the deletion at 211. Both the 142-145 deletion and the 211 deletion with EPE insertion disrupt the epitope positionings at 145-152 and 246-254 respectively. A complete list of detected residue pairs are provided in Supplementary Tables 3.17-3.20. Fig 3.4c shows binding changes relative to WT for Ab 4A8.

3.2.3. Mutations in the FCD

The FCD (also known as the S1/S2 cleavage site) of SARS-CoV-2 differs from that of SARS-CoV-1 by a polybasic insertion beginning at P681 [208]. Successful cleavage of this region by the Furin enzyme is associated with increased cell to cell and viral transmission *in vitro* [209]. Furthermore, the polybasic insertion at the FCD has been shown to confer SARS-CoV-2 with a selective advantage in lung cells and primary human airway epithelial cells [201].

Due to the absence of structural data for the FCD, as well as the FCD-Furin bound complex, there are limited computational studies of the binding domain. This is because the FCD belongs to a rapidly fluctuating random coil region of the protein which has not been resolved by structural probes (see, e.g., Ref [210], PDB structure 7A94, for which residues 677-688 are unresolved). Additionally, there are no bound Furin-FCD structures available due to furin rapidly cleaving the protein at this domain.

For the generic 681-686 sequence of the FCD, our simulations show that the most critical residue appears to be the 685. In the WT, the arginine is able to form a salt bridge in the interior pocket

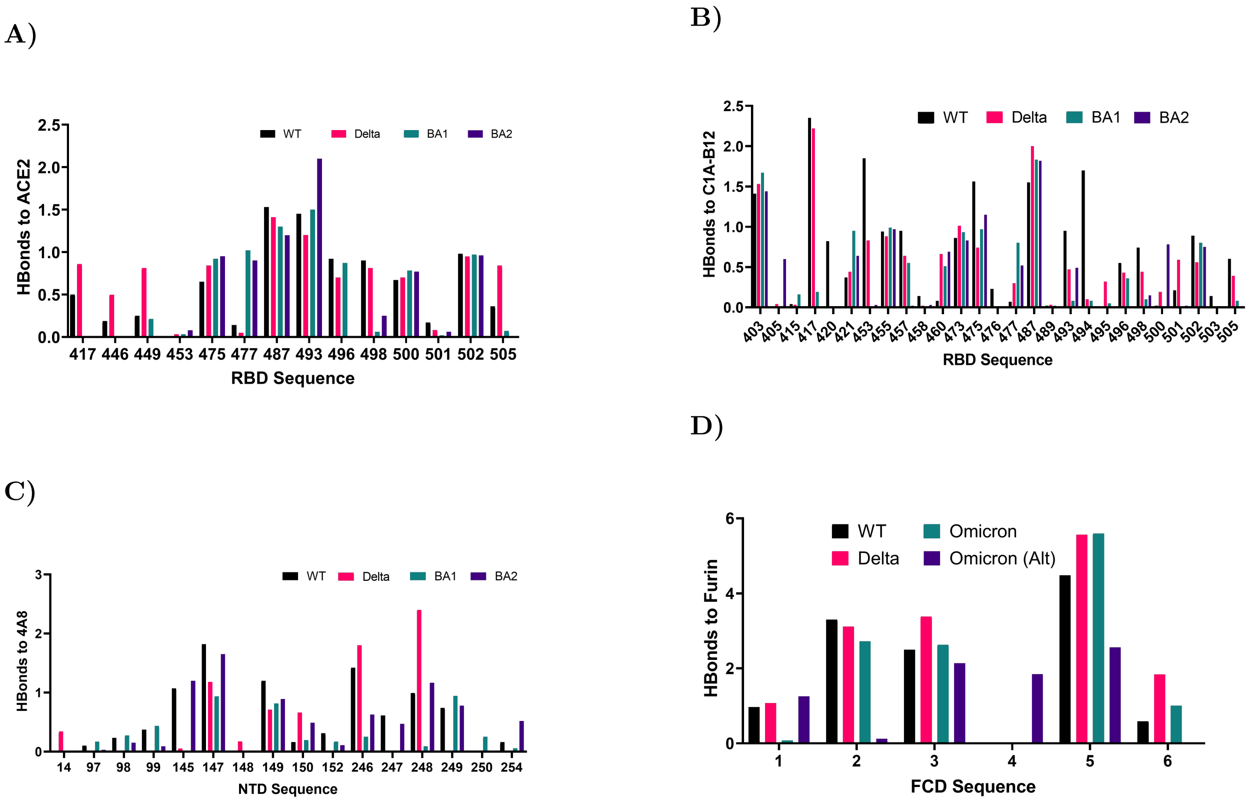


Figure 3.4. **Overview of binding changes for delta and omicron variants relative to WT** Color coding is the same for all charts. For the FCD to furin binding, R1-R6 correspond to 681-686, except for the alternate omicron sequence 679-684. For clarity, RBD binding to P4A1 and CR3022 Abs are not shown. Full residue interaction tables measured by average hydrogen bound strengths are provided in the Supplementary Material.

with D199 of the furin, and bond additionally with S146, W147, D151, A185, and S261. This tendency is illustrated in Fig 3.4d. These bonds are all strengthened for delta and omicron. For the alternate KSHRRA sequence of the omicron, beginning at 679, the position of the arginine in the binding pocket allows only the salt bridge formation with D199. The FCD sequences for the omicron subvariants BA1, BA2, BA2.12.1, BA4, and BA5 are identical and are therefore not differentiated for this part of the study.

As shown in Fig 3.5, we observe that the binding strength, which is determined to a large degree by the binding of the fifth residue of the FCD, correlates inversely with the root mean square fluctuation (RMSF) of the backbone Cα of the first FCD residue at 681. This suggests that locking the 681 Cα as happens for P681R is a key to lowering the fluctuation spectrum of the 685 residue allowing for stronger binding at this site. Evidently, the gain in binding enthalpy offsets any advantages in conformational entropy for the FCD.

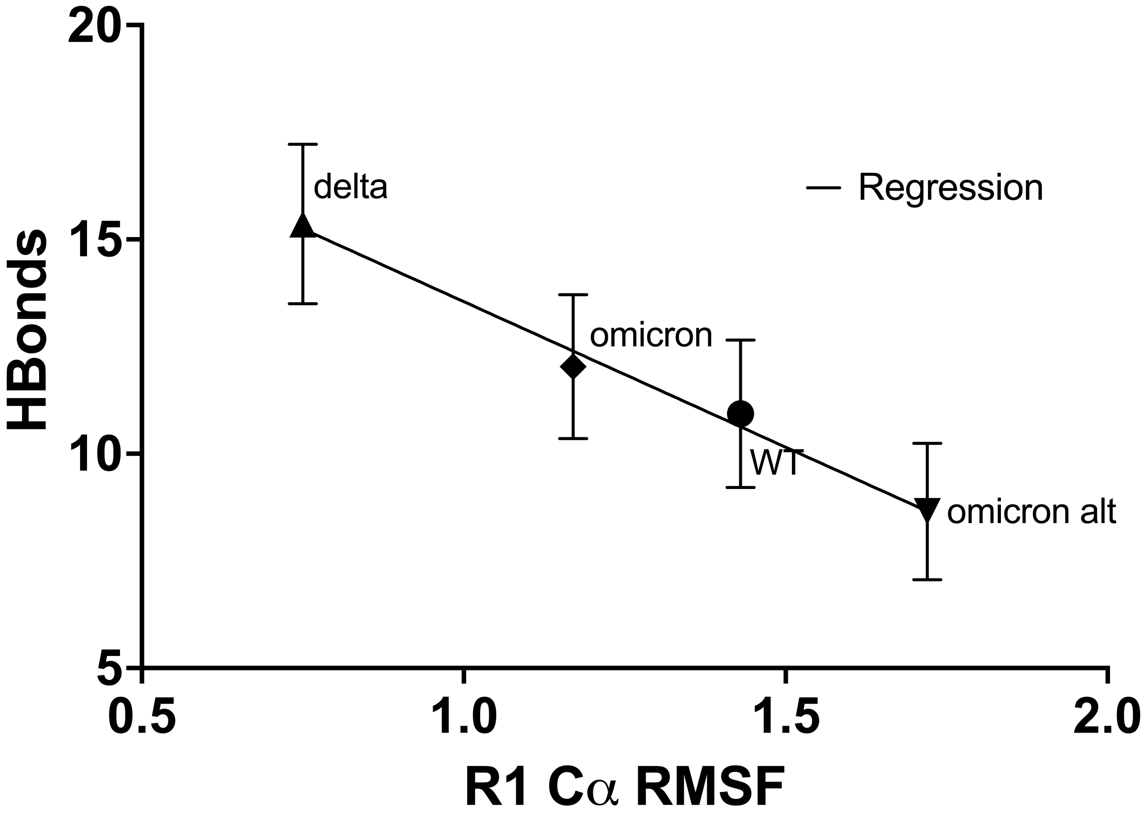


FIGURE 3.5. **Correlation of FCD-furin interfacial HBond count with RMSF of first residue in FCD** The higher the RMSF of the first residue in the FCD, the harder it is to bind to the furin, especially for the critical fifth residue which inserts into the furin pocket as shown in Fig 3.1D. R1 is residue 681 for all but the alternate omicron sequence which starts at residue 679. Full simulation data (RMSF/D) is provided in the respository link of the Supplemental Material. All p-values for HBond counts between pairs are reported in Fig 3.2. The equation of the regression line is $Hbonds = -6.7 \pm 0.7(RMSF) + 20.1 \pm 0.9$, with regression coefficient $R^2 = 0.98$ and is probably negative with $p = 0.01$ (significant)

In a separate work, we test the FCD-Furin binding for over 80 observed and unobserved sequences [207]. We find that among all candidate viral sequences studied, delta is near the very top binding strength within statistical accuracy. The binding strength of several rare sequences match delta within statistical accuracy, as well as some unobserved sequences. Of these, we find that the sequences resulting from P681K (KRRARS) or P681S (SRRARS) mutations in the FCD could, in theory, match delta's binding strength for the FCD-Furin binding. All current omicron variants (BA1-BA5) have P681H [197]. All FCD-Furin hydrogen bonds observed in simulations are summarized in Supplementary Tables 3.21-3.24.

The HBond differences between different FCD sequences are all extremely statistically significant ($p < 0.0001$).

3.3. Discussion

We find weaker binding of the omicron RBD to the ACE2 as measured by HBond counts, with mixed results for GBSA binding energy. In contrast, a number of other theory papers predict stronger ACE2-RBD binding for omicron [211,212,213,214,215,216], but a free energy (alchemical) perturbation analysis of the bound structure predicts weaker binding [217]. The free energy perturbation analysis shares with our work a simulation starting with the observed ACE2-RBD WT structure followed by mutations. In contrast, the other theory approaches separately relax the RBD with mutations and utilize other approaches like docking [212] to bind to the ACE2.

In the Supplementary Figure 3.6, we display the correlation between interfacial RBD-ACE2 H-bond counts and GBSA binding energy for the variants included here as well as six additional variants. The correlation excluding the BA.1 and BA.2 variants is strong, with an R^2 coefficient of 0.85. The high GBSA binding energies for BA.1 and BA.2 suggest an overestimate of binding in the approach, with the largest single contribution at the Q493R residue which contributes -12.7 Kcal/mole for BA.2, versus -5.3 Kcal/mole for the WT RBD. Given our experience of strong correlations of H-bond counts with GBSA energies for antibody and furin binding as well, we believe this does represent an overestimate of binding free energy for the omicron variants.

Since first posting our work, a number of experimental papers have emerged demonstrating explicitly weaker binding of omicron RBD to ACE2 [218], weaker RBD binding and fusogenicity (consistent with weaker furin cleavage) [219, 220], and weaker expression in lung tissues (though stronger in bronchial tissue [221]). These offer support for the predictions here. A surprise from fusogenicity studies, which reflect directly on the furin mediated cleavage at the FCD, is that omicron is 5-10 times weaker than WT or delta at yielding syncytia [219, 220]. If there is a kinetic competition between sequence binding involving the N679K and P681H mutation to get the fifth residue into the deep furin pocket, there could be a strongly reduced cleavage and fusogenicity.

On the other hand, a study examining furin mediated cleavage directly on larger peptides than those considered here found that omicron led to more rapid furin cleavage than WT or delta, and that this was associated with the N679K mutation as the differences largely vanished between the three variants with this mutation [222]. Elsewhere, we have shown that the longer peptides can bind in a reverse orientation, and this rationalizes the difference between the variants [207]. Clarification will come with more experimental studies, including furin-FCD binding studies on the minimal six residue peptides considered here.

The binding strength of Furin to the FCD appears to correlate well with the fluctuations of the initial residue at 681. The lower the fluctuation of the backbone carbon, the lower the fluctuation of the backbone carbon for residue 685, which dominates the bonding to the furin. The P681R mutation provides the lowest Cα RMSF observed amongst the four FCD examples considered here, and the alternate K679 starting point for omicron provides the largest Cα RMSF.

The lower severity of omicron versus delta may be related to the Furin Cleavage Domain. It has been shown that this insertion is critical to the higher transmissibility of SARS-CoV-2 [201, 223] over SARS-CoV-1, and that the mutations P681H for the alpha and omicron variants and P681R for the delta variant play a large role in increased transmissibility of the variants over the wild type (WT) [202]. After initial binding to the human ACE2 protein, Furin protease cleavage breaks the spike to facilitate cell wall fusion [202] and viral reproduction. The stronger the furin-FCD HBond binding, the more efficient the fusion at the molecular level, and ultimately, higher viral load on the host. If the omicron acquired the P681R mutation over the P681H one, the combined antibody escape and enhanced fusion would be highly concerning.

We note that furin is not the only human enzyme which plays a role in spike cleavage, and potential pathogenicity of the virus. Notably, inefficient binding of omicron to the TMPRSS2 compared to delta appears to explain the lower fusogenicity of omicron in lung epithelial cells while having comparable replication in upper respiratory cells that do not express TMPRSS2 [224]. It has also been shown that the metalloprotein enzyme ADAM10 which is expressed in lung tissues facilitates syncytia formation [225].

From an evolutionary perspective, deep mutational scan data for every point mutation of the RBD shows that few mutations lead to enhanced binding, and for the ones that do the effect is modest, while reduced binding by mutation can be dramatic [226]. This suggests that ACE2 binding is already near optimal for the WT RBD. The huge number of RBD mutations that effect antibody escape for omicron inevitably drive the virus away from this optimal binding. Similarly, for the FCD, we find here and elsewhere that the binding is near optimal for the delta variant [207]. Other mutants are more likely to be suboptimal or deleterious to fusion as has been observed.

In summary, a consistent picture of omicron in comparison to the delta strain is emerging. Hospitalization data points to higher disease transmissibility but lower severity for the omicron strain compared to delta [227]. Our simulations see lower interfacial HBond counts for omicron for known RBD and NTD binding regions consistent with this, as well as weaker ACE2 binding and furin binding than the delta variant. Against an immunity background tuned to the delta variant, omicron variants are more transmissible, and subsequent mutations in BA.2, BA.5 will lead to higher transmissibility against an omicron (BA.1) tuned immunity background. Experimental studies of the binding of the RBD to ACE2 and the correlation of fusogenicity with furin binding offer support these predictions as noted above, but more direct experiments are necessary to confirm the predictions here.

Acknowledgments

We acknowledge useful conversations with Javier Arsuaga, Victor Muñoz, and Mariel Vazquez.

3.4. Methods

3.4.1. Molecular Models

A summary of all the mutations in the RBD and N-terminus of the spike protein for the four variants presented here is found in Supplementary Table 3.25.

We drew our starting structures for RBD-ACE2 binding from the PDB file [228]. For Class I ABs, which bind in the same region of the RBD as the ACE2, we used C1A.B12 (PDB:7CJF [229]) and 7KVF P4A1 (PDB:7KVF [230]) (P4A1), while as a representative class III Ab that binds to the RBD away from the ACE2 interface, we used CR.3022 (PDB:6YOR [231]). For an NTD-Ab we used 4A8 (PDB:7C2L [232]).

The antibodies chosen do not comprehensively portray all neutralizing Abs for the SARS-CoV-2 spike, but are representative of the spectrum of antibodies that neutralize the SARS-CoV-2 virus. This study does not account for t-cell binding sites [233]. Fig 3.1 shows the structures of the different complexes studied in this paper.

3.4.2. Molecular Dynamics

To simulate the protein-protein interactions, we used the molecular-modelling package `YASARA` [234] to substitute individual residues and to search for minimum-energy conformations on the resulting modified structures of the complexes listed in Supplementary Table 3.26 (hydrogen bonds) and Supplementary Table 3.27 (binding energy estimates). For all of the structures, we carried out an energy-minimization (EM) routine, which includes steepest descent and simulated annealing (until free energy stabilizes to within 50 J/mol) minimization to remove clashes. All molecular-dynamics simulations were run using the AMBER14 force field with [235] for solute, GAFF2 [236] and AM1BCC [237] for ligands, and TIP3P for water. The cutoff was 8 Å for Van der Waals forces (AMBER's default value [238]) and no cutoff was applied for electrostatic forces (using the Particle Mesh Ewald algorithm [239]). The equations of motion were integrated with a multiple timestep of 1.25 fs for bonded interactions and 2.5 fs for non-bonded interactions at $T = 298$ K and $P = 1$

atm (NPT ensemble) via algorithms described in [240]. Prior to counting the hydrogen bonds and calculating the free energy, we carry out several pre-processing steps on the structure including an optimization of the hydrogen-bonding network [241] to increase the solute stability and a pK_{a} prediction to fine-tune the protonation states of protein residues at the chosen pH of 7.4 [240]. Insertions and mutations were carried out using `YASARA`'s BuildLoop and SwapRes commands [240] respectively. Simulation data was collected every 100ps after 1-2ns of equilibration time, as determined by the solute root mean square deviations (RMSDs) from the starting structure. For all bound structures, we ran for at least 10ns post equilibrium, and verified stability of time series for hydrogen bond counts and root mean square deviation (RMSD) from the starting structures. Because of concerns about the validity of short time simulations, and more variability for the weaker binding for the omicron RBD-ACE2 complexes, we ran for 30ns postequilibration in those cases.

The hydrogen bond (HBond) counts were tabulated using a distance and angle approximation between donor and acceptor atoms as described in [241].

Note that in this approach, salt bridges of proximate residues, are effectively counted as H-bonds between basic side chain amide groups and acidic side chain carboxyl groups.

We provide all molecular dynamics simulation analysis, including PDB snapshots, RMSD/F, as well specific residue-residue Hbond interactions for all 24 of our simulations in the supplemental material. Net hydrogen bond counts are summarized in Supplemental Tables.

3.4.3. Endpoint Free Energy Analysis

We calculated binding free energy for the energy-minimized structure using the molecular mechanics/generalized Born surface area (MM/GBSA) method [242, 243, 244], which is implemented by the `HawkDock` server [245]. While the MM/GBSA approximations overestimate the magnitude of binding free energy relative to *in-vitro* methods, the obtained values correlate well with H-bond counts. For each RBD-ACE2, RBD-AB, and NTD-Ab binding pair we average over ten snapshots of equilibrium conformations. For each FCD-furin pair, we average over five snapshots of equilibrium conformations.

3.4.4. Use of ColabFold/AlphaFold for Furin cleavage domain

Due to the absence of structural data for the FCD-Furin bound complex, we model the FCD-Furin bound structure using the heterocomplex prediction method known as AlphaFold-Multimer [35, 205] as implemented within ColabFold [204] to predict the best bound structure to the furin enzyme of the six residue FCD from the WT protein. We inferred the ordering of this sequence by comparison with a very similar six residue peptide inhibitor of furin with the sequence RRRVR-aminomethyl-benzamidine (RRRVR-amba) [246]. In this case the backbone of the WT FCD aligns well with that of the inhibitor, but the fifth arginine enters a furin pocket while the amba enters the furin pocket for the inhibitor. The serine is in proper cleavage position for furin. The delta and omicron structures were then obtained by mutation from the predicted WT FCD-furin structure. In a separate work, we present a complete description of the use of ColabFold/AlphaFold for modeling the FCD-furin binding as well as simulation results of over 60 observed FCD sequences for SARS-CoV-2 and other commonly observed coronaviruses [207]. In this study, we limit our FCD-Furin binding focus to sequences from WT, delta, and omicron variants.

All PDB files generated using AlphaFold as well as the simulated data associated with them are provided in the supplemental material.

3.4.5. Statistical Analysis

We computed the statistical significance of pairwise differences using GraphPad unpaired t-test.

3.5. Supplementary Materials

All PDB files for simulations and AlphaFold files for the structures here can be found in the link

https://drive.google.com/drive/folders/1d3H4hA4j6L4ICdXrzrswhkz0F2z0ak2W?usp=sharing

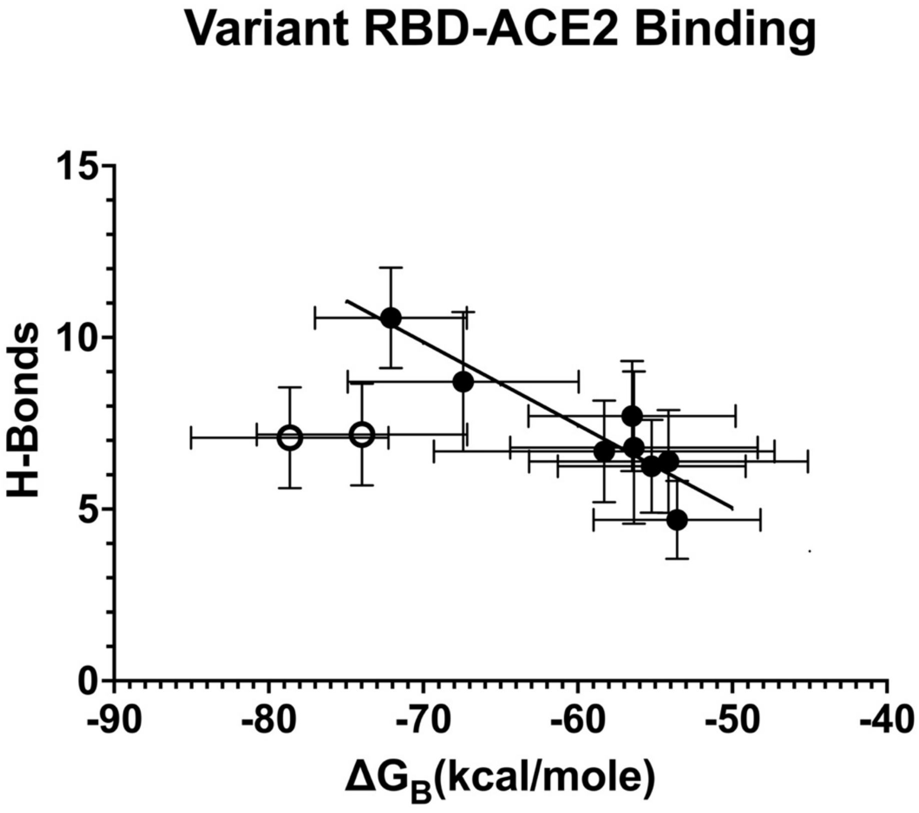


Figure 3.6. **Correlation of HBonds and Binding Free energy** The solid circles are from simulations (left to right) Delta RBD-ACE1, WT RBD-ACE2, N439K RBD-ACE2, L452R RBD-ACE2, N501T RBD-ACE2, Beta RBD-ACE2, V367F RBD-ACE2, Alpha RBD-ACE2. The solid line is a regression fit to those 8 points ($H - Bonds = -0.2415\Delta G_B - 7.09$ which has a regression coefficient $R^2 = 0.85$. Open circles are BA.2 (left) and BA.1 (right).

Detailed list of Hbond pairs

The following tables show Hbond pairs and their detection ratios for our simulations. The Hbond pairs are counted using a distance and angle approximation as described in methods.

The bond detection ratio is defined as n/N, where n is the number of snapshots in which the labelled Hbond is detected, and N is the total number of snapshots. Therefore, a bond detection ratio of 1 means that the corresponding bond was detected in every single snapshot of the simulation.

WT RBD	**ACE2**	**Bond detection ratio**
A475	Q24.A	0.0625
	S19.A	0.596
G446	Q42.A	0.187
G476	S19.A	0.005
G496	K353.A	0.927
G502	K353.A	0.980
K417	D30.A	0.384
	H34.A	0.221
N487	Q24.A	0.557
	Y83.A	0.980
N501	K353.A	0.052
	Y41.A	0.129
Q493	E35.A	0.855
	H34.A	0.043
	K31.A	0.543
Q498	D38.A	0.293
	K353.A	0.596
	Q42.A	0.028
R403	H34.A	0.004
S477	Q24.A	0.005
	S19.A	0.139
T500	D355.A	0.365
	N330.A	0.019
	Y41.A	0.317
Y449	D38.A	0.05
	Q42.A	0.21
Y505	E37.A	0.36
	R393.A	0.01

Table 3.1. HBond Pairs for WT RBD-ACE2 simulation.

Delta RBD	**ACE2**	**Bond detection ratio**
A475	Q24	0.095
	S19	0.754
G446	Q42	0.498
G496	K353	0.706
G502	K353	0.948
K417	D30	0.806
N487	Q24	0.507
	Y83	0.915
N501	K353	0.014
	Y41	0.076
Q493	E35	0.768
	H34	0.024
	K31	0.445
Q498	D38	0.028
	K353	0.735
	Q42	0.009
S477	S19	0.005
T500	D355	0.27
	N330	0.062
	Y41	0.502
Y449	D38	0.924
	Q42	0.379
Y453	H34	0.066
Y505	E37	0.829
	R393	0.043

Table 3.2. HBond Pairs for Delta RBD-ACE2 simulation.

Omicron BA1 RBD	**ACE2**	**Bond detection ratio**
A475	Q24	0.037
	S19	0.893
G502	K353	0.968
H505	K353	0.068
N417	H34	0.006
N477	Q24	0.025
	S19	0.993
N487	Q24	0.262
	Y83	0.875
R403	H34	0.012
R493	E35	1.281
	H34	0.225
	K31	0.006
R498	Q42	0.056
S496	D38	0.862
	K353	0.012
T500	D355	0.443
	N330	0.112
	Y41	0.206
Y449	D38	0.212
Y453	H34	0.031
Y501	D38	0.019

TABLE 3.3. Bond Pairs for Omicron BA1 RBD-ACE2 simulation.

Omicron BA2 RBD	ACE2	Bond detection ratio
A475	Q24	0.074
	S19	0.880
G502	K353	0.955
H519	D615	0.004
N477	Q24	0.024
	S19	0.885
N487	Q24	0.333
	Y83	0.910
R493	D38	1.686
	E35	0.766
	H34	0.024
	K31	0.004
R498	Q42	0.248
T500	D355	0.631
	N330	0.069
	Y41	0.134
Y453	H34	0.084
Y473	E23	0.004
Y489	Q24	0.004
Y501	K353	0.064

Table 3.4. Bond Pairs for Omicron BA2 RBD-ACE2 simulation.

WT RBD	C1A-B12	Bond detection ratio
A475	N32.H T28.H	0.911 0.65
D420	Y58.H	0.821
G476	T28.H	0.228
G496	S30.L	0.553
G502	G28.L	0.894
K417	D96.H S98.H Y52.H	0.992 0.667 0.691
K458	S30.H S31.H	0.041 0.098
L455	Y33.H	0.943
N460	G54.H	0.065
N487	G26.H R94.H	0.496 1.049
N501	S30.L	0.211
Q493	R100A.H Y100B.H	0.935 0.016
Q498	S30.L S67.L	0.268 0.472
R403	I92.L S93.L	1.398 0.008
R457	S53.H	0.951
S477	G26.H	0.065
S494	R100A.H Y32.L	1.659 0.041
T415	Y58.H	0.041
T500	G28.L	0.024
V503	Q27.L	0.138
Y421	G54.H	0.374
Y453	G99.H Y32.L	0.967 0.878
Y473	S31.H S53.H	0.846 0.008
Y489	R94.H	0.024
Y505	D1.L S93.L	0.041 0.528

Table 3.5. HBond Pairs for WT RBD-C1A-B12 (7KFV) antibody simulation. Here, ‘H’ are residues from the heavy chain and ‘L’ are residues from the light chain.

Delta RBD	**C1A-B12**	**Bond detection ratio**
A475	N32.H R94.H	0.65 0.056
G496	S30.L	0.427
G502	G28.L	0.517
K417	D96.H S98.H Y52.H	0.979 0.566 0.678
K458	S30.H S31.H	0.014 0.014
L455	Y33.H	0.881
N460	G54.H	0.664
N487	R94.H Y102.H	1.552 0.441
N501	S30.L	0.573
Q493	R100A.H Y100B.H Y32.L	0.399 0.042 0.035
Q498	S30.L S67.L	0.035 0.413
R403	I92.L S98.H	1.455 0.077
R457	S53.H	0.643
S477	E 1 .H G26.H	0.007 0.294
S494	R100A.H Y32.L	0.049 0.049
T415	Y58.H	0.028
T500	G28.L Q27.L	0.182 0.014
Y421	G54.H S53.H	0.413 0.035
Y453	G99.H Y32.L	0.58 0.252
Y473	S31.H	0.979
Y489	R94.H	0.035
Y495	Y32.L	0.322
Y505	Q27.L S93.L	0.119 0.266

Table 3.6. HBond Pairs for Delta RBD-C1A-B12 (7KFV) antibody simulation. Here, ‘H’ are residues from the heavy chain and ‘L’ are residues from the light chain.

Omicron BA1	C1A-B12	Bond detection ratio
A475	N32.H	0.917355
	T28.H	0.066116
D405	S93.L	0.008264
G502	G28.L	0.809917
H505	Q27.L	0.082645
K478	E 1 .H	0.024793
L455	Y33.H	0.991736
N417	Y33.H	0.066116
	Y52.H	0.132231
N460	G54.H	0.438017
	G55.H	0.082645
N477	G26.H	0.760
	T28.H	0.049
N487	G26.H	0.016
	R94.H	1.636
	Y102.H	0.190
Q474	S31.H	0.008
R403	I92.L	1.677
R408	Y58.H	0.008
R457	S53.H	0.553
R493	Y100B.H	0.082
R498	S30.L	0.107
S494	R100A.H	0.082
S496	R100A.H	0.363
T415	Y58.H	0.165
V503	Q27.L	0.008
Y421	G54.H	0.719
	S53.H	0.016
	Y33.H	0.231
Y449	R100A.H	0.008
Y473	S31.H	0.933
Y489	R94.H	0.024
Y495	Y32.L	0.049
Y501	S30.L	0.008

Table 3.7. HBond Pairs for Omicron BA1 RBD-C1A-B12 (7KFV) antibody simulation. Here, 'H' are residues from the heavy chain and 'L' are residues from the light chain.

Omicron BA2	C1A-B12	Bond detection ratio
A475	N32.H	0.908602
	R94.H	0.010753
	S31.H	0.010753
	T28.H	0.139
G502	Q27.L	0.752
K458	S31.H	0.032
L455	Y33.H	0.973
N417	Y33.H	0.005
N460	G54.H	0.698
N477	G26.H	0.526
N487	G26.H	0.053
	R94.H	1.639
	Y102.H	0.209
R403	I92.L	0.924
	S98.H	0.521
R405	D 1 .L	0.478
	Y94.L	0.139
R457	S53.H	0.021
R493	S31.L	0.010
	Y32.L	0.478
R498	G28.L	0.032
	S30.L	0.118
	S67.L	0.016
T500	G28.L	0.784
Y421	G54.H	0.424
	S53.H	0.091
	Y33.H	0.220
Y453	G99.H	0.032
Y473	S31.H	0.817
	S53.H	0.021
Y495	Y32.L	0.010
Y501	I29.L	0.021

Table 3.8. HBond Pairs for Omicron BA2 RBD-C1A-B12 (7KFV) antibody simulation. Here, ‘H’ are residues from the heavy chain and ‘L’ are residues from the light chain.

WT RBD	P4A1	Bond detection ratio
A475	I28.H	0.667
	N32.H	0.755
D420	S56.H	0.892
G496	S30.C	0.873
G502	G28.C	0.598
K417	E101.H	0.039
	Q100.H	0.912
K458	S30.H	0.176
	S31.H	0.52
L455	Y33.H	0.824
N460	G54.H	0.422
N487	G26.H	0.461
	R97.H	1.48
N501	S30.C	1.167
Q493	E101.H	0.647
Q498	S67.C	0.569
R403	N92.C	1.843
R457	S53.H	0.196
S477	G26.H	0.343
T500	G28.C	0.01
V503	Q27.C	0.01
Y421	G54.H	0.667
Y453	E101.H	0.882
Y473	S31.H	0.853
Y489	R97.H	0.049
Y495	W32.C	0.843
Y505	S93.C	1

Table 3.9. HBond Pairs for WT RBD-P4A1 (7CJF) antibody simulation. Here, 'H' are residues from the heavy chain and 'L' are residues from the light chain.

Delta RBD	**P4A1**	**Bond detection ratio**
A475	I28.H	0.725
	N32.H	0.804
D420	S56.H	0.912
G496	S30.L	0.922
G502	G28.L	0.627
K417	E101.H	0.363
	Q100.H	0.951
K458	S30.H	0.186
	S31.H	0.422
	S53.H	0.02
L455	Y33.H	0.745
N460	G54.H	0.402
N487	G26.H	0.5
	R97.H	1.52
N501	S30.L	1.343
Q493	E101.H	0.167
Q498	S67.L	0.529
R403	N92.L	1.794
R457	S53.H	0.265
S477	G26.H	0.265
V503	Q27.L	0.01
Y421	G54.H	0.559
	S53.H	0.039
Y449	S31.L	0.01
Y453	E101.H	0.853
Y473	S31.H	0.814
Y489	R97.H	0.029
Y495	W32.L	0.843
Y505	S93.L	1

TABLE 3.10. HBond Pairs for Delta RBD-P4A1 (7CJF) antibody simulation. Here, ‘H’ are residues from the heavy chain and ‘L’ are residues from the light chain.

Omicron BA1 RBD	P4A1 (7CJF)	Bond detection ratio
A475	I28.H	0.790
	N32.H	0.856
D420	S56.H	0.950
G416	S56.H	0.011
G502	G28.L	0.093
	S30.L	0.016
H505	G28.L	0.027
	W32.L	0.254
K458	S30.H	0.082
	S31.H	0.121
	S53.H	0.005
L455	Y33.H	0.022
N417	Q100.H	0.723
	Y33.H	0.961
N460	G54.H	0.232
	G55.H	0.027
	S56.H	0.055
N477	G26.H	0.950
N487	R97.H	1.419
Q474	S31.H	0.027
R403	N92.L	0.939
R457	S53.H	0.718
R493	E101.H	0.933
R498	S30.L	0.027
	S67.L	0.033
T500	S30.L	0.027
V503	Q27.L	0.005
Y421	G54.H	0.403
	S53.H	0.033
	Y33.H	0.480
Y453	E101.H	0.624
Y473	S31.H	0.453
Y489	R97.H	0.071
Y501	W32.L	0.071

TABLE 3.11. HBond Pairs for Omicron BA1 RBD-P4A1 (7CJF) antibody simulation. Here, ‘H’ are residues from the heavy chain and ‘L’ are residues from the light chain.

Omicron BA2 RBD	**P4A1 (7CJF)**	**Bond detection ratio**
A475	I28.H N32.H	0.678218 0.816832
A520	T393	0.00495
D420	S56.H	0.886139
E516 ND	N394	0.00495
G502	G28.L	0.985
H505	N92.L Q27.L	0.044 0.128
K458	S30.H S31.H	0.019 0.123
L455	Y33.H	0.079
N405	N92.L	0.108
N417	Q100.H Y33.H Y52.H	0.821 0.905 0.039
N460	G54.H S56.H	0.311 0.029
N477	F27.H G26.H	0.004 0.514
N487	G26.H R97.H Y107.H	0.034 1.569 0.005
Q474	S31.H	0.005
R403	E101.H N92.L	1.524 0.113
R457	S53.H	0.272
R493	E101.H	0.915
T500	S30.L	0.292
Y421	G54.H S53.H Y33.H	0.658 0.019 0.084
Y453	E101.H	0.099
Y473	S31.H S53.H	0.698 0.064
Y489	R97.H	0.054
Y501	S31.L	0.727

Table 3.12. HBond Pairs for Omicron BA2 RBD-P4A1 (7CJF) antibody simulation. Here, 'H' are residues from the heavy chain and 'L' are residues from the light chain.

WT RBD	CR3022	Bond detection ratio
C379	I102.H	0.932
D428	S32.L	1.429
	S33.L	0.91
	Y31.L	0.023
F374	K74.H	0.008
F377	Y52.H	0.932
G381	I102.H	0.91
	Y38.L	0.985
H519	N35.L	0.098
K378	D55.H	0.902
	E57.H	0.857
K386	D107.H	0.895
	E61.L	0.812
N370	Y27.H	0.707
S366	Y27.H	0.008
S383	S100.H	0.511
	T104.H	0.241
T376	D55.H	0.218
T385	Q 1 .H	0.008
	S100.H	0.902
	Y32.H	0.045
T430	S33.L	0.143
	Y31.L	0.113

Table 3.13. HBond Pairs for Omicron BA1 RBD-CR3022 (6YOR) antibody simulation. Here, ‘H’ are residues from the heavy chain and ‘L’ are residues from the light chain.

Delta RBD	CR3022	Bond detection ratio
C379.E	I102.H	0.935
D389.E	T59.L	0.153
D428.E	S32.L	0.089
	Y31.L	0.048
F374.E	K74.H	0.008
F377.E	Y52.H	0.96
G381.E	I102.H	0.935
	T104.H	0.008
	Y38.L	0.984
H519.E	N35.L	0.032
K378.E	D55.H	0.815
	E57.H	0.855
K386.E	D107.H	0.758
	E61.L	0.871
	Y55.L	0.048
L517.E	N35.L	0.919
	S33.L	0.863
N370.E	G28.H	0.185
Q414.E	E57.H	0.008
S375.E	K74.H	0.04
S383.E	S100.H	0.331
	T104.H	0.089
T376.E	D55.H	0.016
T385.E	D107.H	0.056
	Q 1 .H	0.081
	S100.H	0.524
T430.E	S33.L	0.927
V382.E	T104.H	0.032
Y369.E	Q 1 .H	0.065
Y380.E	E57.H	0.137

Table 3.14. HBond Pairs for Delta RBD-CR3022 (6YOR) antibody simulation. Here, ‘H’ are residues from the heavy chain and ‘L’ are residues from the light chain.

Omicron BA1	**CR3022 (6YOR)**	**Bond detection ratio**
C379	I102.H	0.971
D428	S32.L	1.482
	S33.L	0.352
	Y31.L	0.115
F377	Y52.H	0.949
G381	I102.H	0.719
	Y38.L	0.812
H519	N35.L	0.115
K378	D55.H	0.834
	E57.H	0.784
K386	D107.H	0.748
	E61.L	0.755
	Q 1 .H	0.007
L517	N35.L	0.007
N370	G28.H	0.014
	Y27.H	0.244
N388	Q 1 .H	0.021
	Y55.L	0.007
S366	Y27.H	0.165
S383	S100.H	0.899
	T104.H	0.028
T376	D55.H	0.187
T385	Q 1 .H	0.100
	S100.H	0.460
	T31.H	0.064
	Y32.H	0.035
T430	S33.L	0.071

Table 3.15. HBond Pairs for Omicron BA1 RBD-CR_3022 (6YOR) antibody simulation. Here, 'H' are residues from the heavy chain and 'L' are residues from the light chain.

Omicron BA2	**CR3022 (6YOR)**	**Bond detection ratio**
C379	I102.H	0.968
D428	S32.L	0.819
F375	K74.H	0.021
F377	Y52.H	0.936
F515	S33.L	0.010
G381	I102.H Y38.L	0.856 0.984
H519	N35.L	0.015
K378	D55.H E57.H	0.856 0.691
K386	D107.H E61.L	0.898 0.824
L517	N35.L S33.L	0.632 0.989
N370	Y27.H	0.25
N388	Y55.L	0.015
S366	Y27.H	0.164
S383	S100.H	0.872
T385	S100.H Y32.H	0.005 0.010
T430	S33.L	0.425

Table 3.16. HBond Pairs for Omicron BA2 RBD-CR_3022 (6YOR) antibody simulation. Here, 'H' are residues from the heavy chain and 'L' are residues from the light chain.

WT NTD	**4A8**	**Bond detection ratio**
K147	E72.H	0.703
	L29.H	0.673
	L32.H	0.446
K150	D55.H	0.03
	E54.H	0.079
	P53.H	0.01
	Y111.H	0.04
K97	D107.H	0.099
L249	A103.H	0.05
	V102.H	0.693
N149	D55.H	0.465
	E54.H	0.069
	P53.H	0.663
N99	P106.H	0.366
R246	E31.H	0.02
	G26.H	1.396
S247	Y27.H	0.614
S254	E 1 .H	0.158
S98	D107.H	0.228
T250	T105.H	0.01
W152	Y111.H	0.307
Y145	E31.H	0.327
	T30.H	0.743
Y248	E31.H	0.99

TABLE 3.17. HBond Pairs for WT NTD-4A8 (7C2L) antibody simulation. Here, 'H' are residues from the heavy chain and 'L' are residues from the light chain.

Delta NTD	4A8	Bond detection ratio
A292	D290	0.017
D290	R273	0.017
D294	S297	0.05
H146	T30.H	0.008
K147	E72.H	0.125
	L29.H	0.45
	L32.H	0.583
	T30.H	0.025
K150	D55.H	0.008
	E54.H	0.6
	P53.H	0.033
	Y111.H	0.017
N148	D55.H	0.167
N149	D55.H	0.708
P295	T299	0.042
Q14	E 1 .H	0.117
	G26.H	0.183
	S75.H	0.017
	T28.H	0.025
R246	E31.H	1.8
S247	G104.H	0.008
S254	E 1 .H	0.008
	Y27.H	0.008
S297	C291	0.008
S297	D294	0.183
W258	E 1 .H	0.017
Y145	Y111.H	0.05
Y248	E31.H	0.917
	G104.H	1.483

TABLE 3.18. HBond Pairs for Delta NTD-4A8 (7C2L) antibody simulation. Here, 'H' are residues from the heavy chain and 'L' are residues from the light chain.

BA1 NTD	4A8	Bond detection ratio
K147	E72.H	0.540
	L29.H	0.338
	L32.H	0.048
	S33.H	0.008
K150	E54.H	0.169
	P53.H	0.016
	Y113.H	0.008
K97	D107.H	0.169
L249	V102.H	0.943
N149	D55.H	0.096
	E54.H	0.048
	P53.H	0.669
N99	D107.H	0.016
	P106.H	0.419
R246	E31.H	0.169
	G26.H	0.056
	T100.H	0.008
	Y27.H	0.016
S247	Y27.H	0.016
S254	N58.L	0.032
	Y54.L	0.024
S98	D107.H	0.217
	P106.H	0.056
T250	A103.H	0.088
	G104.H	0.161
W152	D107.H	0.008
	T105.H	0.153
	Y111.H	0.008
Y248	E31.H	0.088

Table 3.19. HBond Pairs for Omicron BA1 NTD-4A8 (7C2L) antibody simulation. Here, ‘H’ are residues from the heavy chain and ‘L’ are residues from the light chain.

BA2 NTD	**4A8**	**Bond detection ratio**
K147.A	E72.H	0.675
	L29.H	0.682
	L32.H	0.299
K150.A	E54.H	0.274
	P53.H	0.21
	Y111.H	0.013
K97.A	D107.H	0.038
L249.A	G104.H	0.389
	V102.H	0.395
N148.A	D55.H	0.006
N149.A	D55.H	0.471
	E54.H	0.013
	P53.H	0.408
N99.A	D107.H	0.019
	P106.H	0.076
R246.A	E31.H	0.115
	G26.H	0.503
	T28.H	0.013
S247.A	Y27.H	0.471
S254.A	E 1 .H	0.459
	S61.L	0.07
S98.A	D107.H	0.153
W152.A	Y111.H	0.115
Y145.A	E31.H	0.363
	T30.H	0.841
Y248.A	E31.H	1
	G104.H	0.146
	Y27.H	0.019

Table 3.20. HBond Pairs for Omicron BA2 NTD-4A8 (7C2L) antibody simulation. Here, ‘H’ are residues from the heavy chain and ‘L’ are residues from the light chain.

FCD	**Furin**	**Bond detection ratio**
P1	E123	0.026
	E150	0.043
	V124	0.184
R2	D157	1.473
	E129	0.885
	G158	0.061
	Y201	0.745
R3	D151	0.228
	E150	0.921
	G148	1.771
R5	A185	0.921
	D151	0.017
	D199	1.263
	P149	0.087
	S146	0.798
	S261	1
S6	H87	0.315
	N188	0.324
	S261	0.140
S6	H87	0.008
S6	N188	0.008

Table 3.21. HBond Pairs for WT (PRRARS) FCD-Furin simulation.

Delta FCD	**Furin**	**Bond detection ratio**
R1	D152	0.309
	D157	0.097
	E123	0.085
	E150	0.036
	G122	0.012
	T155	0.533
R2	D157	1.412
	E129	0.891
	G158	0.006
	Y201	0.8
R3	E150	1.648
	G148	1.733
R5	A185	0.909
	D151	1.867
	D199	1.442
	S146	0.964
	S261	0.382
	D199	0.006
S6	H87	0.406
	N188	0.606
	S261	0.564
	T258	0.242
S6	H87	0.018
S6	S261	0.018
S6	N188	0.018

Table 3.22. HBond Pairs for Delta (RRRARS) FCD-Furin simulation.

BA1, BA2 FCD (HRRARS)	Furin	Bond detection ratio
H1	E123	0.05
	E150	0.007
	G122	0.021
R2	D157	1.773
	E129	0.05
	G158	0.085
	Y201	0.816
R3	D151	0.383
	E150	1.326
	G148	0.922
R5	A185	1
	D199	1.794
	P149	0.22
	S146	0.879
	S261	1.702
S6	H87	0.894
	N188	0.106
	S261	0.007

Table 3.23. HBond Pairs for BA1, BA2, FCD-Furin simulation.

BA1, BA2 Alternate FCD (KSHRRA)	Furin	Bond detection ratio
A6	N188	0.583
H3	E150	0.259
	G148	1.878
K1	D121	0.094
	E123	0.554
	E150	0.518
	G122	0.014
	V124	0.072
R4	D46	1.712
	H87	0.101
	S261	0.029
R5	A185	0.022
	D151	0.014
	D199	1.504
	S146	0.849
	S186	0.058
	S261	0.05
	T202	0.05
	T260	0.007
	Y201	0.007
S2	E129	0.108
	V124	0.014

Table 3.24. HBond Pairs for alternate BA1, BA2, FCD-Furin simulation.

empty	**NTD (1-300)**	**RBD (300-540)**	**FCD (679-685)**
Delta	T19R, G142D, E156G, Δ157-158	L452R, T478K	P681R
Omicron BA1	A67V, Δ69-70, T95I, G142D, N211I, Δ212, ins214R	G339D, R346K, S371L, S373P, S375F, K417N, N440K, G446S, S477N, T478K, E484A, Q493R, G496S, Q498R, N501Y, Y505H	N679K, P681H
Omicron BA2	T19I, L24S, Δ25-27, G142D, V213G	G339D, R346K, S371F, S373P, S375F, T376A, D405N, K417N, N440K, S477N, T478K, E484A, Q493R, Q498R, N501Y, Y505H	N679K, P681H

Table 3.25. Spike Protein Mutations relative to WT (Wuhan-Hu-1) in the N-terminal domain (NTD), receptor binding domain (RBD), and the Furin Cleavage Domain (FCD) [247].

Bound Pair	**WT**	**Delta**	**BA1**	**BA2**
RBD-ACE2	8.71 ± 2.03	10.57 ± 1.46	7.17 ± 1.48	7.08 ± 1.47
RBD-C12.B1A	19.23 ± 1.87	14.94 ± 2.61	11.27 ± 1.71	11.33 ± 1.81
RBD-P4A1	17.76 ± 2.01	17.61 ± 2.03	12.17 ± 2.45	14.06 ± 2.07
RBD-CR3022	12.58 ± 1.82	12.22 ± 1.59	10.81 ± 1.97	11.31 ± 1.45
NTD-4A8	9.62 ± 1.78	7.48 ± 1.49	4.84 ± 1.63	7.84 ± 2.47
FCD-Furin	10.73 ± 1.73	15.36 ± 1.88	12.05 ± 1.68	8.65 (Alt) ± 1.59

Table 3.26. Interfacial hydrogen bonds (with standard deviations) between proteins for WT, delta, and omicron.

Bound Pair	**WT**	**Delta**	**BA1**	**BA2**
RBD-ACE2	-67.42 ± 7.47	-72.10 ± 4.91	-73.97 ± 6.81	-78.64 ± 6.39
RBD-C12.B1A	-77.9 ± 8.00	-70.31 ± 8.41	-47.62 ± 5.78	-57.64 ± 6.29
RBD-P4A1	-115.91 ± 7.01	-110.21 ± 7.51	-73.71 ± 5.43	-106.62 ± 5.43
RBD-CR3022	-95.41 ± 2.51	-111.70 ± 7.80	-82.36 ± 10.26	-94.46 ± 9.38
NTD-4A8	-88.9 ± 17.9	-81.1 ± 5.7	-65.2 ± 12.2	-93.67 ± 6.33
FCD-Furin	-83.6 ± 8.4	-117.3 ± 4.8	-93.2 ± 3.8	-63.5 (Alt) ± 3.6

Table 3.27. MM/GBSA Binding Energies (with standard deviations) in kcal/mol between proteins for WT, delta, and omicron.

CHAPTER 4

Computational Study of the Furin Cleavage Domain of SARS-CoV-2: Delta Binds Strongest of Extant Variants

This chapter appears as a preprint at biorxiv [248] and is undergoing the peer review process at the time of writing. This work was done in collaboration with M. Zaki Jawaid, Sofia Jakovcevic, Jacob Lusk, Rustin Mahboubi-Ardakani, Nathan Solomon, Georgina Gonzalez, Javier Arsuaga, Mariel Vazquez, Richard L. Davis, and Daniel L. Cox.

4.1. Introduction

While the spike protein of the SARS-CoV-2 virus is similar to SARS-CoV-1, a key difference is a polybasic insertion beginning at P681 in the spike protein [249]. It has been shown that this insertion is critical to the higher transmissibility of SARS-CoV-2 [201,223] over SARS-CoV-1, and that the mutations P681H for the alpha and omicron variants and P681R for the delta variant play a large role in increased transmissibility of the variants over the wild type (WT) [202]. Similar polybasic furin cleavage domains (FCDs) occur in other human coronaviruses OC43, HUK1, 229E, MERS, and NL63 [250], and in many other viruses including H5N1 influenza [251].

The FCD of SARS-CoV-2 has not been well studied for at least two reasons. First, the FCD belongs to a rapidly fluctuating random coil region of the protein which has not been resolved by structural probes (see, e.g., Ref. [210], PDB structure 7A94, for which residues 677-688 are unresolved) . Second, because the furin rapidly cleaves the protein at this domain, there are no bound structures available. The absence of structural data has limited computational studies of the binding domain.

A number of small peptides that can act as furin inhibitors have been studied elsewhere. It is known that the four amino acid inhibitor RVKR, suitably terminated, is a potent inhibitor of furin activity [252]. Right handed hexa-arginine and nona-arginine peptides are potent inhibitors of furin also [253]. Additionally, the peptide Arg-Arg-Arg-Val-Arg-4-aminomethyl-benzamidine (RRRVR-Amba, I1 peptide) [246], is similar to the delta variant FCD RRRARS, and binds to furin with pM affinity. This leads to the conjecture that the FCDs of SARS-CoV-2 and other coronaviruses may bind in similar fashion to the furin enzyme. For SARS-CoV-2, the insertion that begins with P681 for the WT, alpha, delta, and omicron variants commences a six residue sequence (through 686) that hijacks the furin enzyme from its useful physiological functions to assisting the virus. We have focused on several six amino acid FCDs for SARS-CoV-2 and other viruses.

In the absence of structural data for the FCD, we turned to the deep learning based AlphaFold program [35]. We used AlphaFold Multimer [205], as implemented within the ColabFold environment [204], to generate candidate structures for the FCD-furin complexes. We find that AlphaFold accurately predicts the furin structure and the backbone of the bound furin-I1 structure (Fig 4.1A,B), so it is natural to attempt binding to the FCD, shown for WT in Fig 4.1C. We have used AlphaFold Multimer as the only way to generate a *de novo* structure for the WT FCD to furin binding. With this hypothesis, we can either generate *de novo* structures from AlphaFold Multimer, or assume the WT is well represented by the AlphaFold candidate structure, and mutations from that structure can be used to assess the binding of the FCDs for variants and other viruses. We simulate these structures with molecular dynamics to assess equilibrium binding strength, characterized by two quantities, interfacial hydrogen bonds between the furin and FCD (FCD-furin HBonds), and Generalized Born Surface Area (GBSA) binding energies. The AlphaFold approach reaches different conclusions about the FCD-furin bound structure than an earlier approach based upon docking [254]. That binding should determine cleavage rates is reasonable within a Michaelis-Menten analysis given that the cleavage sites are identical for most of the sequences (between R and S). We explain this in the supplemental material.

In the I1 sequence, the sixth (Amba) residue, a nonstandard amino acid, binds most strongly to furin as we discuss below. When we mutate that nonstandard residue to the structurally similar tyrosine, the binding pocket is occupied by the arginine at sequence position 5. Accordingly, we hypothesize that insertion of the residue at position 5 into the furin S1 pocket is the most important for FCD binding to furin. We confirm this hypothesis by simulating dozens of observed sequences. In 93% of observed SARS-CoV-2 FCD sequences starting from aligned position 681, the fifth amino acid is arginine.

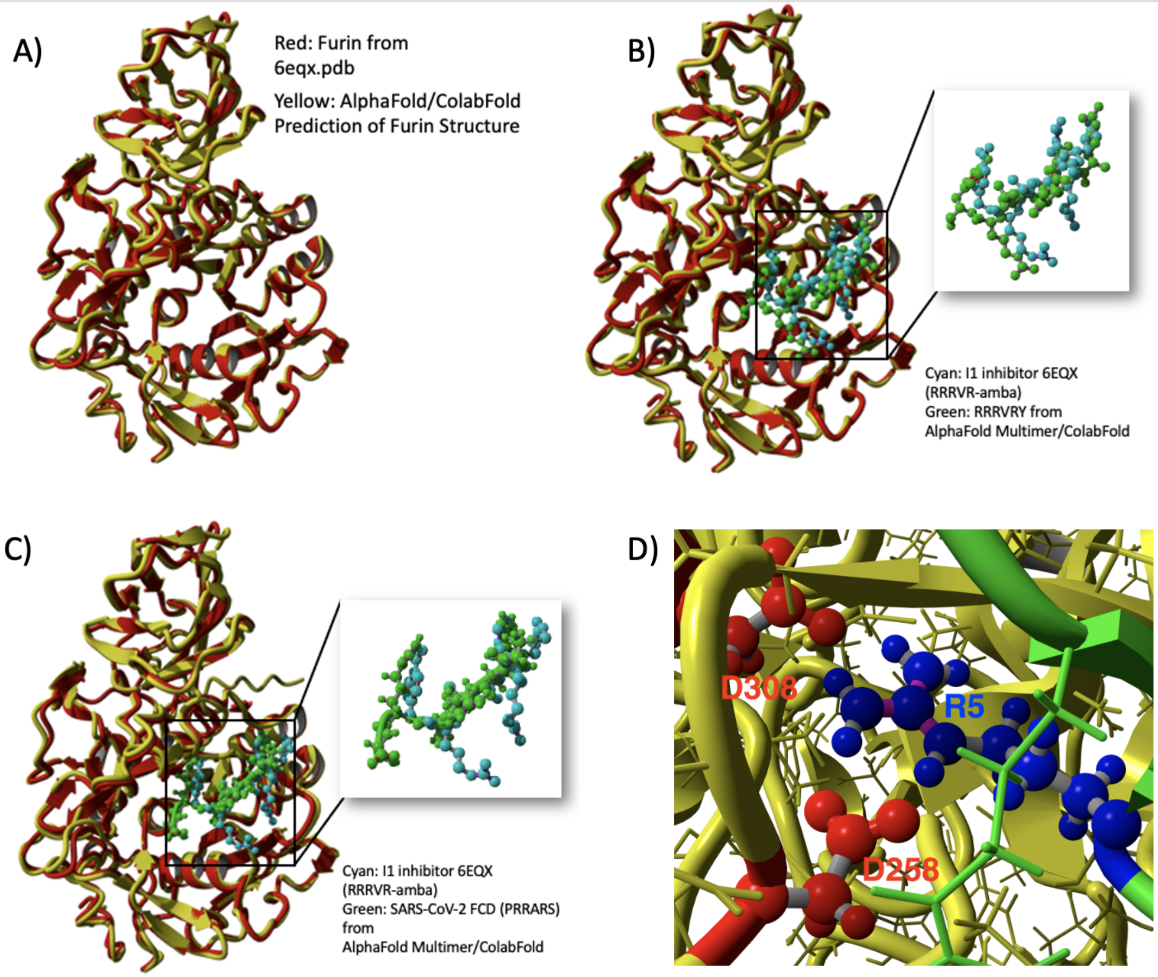

A)
Red: Furin from 6eqx.pdb
Yellow: AlphaFold/ColabFold Prediction of Furin Structure
B)
Cyan: I1 inhibitor 6EQX (RRRVR-amba)
Green: RRRVRY from AlphaFold Multimer/ColabFold
C)
Cyan: I1 inhibitor 6EQX (RRRVR-amba)
Green: SARS-CoV-2 FCD (PRRARS) from AlphaFold Multimer/ColabFold
D)
D308
R5
D258

FIGURE 4.1. A) Comparison of structure of furin from Ref. [246] and PDB file 6EQX with the structure from AlphaFold [35] using the ColabFold environment [204]. Clearly, the agreement is excellent (RMSD of 1.79Å). B) Comparison of structure of furin with RRRVR-Amba inhibitor from Ref. [246] with structure generated for the similar sequence RRRVRY by AlphaFold Multimer [205] using ColabFold [204]. The Amba is buried in the furin S1 pocket [246] for the inhibitor, while AlphaFold predicts burial of the R at position 5. The backbone RMSD between the I1 and RRRVRY peptides is 2.77Å . C) Predicted structure by AlphaFold Multimer [205] for the WT PRRARS sequence of SARS-CoV-2 compared to Furin-I1 structure. D) Close up of binding pocket for fifth residue of PRRARS (WT FCD). Furin backbone in yellow, FCD backbone in green, R5 from FCD is blue, D258,D306 from furin in red.

We obtain a number of important results. First, per Fig 4.2, the delta variant has the strongest binding of existing extant SARS-CoV-2 variants, and only two rare or unobserved FCD sequences bind as strongly within statistical accuracy. This dominance of the delta variant FCD extends to other coronaviruses and the H5N1 influenza virus. Second, as made clear in the heat map of Fig 4.3, the most important residue is the fifth, which binds in the S1 pocket of furin [246] containing two aspartic acid residues. In particular, this pocket matches structurally to arginine better than lysine as discussed below. Third, we find that there is mechanistic predictive power in three quantities that help explain the differences between delta and other variants and viruses: (1) the strength of the binding strongly correlates inversely with the root mean square fluctuation (RMSF) of the first residue. This suggests that the more the backbone outside the pocket fluctuates, the less likely the arginine at position 5 can bind well to the furin. (2) The number of FCD-furin hydrogen bonds between residue 5 and the furin strongly predicts the total binding strength, even though it only represents a plurality of the HBonds. (3) The maximum mean number of FCD-furin HBonds for a given number of basic residues peaks at 15.7 hydrogen bonds for 4.06 basic residues.

4.2. Results

To avoid confusion between the conventional N-to-C terminal sequence numbering of peptides and proteins vs. the reverse numbering used in the Furin Data Base (FurinDB) [255] and other references [246, 252, 253], we will refer to the FCD residues as positions 1-6, which for all viral sequences considered here will correspond to the FurinDB notation P5-P4-P3-P2-P1-P1', with the cleavage site between P1 and P1'. For example, in the WT SARS-CoV-2 FCD sequence PRRARS, the R at position 5 corresponds to P1, the S to position P1'. We will note the FurinDB identification parenthetically.

We first applied AlphaFold [35] through the ColabFold [204] environment to examine how well we could match the folded structure of furin. The result is shown in Fig 4.1A. The AlphaFold structure for furin matches that from the PDB entry 6EQX [246] to within a root mean square deviation of 1.79Å. Next, we included the furin inhibitor RRRVR-4-aminomethyl-benzamidine (RRRVR-Amba)

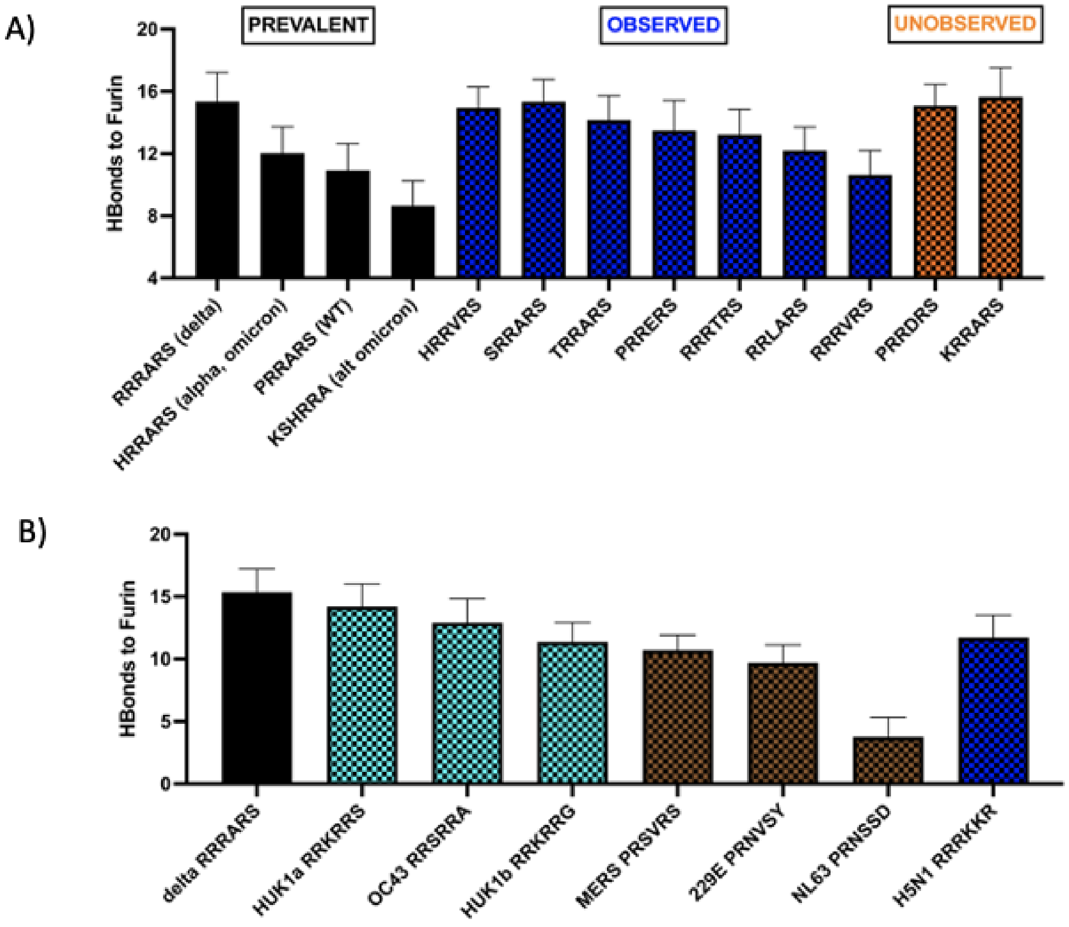


FIGURE 4.2. A) FCD-furin hydrogen bond counts between furin and SARS-CoV-2 binding sequences at 681-686 of the spike protein. The first four bars are prevalent forms (WT, delta, omicron/alpha, and alt omicron where we assume the sequence starts at position 679. The blue sequences are rare but observed in the GISAID [256] database; of these HRRARN and SRRARS bind as strongly to furin with in statistical accuracy as the delta sequence (RRRARS). The two unobserved sequences require double base mutations from existing extant codons, but bear watching because of their strong binding to furin. B) FCD-furin hydrogen bond counts between furin and other viruses. The SARS-CoV-2 delta variant shows the strongest binding of any human coronavirus and exceeds the H5N1 influenza cleavage site.

from Ref. [246] into the AlphaFold Multimer program [205], but because we could not enter the nonstandard residue Amba into the AlphaFold search we substituted tyrosine, which is similar to Amba away from the side chain terminus. As shown in Fig 4.1B, this produces a structure substantially similar to furin with bound RRRVR-Amba, except that the S1 pocket, which binds the position 6(P1) Amba nonstandard residue, accepts the position 5(P2) arginine for RRRVRY.

In essence, the Y for Amba substitution shifts the sequence to P5-P4-P3-P2-P1-P1'. The RMSD deviation of the RRRVR-Amba backbone from the RRRVRY backbone in the binding position is 2.77Å, which is relatively small and reasonable given the different placement of the Amba vs. arginine in the S1 pocket.

This sequence is very similar to the SARS-CoV-2 delta sequence commencing with arginine at 681, namely, RRRARS. It is known that the furin cleaves between the arginine at 685 and serine at 686. Hence, we hypothesize that the fifth residue(P1) enters the furin S1 pocket. When we utilize AlphaFold Multimer to explore the binding of the WT sequence (beginning at 681) PRRARS, we do find that the fifth arginine(P1) enters the furin S1 pocket, and binds strongly to two aspartic acid residues at positions 258 near the pocket entry, and 306 at the interior end of the pocket (Fig 4.1D). We note that the last arginine in the RVKR sequence of Ref. [252] also has close proximity to D258 and D306. The RMSD deviation of the RRRVR-Amba backbone from the PRRARS backbone of the FCD for SARS-CoV-2 from AlphaFold Multimer is 4.1Å. This is not surprising given the large sequence difference.

Arginine is particularly well suited for this binding, with its three side chain nitrogens in contrast to the single nitrogen in lysine. A lysine at the fifth position (P1) is only able to bind to the D306. Of all 62 observed sequences identified from GISAID for SARS-CoV-2, 58 have an arginine at position 5 (P1). For the other human coronaviruses, four (MERS, OC42, HUK1a,b) have an arginine at this position. NL63 and 229E have serines at this position, and the H5N1 flu has lysine.

There is also a strong bias towards a hydrophobic residue at position 4(P2) in the SARS-CoV-2 sequences. Alanine arises there in 46 of 62 sequences, and valine in 4 of 62. The alanine side chain carbon is within 5Å of side chain carbons on W147 and L120 from the furin in the delta structure. Of the other viruses, MERS and 229E have valine at position 4, while the others have arginine (OC43, HUK1a,b) or serine (NL63) at this position. The H5N1 flu has lysine at this position.

Given a starting structure, we can simulate and measure characteristics of the binding, such as counting FCD-furin HBonds, calculating the binding energy of the complex, or measuring the interfacial surface area, defined as half the difference between the solvent accessible surface area of the separated furin and FCD vs the solvent accessible surface area of the complex. We utilize the YASARA molecular modeling program [234], simulating each bound FCD-furin complex for at least 10 ns past energy minimization and equilibration. We then count interfacial protein hydrogen bonds using the criteria outlined for YASARA [241]. For computing the binding free energy, we use the Generalized Born Surface Area (GBSA) endpoint free energy calculation from the HawkDock server [245]. Because the binding interface is tight, there is essentially no water entry between the peptide and furin. As shown in the supplemental information, we obtain a strong correlation between the GBSA binding energy and the FCD-furin HBond count (Fig 4.5). For the rest of this paper, we shall use the FCD-furin HBond count as a proxy for binding strength. Note that in this approach, salt bridges of proximate residues, are effectively counted as H-bonds between basic side chain amide groups and acidic side chain carboxyl groups. Hence, the R685 residue of the spike protein FCD forms a salt bridge with the D306 residue of the furin protein, but this is counted in FCD-furin HBonds in this approach.

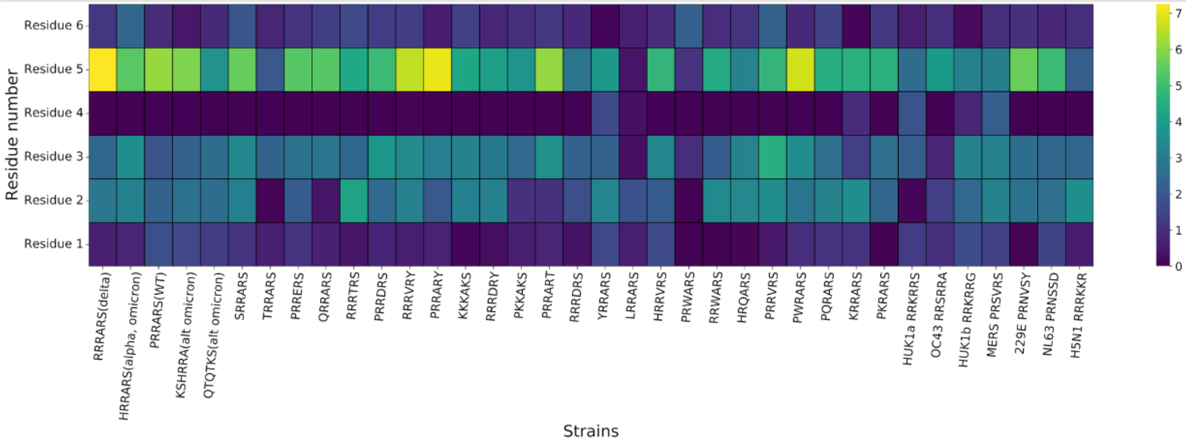


FIGURE 4.3. Heat map of interfacial hydrogen bonds from furin to the six residue peptide by residue number (vertical) for various observed SARS-CoV-2 along with two unobserved, and for other human coronaviruses and H5N1. Clearly, the key residue for binding is the fifth.

We have used AlphaFold as the only way to generate a candidate structure for the binding of the WT peptide to furin. With the other sequences we have a choice of using AlphaFold or using the mutation approach within YASARA. We generally find that there are small differences in favor of the mutation approach as we discuss in detail in the supplemental information (Fig 4.6).

We have surveyed a total of 62 observed six member SARS-CoV-2 furin FCD sequences at 681-686 for this paper drawn from from the GISAID database [256, 257, 258], out of which the delta sequence RRRARS is the top binder to within statistical significance. Those used in this paper are acknowledged in the Supplementary Materials. Fig 4.2A shows the FCD-furin HBond counts for the prevalent 681-686 sequences WT(PRRARS), delta (RRRARS), and omicron/alpha (HRRARS). We have also included KSHRRA as an alternate omicron sequence in view of the N679K mutation. Additionally, we include seven observed but rare sequences found from GISAID chosen either for their frequency of occurrence or their high FCD-furin HBond count. Finally, we include two sequences (PRRDRS and KRRARS) which can be arrived at by two base mutations from either the WT or delta variants. The prevalent codon at position 684 cannot swap by a single base to obtain D, and the prevalent codon at position 681 cannot swap by a single base to obtain K. By performing pairwise t-tests within GraphPad, we find that the FCD-furin HBond count for the delta variant sequence binding to furin exceeds all but one of the observed sequences with statistical significance ($p < 0.05$) or extreme ($p < 0.0001$) statistical significance, and differences are statistically insignificant in comparison to the observed SRRARS and unobserved PRRDRS and KRRARS sequences ($p > .1$ for each).

A similar picture emerges compared to other human coronaviruses and the H5N1 flu as shown in Fig 4.2B. The candidate sequences for OC43, NL63, HUK1a,b, 229E, and MERS were obtained by homology alignment of the spike proteins using BLAST [259]. The FCD-furin HBond count difference between the delta variant and these viral sequences is extremely significant ($p < 0.0001$). We note that the binding is strongest for the cold viruses HUK1a,b and OC43.

To assess the importance of the different residues in the six member peptide to binding strength, we analyzed the hydrogen bonding patterns in detail. We display a heat map in Fig 4.3 for many of the sequences shown in Fig 4.2. We find in nearly every case that the strongest binding, representing a significant plurality of the binding strength, is for the position 5(P1) residue, with arginine the preferred amino acid there. Notably, the H5N1 sequence with a K at position 5, and the trial sequence PKKAKS where all arginines are replaced by lysines, fare poorly at position 5(P1) compared to the other sequences.

In searching for an understanding for these observations, we have uncovered three correlations, two of which that can independently explain nearly 50% of the variation between SARS-CoV-2 sequences and separately between viruses. First, by examining the root mean square fluctuation of the backbone C-alpha of the first residue (P5), compared to the FCD-furin HBond counts of the observed sequences with at least 50 appearances in the GISAID tables for SARS-CoV-2, we see in Fig 4.4A that this backbone fluctuation correlates inversely with the binding strength with a linear regression coefficient of $R^2 = 0.53$. Fig 4.4B shows the correlation between the FCD-furin HBond count and the RMSF of the first alpha carbon (CA) atom for delta, the six other human coronaviruses with homology in this region, and the H5N1 flu virus. The linear regression coefficient is $r^2 = 0.49$. The best fit slope of -2.74±1.25 FCD-furin HBonds/Å is less than that for SARS-CoV-2 (-4.33± 1.54 FCD-furin Hbonds/Å), but the difference is statistically insignificant. Second, by examining the number of FCD-furin HBonds associated with the residue at position 5(P1), we observe (Fig 4.4C,D) that there is a high degree of correlation with the total FCD-furin HBond count. For the observed higher frequency furin binding sequences, the best fit slope is 1.41±.38 with $R^2 = 0.59$, and for comparison of delta to other viruses, the slope is similar 1.77±.51 with $R^2 = 0.66$. Third, as shown in Fig 4.4E, the number of basic residues (H,K, or R) in the six residue sequence helps determine the maximum number of FCD-furin HBonds. Fitting the maximum envelope of the plot to a quadratic, as in Fig 4.4F, gives

$$Hbonds = -0.66(N_B - 4.06)^2 + 15.7 \tag{4.1}$$

where N_B is the number of bases. The nonlinear regression coefficient is R^2=0.98. This suggests that the maximum number of FCD-furin HBonds is 16, for four basic residues (as per the delta variant), and to within statistical accuracy, no sequence exceeds delta in the number of FCD-furin HBonds to furin.

4.3. Discussion

The most important results of this paper are: 1) by using AlphaFold Multimer [205] we have validated by comparison to the binding of furin with a known six residue inhibitor, we are able to predict bound structures for over 60 observed FCD sequences of SARS-CoV-2 (at residues 681-686 of the spike protein and two alternate sequences for omicron) and eight other viruses (six human coronaviruses (OC43, HUK1a, HUK1b, MERS, NL63, 299E), the H5N1 influenza, and Epstein-Barr virus). From among these, the delta variant FCD of SARS-CoV-2 has the strongest binding to furin within statistical accuracy, with 15.3 mean FCD-furin hydrogen bonds. 2) Within these sequences we find selection for arginine at position 5 (P1), which fits into a furin S1 pocket having aspartic acids at the entrance and within. The structure of arginine allows binding to both aspartic acids, while lysine's structure does not.

3) There is also bias towards a hydrophobic residue at position 4(P2) of the six residue FCD, which appears to interface favorably with W147 and L120 of the furin. 4) We find that two features of the sequences each predict about half of the binding strength: (i) the backbone fluctuation of the first residue in the binding sequence correlates inversely with the overall binding strength as measured by FCD-furin HBonds, and (ii) the number of hydrogen bonds associated with the binding of residue 5(P1) in the furin S1 pocket correlates positively with FCD-furin HBond count. This residue never accounts for more than a plurality of the FCD-furin HBonds so it is somewhat surprising that it correlates with the observed trend of binding. (iii) By considering the variation of the FCD sequence with the number of basic residues, we conclude that no more than 16 FCD-furin HBonds are possible, and within statistical accuracy delta achieves the maximum value. We conjecture that the physical basis for this is a tradeoff between binding efficacy of the basic residues (especially arginine) and Coulomb repulsion as more are added.

In preparing this for submission, we noted an article which directly measured furin cleavage rates for 14 residue peptide designs and found that omicron was cleaved most rapidly [261]. We were not able to reproduce this result for the usual orientation of the peptides bound to the furin, but using AlphaFold we found that a reverse orientation was preferred for the 14 residue peptide, in which the P6-P5 residues take the place of the P1-P1' residues and vice versa. In this orientation, the omicron sequence is preferred over wild type and delta, while both wild type and delta are increased in stability with the N679K mutation. These results are presented in the supplementary materials.

In conclusion, we find that spike FCD-furin binding depends critically upon insertion of arginine in the fifth position (P1) of the FCD in a furin pocket that includes D258 at the opening and

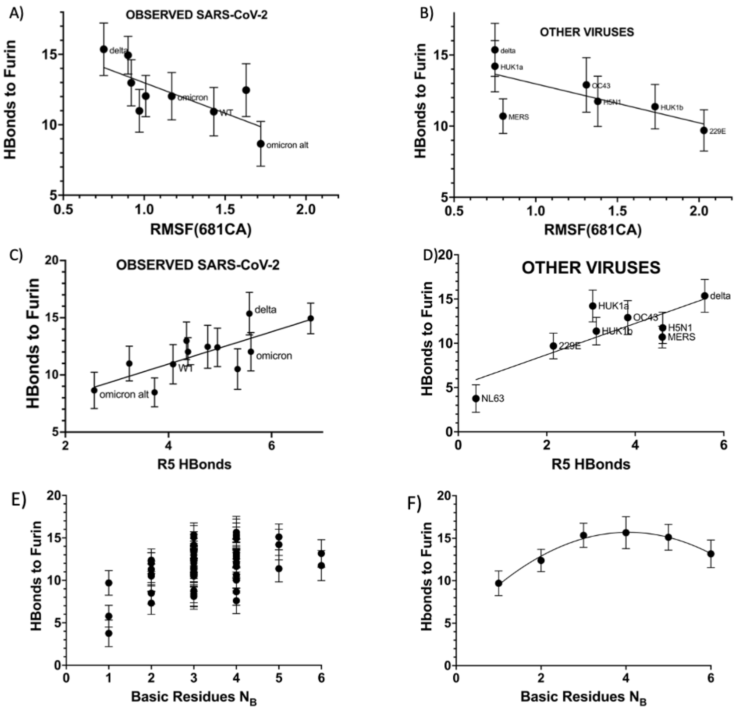


Figure 4.4. A) Correlation of the backbone fluctuation from Residue 1 of the sequence with the total number of FCD-furin HBonds between the binding sequence and furin for SARS-CoV-2 sequences observed at least 50 times. B) Correlation of the residue 1 backbone fluctuation with FCD-furin HBonds for delta and other viruses. C) Correlation of the interfacial HBonds for Residue 5 with the total number of HBonds for observed SARS-CoV-2 sequences of A). D) Correlation of the FCD-furin HBonds for Residue 5 with total number of FCD-furin HBonds for delta and other viruses. E) The number of HBonds for a given number of basic FCD residues plotted for 56 sequences. F) The maximum FCD-furin HBonds envelope as a function of the number of basic residues. This is fit with R^2=0.98 by Eq. (1) of the main text. The sequences for the peak values are for 1-6 respectively: PRNSVY (229E coronavirus), PRQARS (SARS-CoV-2), SRRARS (SARS-CoV-2), KRRARS (SARS-CoV-2, unobserved), RRRRRD (Epstein-Barr, ref. [260]), RRRRRR (unobserved).

D306 at the interior end. This prediction emerges uniquely from the application of AlphaFold Multimer [205] to predict the bound structure, and contrasts with earlier work that employed a docking program for interface prediction [254]. It is therefore critical to have experimental structural biology test of this prediction.

Note that the omicron FCD sequence is the same as alpha, and alternate FCD sequences (KSHRRA, beginning at K679, or QTQTKS, with K679 at position 5(P1)) have fewer FCD-furin HBonds than any observed variants, consistent with the observed milder impact of omicron on the lungs [195, 219, 220].

We conclude that it is quantitatively unlikely that any SARS-CoV-2 variant, or any other virus can bind significantly more strongly to the furin protease than the delta variant. This is based on a survey of a large number of observed SARS-CoV-2 spike sequences new SARS-CoV-2 spike sequences not yet observed, other human coronaviruses, H5N1 influenza, and Epstein-Barr virus. The basic model for viral infection is that after spike RBD binding to ACE2, furin cleavage at the FCD regulates fusogenicity leading to syncytia and viral reproduction. Our theoretical studies indicate that furin-FCD driven fusogenicity is at its worst with the delta variant among all observed SARS-CoV-2 variants of interest or concern. Of concern and cause for caution are some rarely observed or unobserved FCD sequences which could be just as consequential for furin cleavage as delta (observed: SRRARS, RRRARN,HRRVRS; unobserved: PRRDRS, KRRARS).

Data Availability

The pdb files used/generated in this study are available in Google Drive, at the link: PDB Files.

A summary of all available hydrogen bond data and GBSA calculations is in an excel file available in Google Drive, at the link FurinPaper - AvailableData.xlsx.

YASARA simulation files and AlphaFold result files for all the results presented here are available upon reasonable request to the corresponding author.

Acknowledgments

We acknowledge fruitful discussions with Victor Muñoz and Gary Whittaker. J.A., M. V, G.G. were partially supported by NSF grant DMS-2030491, G.G. was supported by a grant from the UC

Davis Center for Data Science and Artificial Intelligence and by a donation from Protein Architects to J.A and M.V., S. J. was supported by a Md. Abdur Razzaq Global HealthShare Undergraduate Research Fellowship.

Competing Interests

D.C. and R.D. are officers of Protein Architects Corp. Protein Architects has no commercial or IP interests in this work. G.G. received support from Protein Architects via a gift to the research program of J.A. and M.V. No other authors have competing interests.

4.4. Methods

4.4.1. Molecular Models

For the furin structure and for furin binding to the inhibitor RRRVR-Amba, we used PDB entry 6EQX [246].

4.4.2. Sequence Alignment for Other Coronaviruses

To identify homology in the furin cleavage domain for the other coronaviruses OC43, NL63, 229E, MERS, HUK1a, HUK1b, we utilized BLAST [259] at the National Center for Biotechnology Information. We compared entire spike sequences and zoomed in on the furin cleavage domain based upon the PRRARS sequence for SARS-CoV-2.

4.4.3. Genomic Data Set and Sequence Pre-Processing

We obtained SARS-CoV-2 sequences for this study from the GISAID database on Nov 11, 2020 [262]. Our data set contains FASTA files for every complete human SARS-CoV-2 nucleotide sequence (from all geographical locations) available in GISAID between and 12/1/19 and 7/11/2021. The sequences were then aligned using `ClustalOmega` with the default parameters [263]. We found that `ClustalOmega` ran faster on our data set than common alternatives like `ClustalW` [264] and `MUSCLE` [265].

After aligning the sequences, we extracted the spike protein by comparing the aligned sequence with the NCBI's SARS-CoV-2 reference sequence (NC_045512.2; "WT") [266] and tabulated the frequencies of different furin binding domain inserts.

The accession numbers and acknowledgments for the first of each 111 unique nucleotide sequences referenced in this paper as they appear in GISAID are provided in the Supplementary Materials.

4.4.4. Molecular Dynamics

To simulate the protein-protein interactions, we used the molecular-modelling package `YASARA` [234] to substitute individual residues and to search for minimum-energy conformations on the resulting modified structures of the FCD-furin. For all of the structures, we carried out an energy-minimization (EM) routine, which includes steepest descent and simulated annealing (until free energy stabilizes to within 50 J/mol) minimization to remove clashes. All molecular-dynamics simulations were run using the AMBER14 force field with [235] for solute, GAFF2 [236] and AM1BCC [237] for ligands, and TIP3P for water. The cutoff was 8 Å for Van der Waals forces (AMBER's default value [238]) and no cutoff was applied for electrostatic forces (using the Particle Mesh Ewald algorithm [239]). The equations of motion were integrated with a multiple timestep of 1.25 fs for bonded interactions and 2.5 fs for non-bonded interactions at $T = 298$ K and $P = 1$ atm (NPT ensemble) via algorithms described in [240]. Prior to counting the FCD-furin hydrogen bonds and calculating the free energy, we carry out several pre-processing steps on the structure including an optimization of the hydrogen-bonding network [241] to increase the solute stability and a pK_a prediction to fine-tune the protonation states of protein residues at the chosen pH of 7.4 [240]. Insertions and mutations were carried out using `YASARA`'s BuildLoop and SwapRes commands [240] respectively. Simulation data was collected every 100ps after 1-2ns of equilibration time, as determined by the solute root mean square deviations (RMSDs) from the starting structure. For all bound structures, we ran for at least 10 ns post equilibrium, and verified stability of time series for FCD-furin hydrogen bond counts and root mean square deviation (RMSD) from these starting structure. Because of concerns about the validity of short time simulations, and more variability for the weaker binding for the omicron RBD-ACE2 complex, we ran for 40 ns postequilibration in that case.

The FCD-furin hydrogen bond (HBond) counts were tabulated using a distance and angle approximation between donor and acceptor atoms as described in [241]. Note that in this approach, salt bridges of proximate residues, are effectively counted as H-bonds between basic side chain amide groups and acidic side chain carboxyl groups. Hence, the R685 residue of the spike protein FCD forms a salt bridge with the D306 residue of the furin protein, but this is counted in HBonds in this approach.

Note that in view of the likely ambient pH for cell surface or endosomal furin cleavage, and the polybasic environment of the FCD, we have assumed all histidines to be singly protonated at the delta site. Choosing the epsilon site makes little difference. For the alpha sequence, doubly protonated histidine binds more strongly, but for the alternate omicron sequence, there is little difference among the three protonation states.

4.4.5. Endpoint Free Energy Analysis

We calculated binding free energy for the energy-minimized structure using the molecular mechanics/generalized Born surface area (MM/GBSA) method [242, 243, 244], which is implemented by the `HawkDock` server [245]. While the MM/GBSA approximations overestimate the magnitude of binding free energy relative to *in-vitro* methods, the obtained values correlate well with H-bond counts. For each RBD-ACE2, RBD-AB, and NTD-Ab binding pair we average over five snapshots of equilibrium conformations. For each FCD-furin pair, we average over ten snapshots of equilibrium conformations.

4.4.6. Use of ColabFold/AlphaFold for Furin binding domain

Full details of this method are provided in [35, 204, 205]. In brief, we used the heterocomplex prediction method known as AlphaFold-Multimer [35,205] as implemented within ColabFold [204] to predict the best bound structure to the furin enzyme of the six residue FCD from the WT protein. We inferred the ordering of this sequence by comparison with a very similar six residue peptide inhibitor of furin with the sequence RRRVR-aminomethyl-benzamidine (RRRVR-Amba) [246]. In this case the backbone of the WT FCD aligns well with that of the inhibitor, but the arginine at residue 5 enters the furin S1 pocket [246] while the Amba enters the furin S1 pocket for the inhibitor. The serine is in proper cleavage position for furin. Most other structures were then obtained by mutation from the predicted WT FCD-furin structure.

4.4.7. Statistical Analysis and Graphics

We computed the statistical significance of pairwise differences using the GraphPad unpaired t-test calculator. Regression analysis for Fig 4.4, Fig 4.5 was carried out using the GraphPad Prism (v. 9) package. Structural images for Fig 4.1 were created in YASARA. Fig 4.2, 4.4, 4.5, 4.6 were created with GraphPad Prism (v. 9). Fig 4.2 was created with Seaborn (v. 0.11.2), a Python data visualization library.

4.5. Supplementary Materials

Correlation of FCD-furin HBonds and MM/GBSA binding energy estimates

Fig 4.5 summarizes the correlation between FCD-furin HBonds and the MM/GBSA estimates for binding energy in kcal/mole. Note that MM/GBSA usually overestimates binding energy strength significantly but is good for producing binding energy trends. The regression coefficient is $R^2 = 0.61$, and the best fit slope is -.107 Hbonds/(Kcal/mole) with 95% confidence intervals of -.110 to -.1042. Clearly, the correlation is strong between FCD-furin HBonds and binding energy.

Differences between simulations with AlphaFold and mutation

While we must take the WT FCD-furin structure from AlphaFold [35,205], we can mutate using the Swap command in YASARA from there to obtain other starting structures for molecular dynamics simulations. In general, AlphaFold produces structures with slightly less binding strength than mutating from the WT, with a few exceptions, the delta variant being one. This is demonstrated for five sequences in Fig 4.6. Accordingly, because the resultant binding is stronger we have used the mutant results where possible to provide a more accurate starting point for the equilibration runs in molecular dynamics.

Examples of sequence frequency and codons

Fig 4.7 shows a table of FCD sequences used in the figures as well as one synonymous/silent mutation based upon the consensus codons for WT, alpha, and delta. The last two entries are for unobserved but potentially potent FCD sequences. Mutation to those would require two base swaps from either WT or delta.

Michaelis-Menten Analysis and argument that binding dominates cleavage rates in the low concentration regime

The following reaction scheme applies to the furin cleavage process, where P is the binding peptide, F is the furin enzyme, and $P1, P2$ are the cleavage products:

$$P + F \quad \underset{k_-}{\overset{k_+}{\rightleftharpoons}} \quad PF \quad \xrightarrow{k_c} \quad P1 + P2 + F \tag{4.2}$$

If, per usual, we assume steady state for the bound complex, it is straightforward to show that the rate of cleavage V_c is given by

$$V_c = k_c[F]_0[P]/(K_M + [P]) \tag{4.3}$$

with $[F]_0$ the initial furin concentration, $[P]$ the concentration of furin binding polymer, and

$$K_M = \frac{k_- + k_c}{k_+} \ . \tag{4.4}$$

In the low concentration limit of P, $V_c \approx k_c[F]_0/K_M$. If $k_c >> k_-$, then $V_c \approx k_+[F]_0$, and since k_+ depends upon the binding strength of P to F, the rate varies with the binding. If $k_c << k_-$, then $V_c \approx k_c[F]_0/K_D$ where K_D is the dissociation constant of P from F. Assuming k_c independent of P, then again we conclude that the binding strength determines the cleavage rate.

Analysis of length 14 residue peptide cleavage data

In Ref. [261], length 14 amino acid peptides with donor and acceptor fluorophores at the ends have been used to study furin cleavage rates. The quenching of FRET signal between the fluorophores is a proxy for cleavage.

This work found that among the original WT variant, delta, and omicron variants, that omicron bound the strongest. Moreover, they found that the N679K mutation is critical, and that binding of the WT and delta variants with this mutation leads to enhanced cleavage rates. This is in clear conflict with predictions of this study on binding of six residue peptides.

Because the possibility exists that the flanking regions can affect the binding of the 14 residue peptides to the furin cleavage region, possibly altering the relative stability, we carried out AlphaFold and simulation studies on the peptides shown in Table 4.1.

We found that four the 14 residue peptides, AlphaFold favored a 180^o reverse of the binding to the furin, with P6-P1' being inverted to P1'-P6.

This is shown in Fig 4.8, compared with the conventional P6-P1' binding of the six residue peptides.

When we simulate the WT, delta, and omicron variants in the original orientation (P6-P1') we find that the hydrogen bond counts track the six residue peptides with delta binding the strongest and WT and omicron being indistinguishable statistically, as shown in Fig 4.9a. In each case, approximately 3 interfacial hydrogen bonds are contributed from the flanking regions.

When we run the peptides in the reverse conformations, WT and delta bind more weakly than omicron, which binds more strongly than in the original conformation by approximately two interfacial hydrogen bonds, as shown in Fig 4.9b. Additionally, the binding strength of WT and delta variants in this reversed orientation are enhanced by the N679K mutation present in omicron, though not above omicron in contrast to the experimental cleavage rate measurements.

Besides the potential for reversal of the synthetic peptides with flanking regions, we note that the binding of the P6-P1' sequence in the actual protein can be very different than the 14 residue peptides, because of the constraint to connect back to the full spike protein.

In Fig 4.10, we show the results of grafting the furin cleavage domain to the structure of Ref. [267]. Because the furin cleavage domain is not resolved structurally due to fluctuations, we grafted the loop on with the bridge protocol of YASARA. The P5-P1' sequence of the 14 residue peptides as bound to furin (not shown) and of the *in situ* furin cleavage domain P5-P1' sequence are highlighed in magenta. It is clear that the constraint of attaching to the full protein denies the adaption of the flanking regions to enhance the furin binding.

We conclude that (i) there is a high probability the 14 residue peptides reverse orientation relative to the usual one in binding to furin, particularly for omicron, and (ii) that because of the *in situ* constraints on the flanking regions, the smaller 6 residue peptides are likely to be more realistic indicators of binding efficacy than observations on the 14 residue peptides.

Supplementary Figures and Tables

A table of the accession numbers and acknowledgments for the first of each 111 unique nucleotide sequences referenced in this paper as they appear in GISAID can be found here.

TABLE 4.1. 14 residue peptides to compare with cleavage data

Peptide	Sequence
WT	TQTNSPRRARSVAK
delta	TQTNSRRRARSVAK
omicron	TQTKSHRRARSVAK
WT N679K	TQTKSPRRARSVAK
detal N679K	TQTKSRRRARSVAK

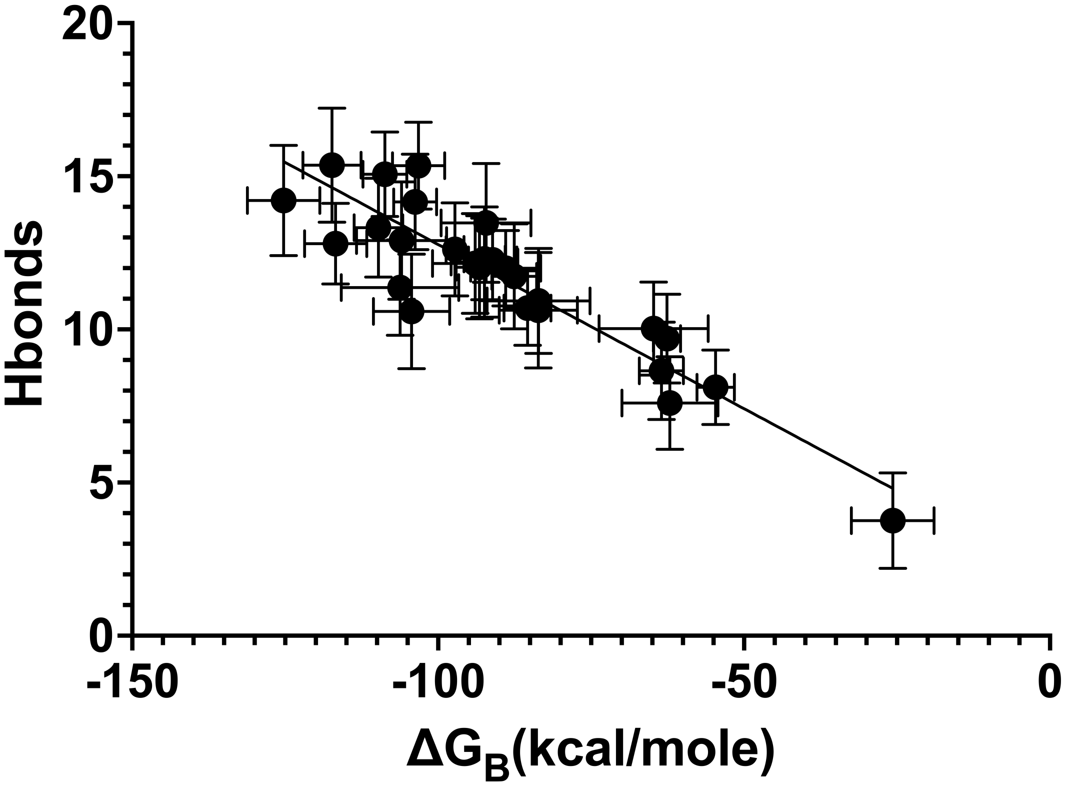


FIGURE 4.5. **Correlation of FCD-furin HBond counts with MM/GBSA Binding Energy** FCD-furin HBond counts are estimates from YASARA [234,241] simulations, while MM/GBSA Binding Energy comes from the HawkDock server [245]. Regression analysis using GraphPad Prism 9 provides the straight line fit (see text for details).

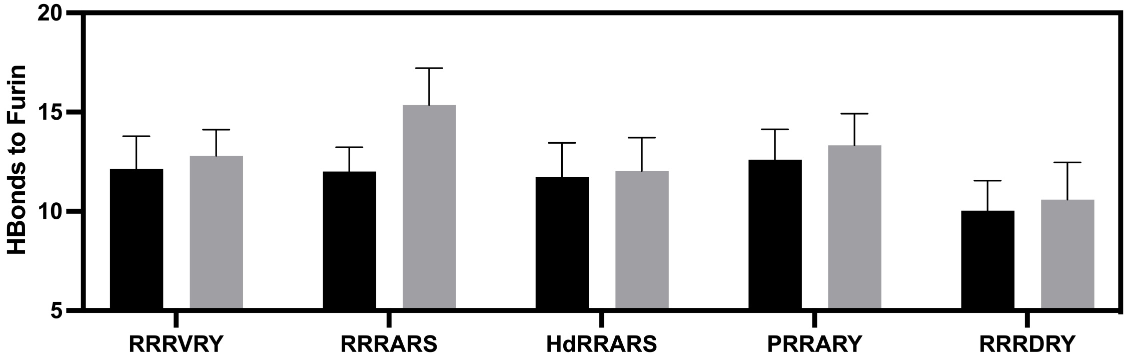
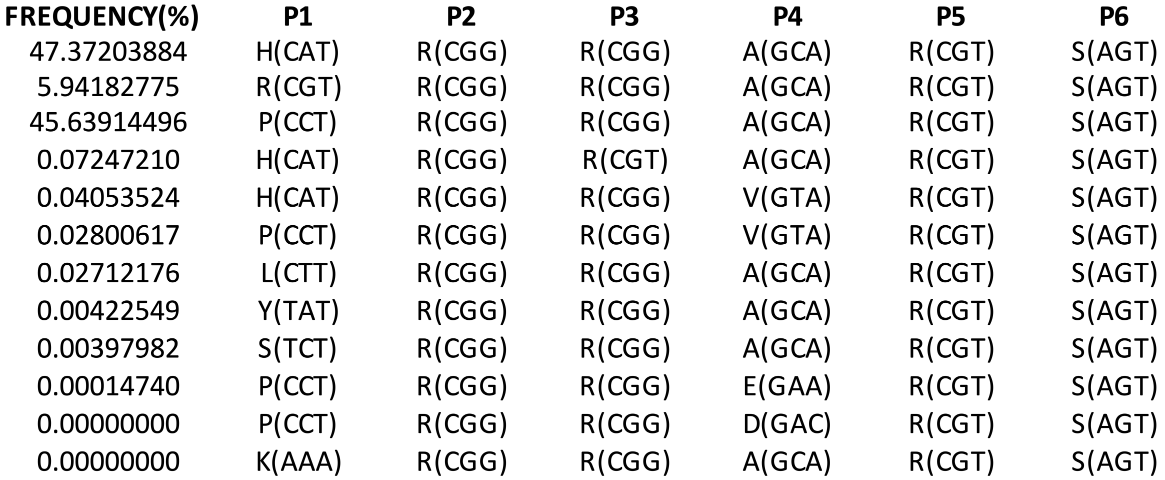


Figure 4.6. **Differences in equilibrated FCD-furin HBond counts between AlphaFold generated starting FCD-furin structures and starting structures mutated from the AlphaFold WT structure** In general, after equilibration the AlphaFold structures have slightly less binding strength, with a few exceptions such as the delta variant where AlphaFold misses dramatically. For comparison, the p-values for AlphaFold vs mutant in this plot are RRRVRY- p=0.0035 (very significant); RRRARS (delta)- p¡0.000001 (extremely signficant); HRRARS (alpha/omicron)- p=0.181 (not significant); PRRARY - p=0.00094 (very significant); RRRDRY - p=0.0164 (very significant)



FREQUENCY(%)	P1	P2	P3	P4	P5	P6
47.37203884	H(CAT)	R(CGG)	R(CGG)	A(GCA)	R(CGT)	S(AGT)
5.94182775	R(CGT)	R(CGG)	R(CGG)	A(GCA)	R(CGT)	S(AGT)
45.63914496	P(CCT)	R(CGG)	R(CGG)	A(GCA)	R(CGT)	S(AGT)
0.07247210	H(CAT)	R(CGG)	R(CGT)	A(GCA)	R(CGT)	S(AGT)
0.04053524	H(CAT)	R(CGG)	R(CGG)	V(GTA)	R(CGT)	S(AGT)
0.02800617	P(CCT)	R(CGG)	R(CGG)	V(GTA)	R(CGT)	S(AGT)
0.02712176	L(CTT)	R(CGG)	R(CGG)	A(GCA)	R(CGT)	S(AGT)
0.00422549	Y(TAT)	R(CGG)	R(CGG)	A(GCA)	R(CGT)	S(AGT)
0.00397982	S(TCT)	R(CGG)	R(CGG)	A(GCA)	R(CGT)	S(AGT)
0.00014740	P(CCT)	R(CGG)	R(CGG)	E(GAA)	R(CGT)	S(AGT)
0.00000000	P(CCT)	R(CGG)	R(CGG)	D(GAC)	R(CGT)	S(AGT)
0.00000000	K(AAA)	R(CGG)	R(CGG)	A(GCA)	R(CGT)	S(AGT)

Figure 4.7. **Examples of FCD sequences from GISAID for analysis here with codons** The observed frequencies of sequences between 12/1/19 and 7/11/21 appear at left, and the predominant codons for each position are tabulated. Row 4 shows a synonymous/silent mutation to the alpha variant, while the rest show missense mutations. The last two sequences are unobserved (requiring double codon swaps relative to either WT or delta) but bind as strongly to furin as the delta FCD. Note that over this entire pre-omicron time frame that delta (RRRARS) has less accumulated percentage of the sequences than WT (PRRARS) or alpha (HRRARS).

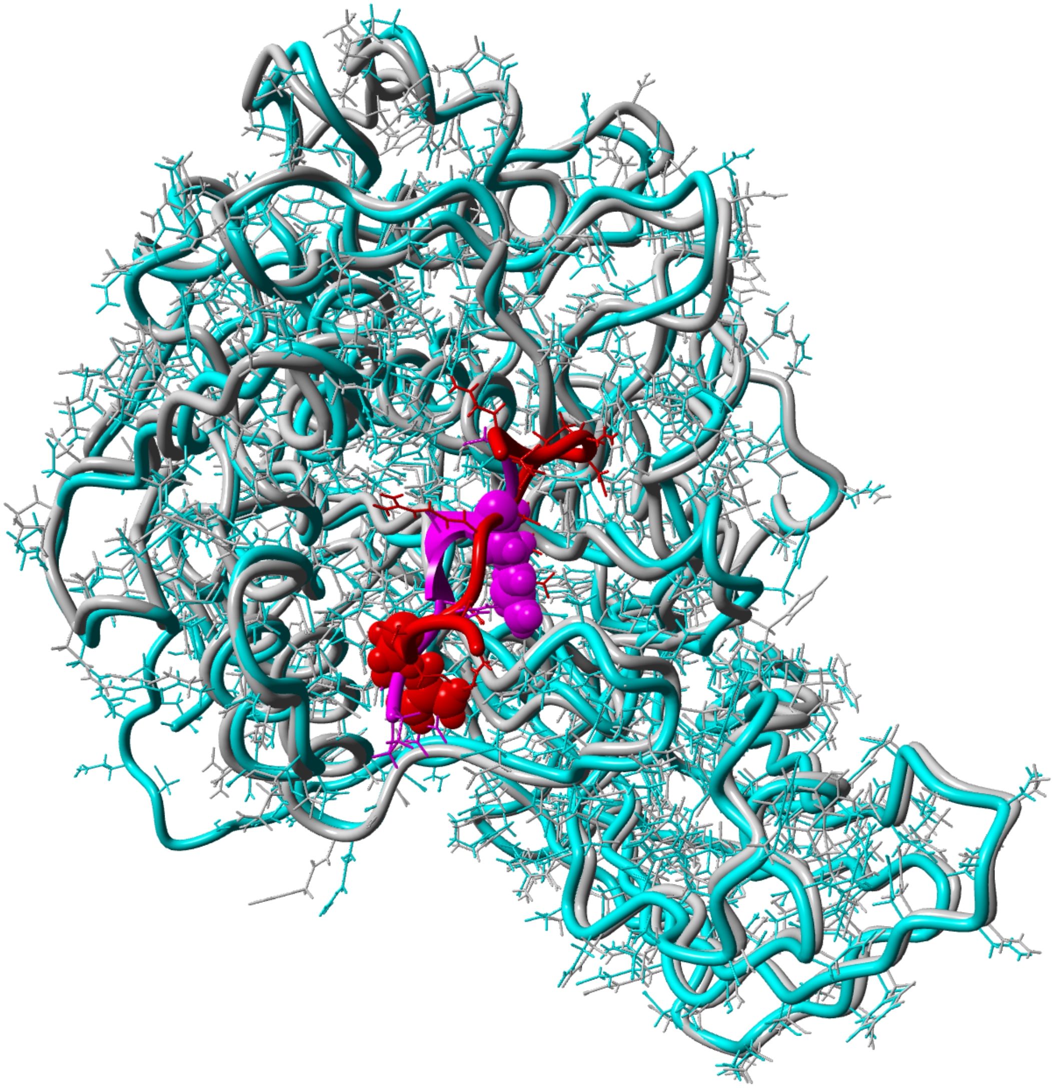


FIGURE 4.8. **Structure of peptides bound to furin** Furin (gray) bound to six residue peptide (magenta), Furin (cyan) bound to 14 residue peptide in reverse conformation (red). The P1 arginines for each peptide are shown in van der Waals sphere format. Image prepared by YASARA.

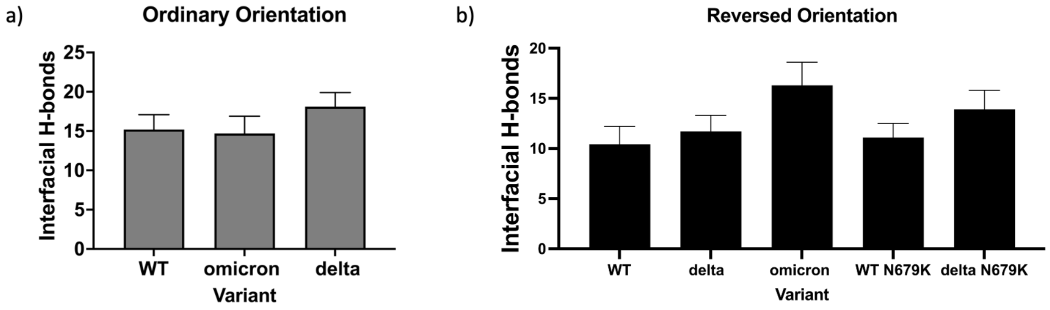


FIGURE 4.9. **Interfacial hydrogen bond counts of 14 residue peptides bound to furin** a) Results for ordinary P6-P1' orientation. As for the six residue peptides, the furin binds more strongly to delta, with statistical insignificance in bond counts between omicron and WT. b) Results for reversed P6-P1' orientation suggested by AlphaFold. In this orientation, omicron binds the strongest, and binds stronger than omicron in the normal orientation. The N679K label refers to mutation of N679 to K for the WT and delta sequences. For each of WT and delta, the binding is enhanced by this mutation, though the binding for WT and delta is stronger in the normal orientation.

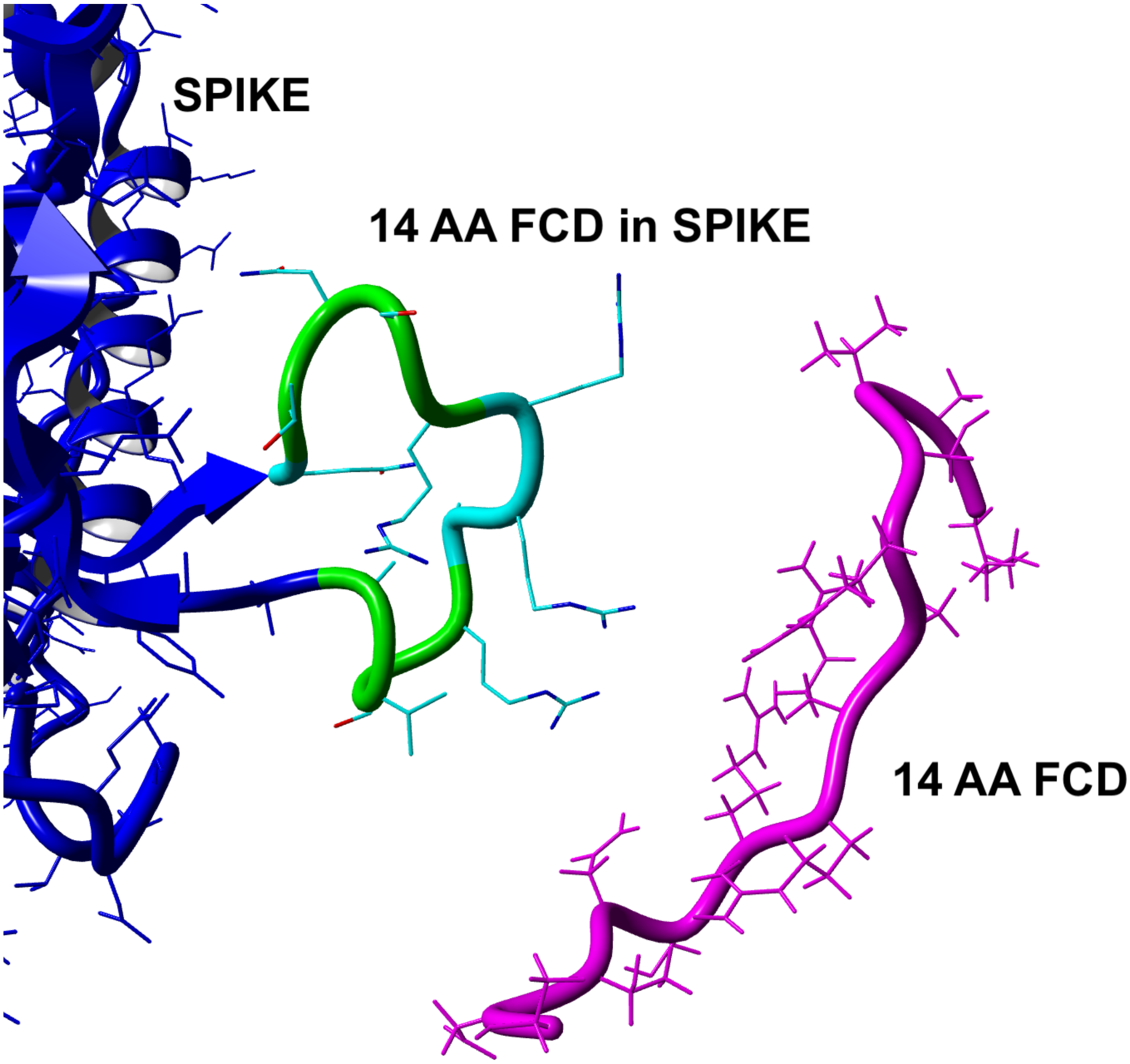


FIGURE 4.10. **Structure of 14 residue FCD in the spike protein vs. 14 residue peptide** The SARS-CoV-2 spike protein is in blue. The 14 residue FCD is shown in green as attached using the bridge feature of YASARA. The 14 residue peptide mimicking the FCD as it would appear bound to the furin enzyme from molecular dynamics simulations is shown in magenta. Clearly the constraint of attaching the 14 residue FCD to the protein gives less conformational flexibility in the binding region than when the ends are free.

CHAPTER 5

Combining Different V1 Brain Model Variants to Improve Robustness to Image Corruptions in CNNs

This chapter appears as an article [268] published in NeurIPS 2021 in the Shared Visual Representations in Human and Machine Intelligence (SVRHM) workshop. This work was done in collaboration with Joel Dapello, James J. DiCarlo, and Tiago Marques.

5.1. Introduction

Recently, convolutional neural networks (CNNs) have not only dominated several computer vision applications [269, 270, 271] but have also surpassed human visual abilities in specific domains such as object classification [272]. However, unlike humans, CNNs show a striking lack of robustness: they are vulnerable to small perturbations optimized to fool them (adversarial attacks) [36, 37, 38]; and, perhaps more relevant for real-world applications, they struggle to recognize objects in images corrupted with common noise patterns [39, 40, 41]. These two perturbation types expose different aspects of robustness: models designed to better withstand one usually fail to generalize to the other [44, 51].

Recently, Dapello, Marques et al. observed that models that were more robust to adversarial attacks had early stages that better predicted neuronal responses in the macaque primary visual cortex (V1) [51]. Inspired by this, the authors developed a novel hybrid CNN, containing a model of V1 as the front-end, followed by a trainable standard architecture back-end. This new model, the VOneNet, was substantially more robust to adversarial attacks than the corresponding base-models and rivaled more computationally expensive methods such as adversarial training. Surprisingly, VOneNet models also showed small gains in robustness to common corruptions, with different variants of the V1 front-end leading to specific trade-offs in accuracy when considering all the corruption types.

Here, we extend this last finding to make the following novel contributions. First, we adapt the VOneNet model to the Tiny ImageNet dataset [273] and reproduce results from Dapello, Marques et al., particularly the existence of specific trade-offs in dealing with common corruptions for several variants. Then, we build a new model using an ensembling technique which combines VOneNet models with different V1 front-end variants, eliminating trade-offs and showing a remarkable improvement in robustness to common corruptions (38% overall). Finally, we show that distillation training is able to partially compress the knowledge in the ensemble into a single VOneNet model, resulting in a compact architecture that improves over the baseline on all the corruption categories (13% overall). Together, these results, demonstrate that by combining the specific strengths of different neuronal populations in V1 it is possible to improve the robustness of CNNs.

Related Work

Common corruptions Several recent works have studied the robustness of CNNs against common corruptions [41, 42, 43, 44, 45, 46, 47, 48, 49, 50]. The current state-of-the-art for common image corruptions (DeepAugment+AugMix) [43] involves using an image-to-image network to add perturbations to the input image combined with a technique that mixes randomly generated augmentations. Other data augmentation techniques have also been shown to improve robustness. [44] showed that augmentation with Gaussian noise or adversarial noise can significantly improve model robustness. [45] apply Gaussian noise to small image patches to improve robustness. Gaussian data augmentation, however, can impact clean image performance [44] and can cause models to be vulnerable to low frequency corruptions [46]. [47] assemble common CNN techniques, including knowledge distillation, into a single CNN to achieve improved performance on clean and corrupted images. Other techniques to increase robustness involve using: anti-aliasing module to restore the shift-equivariance [48], stylized images to increase shape bias [49] and stability training [50].

Biologically-inspired methods for improving robustness [51] showed that simulating V1 in front of CNNs can substantially improve white box adversarial robustness with smaller gains in the case of common corruptions. In a similar work, [52] replaced the first convolutional layer of a standard CNN by Gabor filters to improve robustness to noise. Other biologically-inspired works to improve robustness include: regularizing CNN models' representations to approximate mouse V1 [54] and training to predict neural activity in V1 while performing image classification [55].

Ensemble and distillation Ensembling is a well-known machine learning technique to combine smaller individual models into a larger model leading to superior performance compared to the individual models in a diverse range of supervised learning problems [62,63,64,65,66,67], including generalization to out-of-domain (OOD) datasets [274, 275, 276, 277, 278]. Knowledge Distillation is a popular technique [68, 279, 280, 281, 282, 283, 284] to transfer the superior performance of the ensemble (teacher) into a single smaller model (student). This technique has been shown to improve the generalization ability of the student models [47, 281, 283, 285].

5.2. Results

5.2.1. V1 model variants show performance trade-offs on different corruptions

VOneNets are CNNs with a biologically-constrained fixed-weight front-end that simulates V1 (the VOneBlock) followed by a conventional neural network architecture [51]. The VOneBlock consists of a fixed-weight Gabor filter bank (GFB) [53], simple and complex cell [57] nonlinearities, and neuronal stochasticity (Fig. 5.1A) [58]. Here, we trained a VOneNet model for object classification on the Tiny ImageNet dataset [273] using the ResNet18 [272] as the back-end architecture, which we call the standard VOneResNet18 model. In addition, we created seven model variants by removing or modifying one of the VOneBlock components (Fig. 5.1B, Table 5.1, Section 5.4.2.1). All the variants' back-ends, including the standard model, were optimized from scratch on Tiny ImageNet following an identical training procedure (Section 5.4.3). We evaluated model robustness using the Tiny ImageNet-C dataset [41] which consists of 15 different corruption types, each at five levels of severity, divided into four categories: digital, weather, blur and noise (Section 5.4.1.2).

While all of the model variants were found to perform worse than ResNet18 on clean Tiny ImageNet images, all of them were considerably more robust on at least one corruption category (Fig. 5.1C, Table 5.2). Still, no single variant outperformed ResNet18 in all four categories of image corruptions. For example, the standard VOneResNet18 and Low SF variant were more robust than ResNet18 to blur and noise corruptions but they performed worse on digital and weather corruptions. Similarly, Mid SF and Only Simple variants were more robust than ResNet18 to all corruptions except for weather corruptions. Furthermore, some model variants that outperformed other variants in all four categories of corruptions performed worse on clean images. These results show that while some variants of the VOneBlock lead to large gains in robustness for specific corruption types, this comes with losses for others. This type of trade-off is present for all the variants analyzed (Fig. 5.1C, Table 5.2).

5.2.2. Ensemble of different VOneNet variants eliminates robustness trade-offs

We used a common ensembling technique of uniformly averaging the outputs (logits) of the VOneNet variants described in the previous section to create the Variants Ensemble. The Variants Ensemble not only outperformed all the variants but also performed on par with ResNet18 on clean images (Fig. 5.2, Table 5.3). Remarkably, the Variants Ensemble was found to be substantially more robust than ResNet18 in all corruption categories (and in 13 out of 15 individual corruption types, Fig. 5.5), showing that ensembling is able to combine the diverse strengths of the individual variants. As a result, we developed a model that considerably outperforms ResNet18 in all corruption categories (between 17% and 60% with 38% overall) without compromising on clean accuracy.

While diversity in the members of an ensemble has been found to be important to its generalization ability [274, 277, 286, 287], ensembles of networks with the same architecture that only differ in their random initialization also improve robustness [275,276]. To test if the diversity in the variants was critical for the observed gains, we created two Seeds Ensembles by combining eight different seeds of the standard VOneResNet18 and of the ResNet18. The Variants Ensemble consistently outperformed the other ensemble models on all the corruption categories (Fig. 5.2, Table 5.3). We also compared Variants Ensemble to a popular defense method that uses Gaussian Noise Training (GNT) as data augmentation [44]. We trained both ResNet18 (ResNet18-GNT) and standard VOneResNet18 (VOneResNet18-GNT) with GNT, observing an increased robustness for noise and

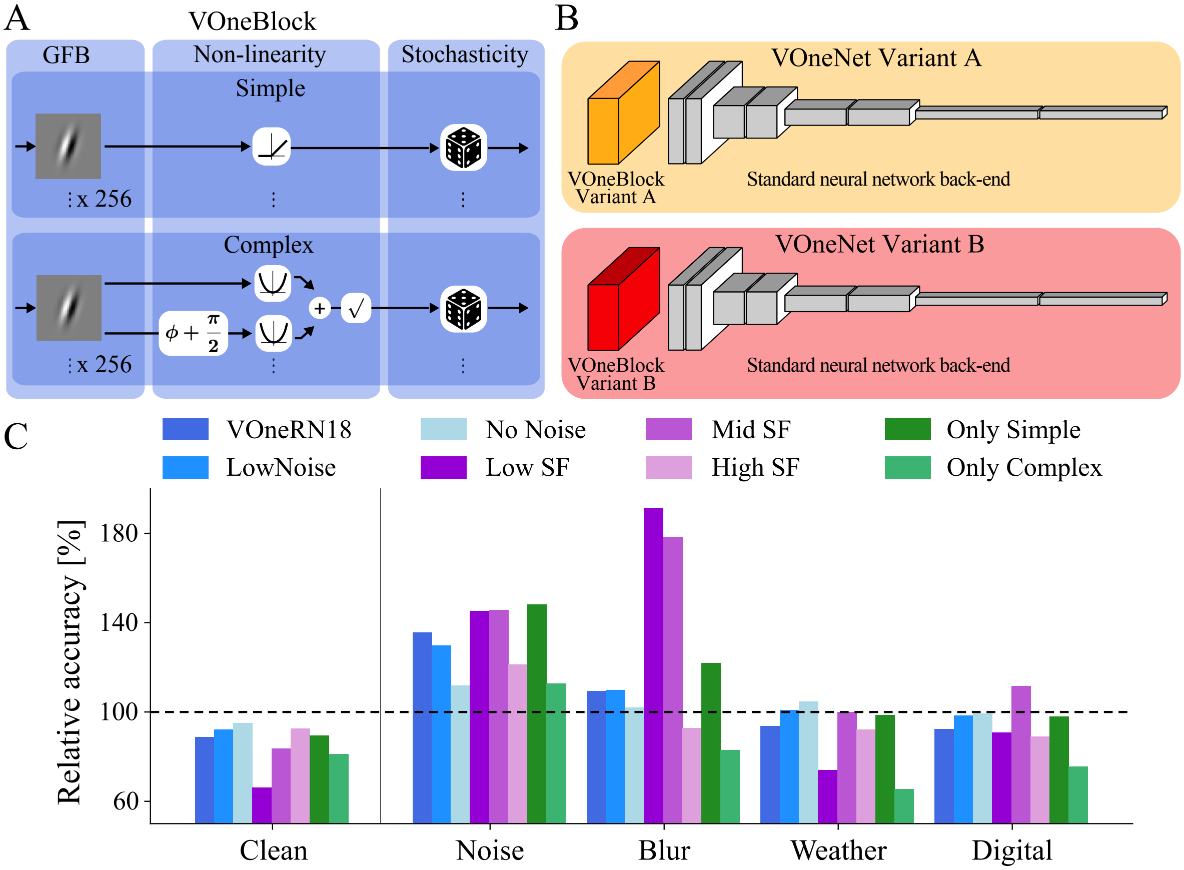


FIGURE 5.1. **Different V1 model variants show distinct robustness trade-offs.** **A** The VOneBlock is a model of V1 with a GFB, a non-linear stage and stochasticity generator. **B** Each VOneNet variant contains a different VOneBlock, built by removing or modifying one of its components. Here, we used eight different variants: standard VOneBlock, no neural stochasticity (No Noise), sub-Poisson stochasticity (Low Noise), only low SF filters (Low SF), only intermediate SF filters (Mid SF), only high SF filters (High SF), only simple-cells (Only Simple), and only complex-cells (Only Complex). **C** Relative accuracy (normalized by the base model, ResNet18) of the eight variants of VOneResNet18 for clean images and corruption categories (see Table 5.2 for absolute accuracies).

blur categories (Fig. 5.2). However, models trained with GNT were significantly less robust than Variants Ensemble in all categories of corruptions and performed worse on clean images.

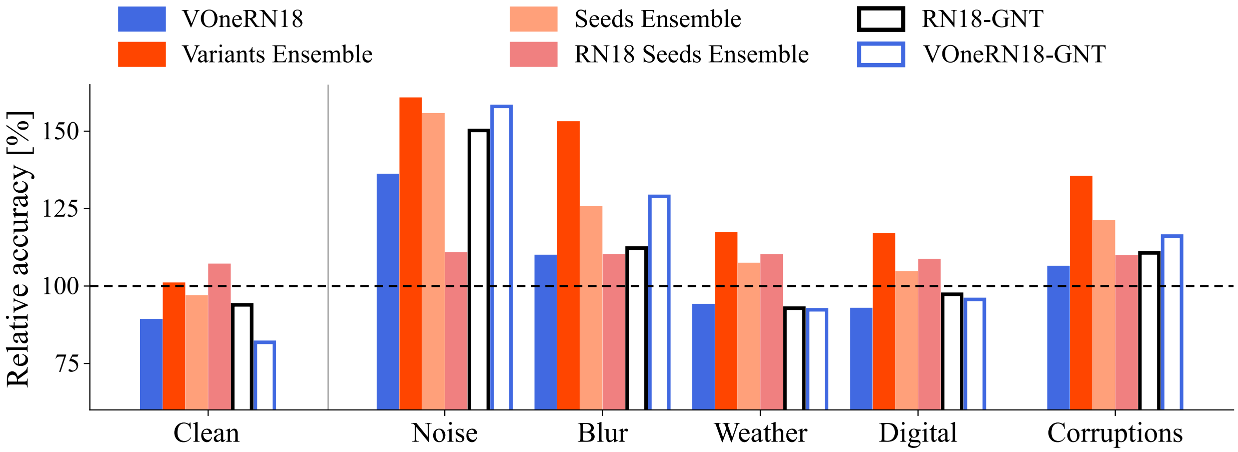


FIGURE 5.2. **Combining different VOneNet variants with model ensembling improves robustness to all corruption categories.** Relative accuracy (normalized by ResNet18) on clean images, all corruptions categories, and overall corruptions for the standard VOneResNet18, the Variants Ensemble, the Seeds Ensemble, the ResNet18 Seeds Ensemble, and the ResNet18 and VOneResNet18 trained with Gaussian Noise augmentation (see Table 5.3 for absolute accuracies).

5.2.3. Training with distillation improves VOneNet robustness against corruptions

While the Variants Ensemble is consistently more robust and has better clean accuracy than any of the individual variants, it is also computationally more expensive. Knowledge Distillation can be used to compress the knowledge in an ensemble (teacher) into a single model (student) [68, 280, 281, 282, 283, 284]. Using this technique, we trained a ResNet18, a standard VOneResNet18 and a No Noise variant by distilling the Variants Ensemble into these three models (Section 5.4.3). Interestingly, distillation has little effect on the performance of the standard ResNet18 and VOneResNet18 (Fig. 5.3, Table 5.4). While the first result using ResNet18 suggests that the VOneBlock in the student architecture is critical for the success of this approach, the latter implies that stochasticity undermines the ability of the student to distill the knowledge in the teacher. Surprisingly, we observed consistent and considerable improvement in the performance of the No Noise variant on clean images and all corruption categories (Fig. 5.3, Table 5.4). In fact, the distilled version of the No Noise variant performed nearly as well as ResNet18 on clean images (98% relative accuracy) and considerably outperformed ResNet18 in all categories of corruptions (between 9% and 21% with 13% overall). Still, it failed to come close to the performance of the

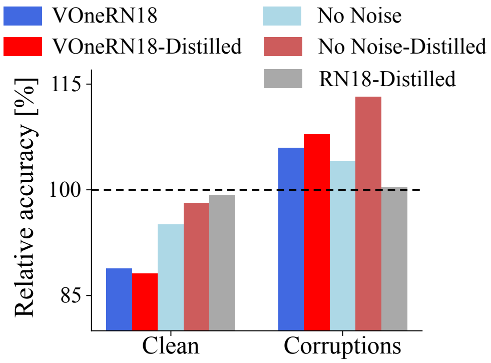


FIGURE 5.3. **VOneResNet18 without stochasticity can partially compress the knowledge in the Variants Ensemble with distillation.** Relative accuracy (normalized by ResNet18) on clean images and overall corruptions for VOneResNet18 trained with and without knowledge distillation, the VOneResNet18 No Noise variant trained with and without knowledge distillation, and ResNet18 trained with knowledge distillation. In all distilled models the Variants Ensemble was used as the teacher (see Table 5.4 for absolute accuracies).

much larger Variants Ensemble, showing that diversity at the level of the VOneBlock is required for the gains observed before. Thus, we developed a VOneNet model that can partially compress the knowledge in the Variants Ensemble, outperforming ResNet18 in all categories of corruptions while maintaining clean accuracy.

5.3. Discussion

Developing models that are more robust to image perturbations and can better generalize to OOD stimuli is a major goal in computer vision. Here, we built models that improve robustness to all categories of common image corruptions without compromising on clean accuracy by combining machine learning techniques (ensembling and distillation) and a biologically-constrained front-end (the VOneBlock). While these models fail short of fully addressing the problem of robust generalization and employ techniques that are hardly biologically-plausible, this work demonstrates that it is possible to combine the different advantages of specific V1 neuronal populations to build models with considerable gains in robustness to common corruptions.

There have been extensive studies characterizing the different neuronal populations in primate V1 [59, 60, 61, 288]. However, there remains a significant gap in our understanding of the roles played by these various neuronal types in dealing with different image statistics. By simulating a V1-like front-end whose components are mappable to the brain, Dapello, Marques, et al. took the first steps to investigate the role of specific neuronal types in dealing with adversarial attacks and common image corruptions [51]. Here, we build on their work by generating variants of additional cell types and investigating their roles in robustness to multiple common image corruptions. Our results demonstrate that different V1 cell types are vulnerable to different corruptions while conferring benefits to others. For example, neurons with high peak spatial frequencies are important for clean image performance but cause models to be more susceptible to blur and noise corruptions. Interestingly, simple cells were found to be beneficial in all cases while complex cells increased vulnerability to all corruptions (although complex cells had been found to be beneficial for adversarial robustness [51]).

Ensembling techniques constructed using diverse individual models have better generalization abilities [274,277,286,287]. To our knowledge, this is the first study to leverage the different properties of V1 neuronal populations to create diverse members of an ensemble. The gains achieved by the ensemble are substantial with 38% relative improvement over all corruptions with same clean accuracy. To contextualize these gains, GNT, a popular defense method, leads to only 11% better accuracy to corruptions and decreases the clean performance by 6% in our experiments. Since training-based defense methods can be applied to the individual variants, the performance gains of the ensemble could potentially be stacked with other methods to achieve even better robustness. Finally, while our ensemble is created simply by averaging the outputs of its individual members, other more elaborate approaches can be used to optimally combine the individual models to further improve performance.

Knowledge distillation has been shown to improve robustness to image corruptions [47,281,283,285]. Here, we demonstrate that a V1-inspired CNN can lead to robustness gains through distillation. In addition, our results show that stochasticity in the student hinders robustness gains through distillation. While we are aware of studies [285,289] that add noise to the teacher, labels, or inputs to help improve distillation, this is the first study to report effects of neuronal stochasticity in the student.

Unfortunately, the models presented here remain far from perfect robust generalization. Still, future work may expand on this work in multiple directions. For example, it remains to be seen how different V1 neuronal properties interact to improve the network's performance (e.g. high spatial frequency-selective and simple cells). Future work may also explore the role of individual variants and their interactions in the ensemble performance to develop even more robust and efficient ensembles. Furthermore, while the ensembling approach taken here to combine different V1 neuronal populations is not biologically-plausible, other modeling strategies that more closely emulate brain processing may produce similar outcomes. These could include cortical computations such as divisive normalization or gain-control mechanisms to combine the different V1 neuronal populations and generate even stronger improvements in robustness. Additionally, while the improvements in robustness suggest that these models may indeed be more aligned with primate vision, it remains to be seen whether these models better approximate the primate ventral stream. Future work may evaluate how well these models predict neuronal responses in multiple cortical areas and how aligned their outputs are with object recognition behavior [290, 291, 292].

Acknowledgements

This work resulted from a project started during the Brains, Minds, and Machines 2021 Summer Course in Woods Hole, MA. A.B. and T.M. would like to thank the course directors Gabriel Kreiman, Boris Katz, and Tomaso Poggio, as well as all the other students and instructors for creating such a unique scientific environment. This work was supported by the PhRMA Foundation Postdoctoral Fellowship in Informatics (T.M.), the Semiconductor Research Corporation (SRC) and DARPA (J.D., J.J.D.), Office of Naval Research grant MURI-114407 (J.J.D.), the Simons Foundation grant SCGB-542965 (J.J.D.), the MIT-IBM Watson AI Lab grant W1771646 (J.J.D.).

5.4. Methods

5.4.1. Datasets

5.4.1.1. Tiny ImageNet

We used the Tiny ImageNet dataset for model training and evaluating model clean accuracy [273]. Tiny ImageNet contains 100.000 images of 200 classes (500 for each class) downsized to 64×64 colored images. Each class has 500 training images, 50 validation images and 50 test images. Tiny ImageNet is publicly available at https://www.kaggle.com/c/tiny-imagenet.

5.4.1.2. Common Corruptions (Tiny ImageNet-C)

For evaluating model robustness to common corruptions we used the Tiny ImageNet-C dataset [41]. The Tiny ImageNet-C dataset consists of 15 different corruption types, each at 5 levels of severity for a total of 75 different perturbations, applied to validation images of Tiny ImageNet 5.4. The individual corruption types are: Gaussian noise, shot noise, impulse noise, defocus blur, glass blur, motion blur, zoom blur, snow, frost, fog, brightness, contrast, elastic transform, pixelate and JPEG compression (Fig. 5.4). The individual corruption types are grouped into 4 categories: noise, blur, weather, and digital effects. The Tiny ImageNet-C is publicly available at https://github.com/hendrycks/robustness under Creative Commons Attribution 4.0 International.

5.4.2. Models

5.4.2.1. VOneNets

VOneNet Model Family VOneNets [51] are CNNs with a biologically-constrained fixed-weight front-end that simulates V1, the VOneBlock - a linear-nonlinear-Poisson (LNP) model of V1 [56], consisting of a fixed-weight Gabor filter bank (GFB) [53], simple and complex cell [57] nonlinearities, and neuronal stochasticity [58]. For the standard model, the GFB parameters are generated

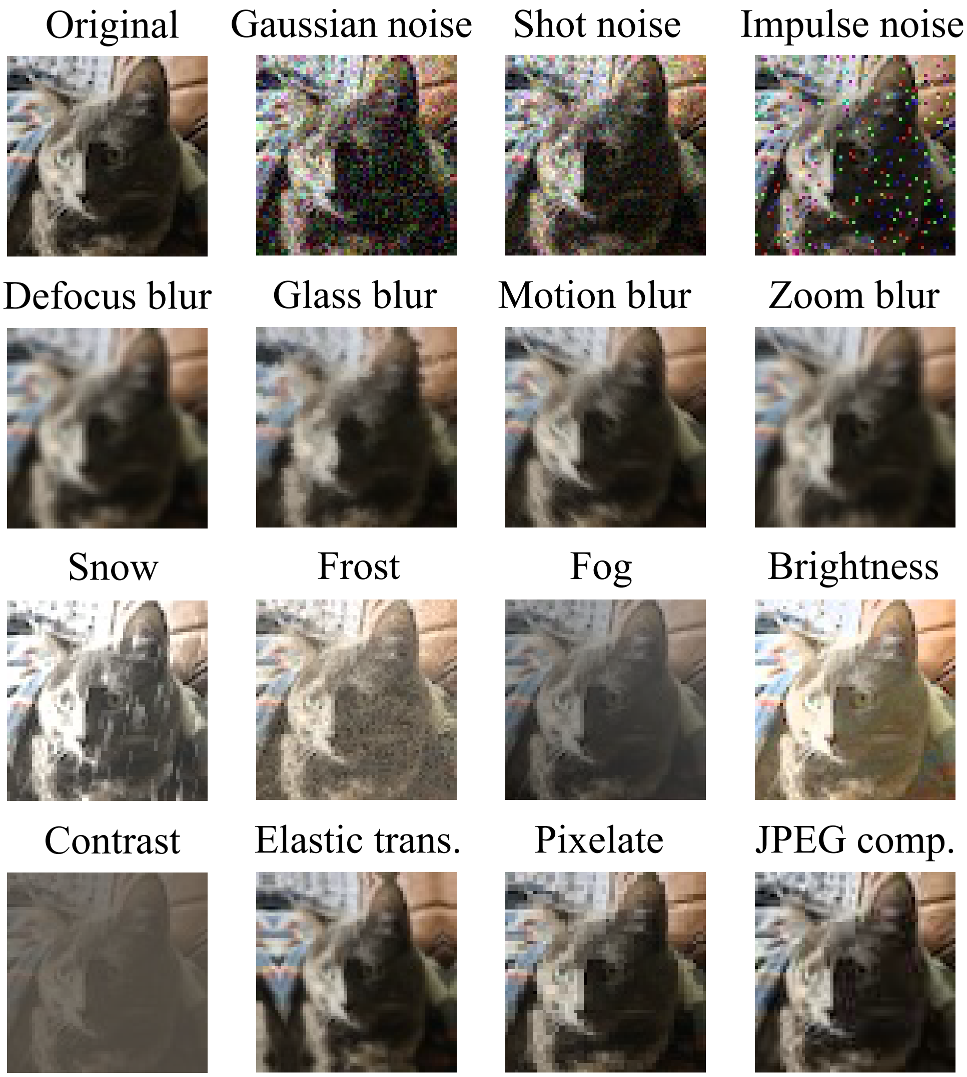


FIGURE 5.4. **Common image corruptions at Tiny ImageNet resolution.** All 15 types of common image corruptions evaluated at severity = 3 for an image at the resolution of Tiny ImageNet (64px). Picture of Milou (credits Tiago Marques).

by randomly sampling from empirically observed distributions of preferred orientation, peak spatial frequency (SF), and size/shape of receptive fields [59, 60, 61], the channels are divided equally between simple- and complex-cells (256 each), and a Poisson-like stochasticity generator is used. Code for the VOneNet model family is publicly available at `https://github.com/dicarlolab/vonenet` under GNU General Public License v3.0.

Adapting VOneNets to Tiny ImageNet To facilitate model development and evaluation, we adapted the VOneNet architecture to the Tiny ImageNet image size and chose the ResNet18 architecture [272] for the back-end. Specifically, VOneResNet18 is built by replacing the first block (one stack of convolution, normalization, non-linearity and pooling layers) of ResNet18 [272] by the VOneBlock and a trainable bottleneck layer. Due to the difference in input size (from 224px for ImageNet to 64px in Tiny ImageNet), we made several modifications to the model architecture. First, we set the stride of the convolution layer (GFB) at two instead of four such that the output of the VOneBlock does not have a very small spatial map. We also adjusted the input field of view to 2deg for Tiny ImageNet instead of 8deg for ImageNet to account for the fact that images in the first represent a narrower portion of the visual space. This change resulted in an input resolution - number of pixels per degree (ppd) - of 32 ppd for Tiny ImageNet which is similar to that of ImageNet (28 ppd).

VOneResNet18 Variants We created seven VOneResNet18 model variants by removing or modifying one of the VOneBlock components (Table 5.1). Three variants targeted the GFB: one with only low SF filters (Low SF; SF $< 2cpd$), one with only intermediate SF filters (Mid SF; $2cpd <$ SF $< 5.6cpd$), and one with only high SF filters (High SF; SF $> 5.6cpd$). Two additional variants targeted the nonlinearities: one with only simple-cells (Only Simple) and one with only complex-cells (Only Complex). Finally, the last two variants targeted the stochasticity generator: one with a sub-Poisson like ($\sigma = \frac{\sqrt{\mu}}{2}$) stochasticity generator (Low Noise) and one without the stochasticity generator (No Noise).

TABLE 5.1. **Parameters of VOneBlock Variants.**

Variant Name	Spatial Frequency [cpd]	Cell Types [simple/complex]	Stochasticity [type]
VOneResNet18	0.5 - 11.2	256/256	Poisson
Low SF	0.5 - 2.0	256/256	Poisson
Mid SF	2.0 - 5.6	256/256	Poisson
High SF	5.6 - 11.2	256/256	Poisson
Only Simple	0.5 - 11.2	512/0	Poisson
Only Complex	0.5 - 11.2	0/512	Poisson
Low Noise	0.5 - 11.2	256/256	Sub-Poisson
No Noise	0.5 - 11.2	256/256	None

5.4.2.2. ResNet18

We used a variant of the Torchvision implementation of ResNet18 [272] as the base model and as the model back-end for the VOneResNet18. The original ResNet18 model, contains a combined stride of four in the first block (two in the convolution layer and two in the maxpool layer) which is the block replaced by VOneBlock in VOneResNet18. In order to maintain the size of the model comparable to the VOneResNet18, we adapted the ResNet18 architecture so that it has a stride of one in the first convolutional layer and kept the stride of two in the maxpool layer, resulting in a combined stride of two in the first block which is the same as the VOneBlock. We found that this variant of ResNet18 (58.93% accuracy) performed considerably better than the standard Torchvision implementation of ResNet18 (50.45% accuracy) on clean Tiny ImageNet images after training from scratch following an identical training procedure (Section 5.4.3).

5.4.2.3. Ensembles

Variants Ensemble We created this ensemble by uniformly averaging the outputs (logits) of the eight VOneNet variants shown in Table 5.1. The variants were trained individually prior to ensembling.

Seeds Ensemble We created this ensemble by combining eight different training seeds of the standard VOneResNet18 model using the same approach as with the Variants Ensemble. The different training seeds of the standard VOneResNet18 model were created by using different seeds to instantiate the GFB parameters of the VOneBlock and then training the back-ends.

ResNet18 Seeds Ensemble We created this ensemble by combining eight different training seeds of the ResNet18 model using the same approach as with the Variants Ensemble and the VOneResNet18 Seeds Ensemble.

5.4.3. Training

PyTorch version 1.9.0 was used. All models were trained on Google Colab which provided access to 1 GPU (Nvidia K80, T4, P4 or P100) per session. The details of the training are described as follows.

Preprocessing During training, preprocessing included scaling the images using a scaling factor randomly sampled between 1-1.2, rotating the images using a rotation angle randomly sampled between -30 to 30 degrees, shifting the images in the horizontal direction by a pixel distance randomly sampled between -5% to 5% of the image width, shifting the images in the vertical direction by a pixel distance randomly sampled between -5% to 5% of the image height, and flipping the images horizontally with a random probability of 0.5. Images were normalized by subtraction and division by [0.5, 0.5, 0.5]. For models trained with GNT [44], preprocessing involved an additional step of adding Gaussian noise (standard deviation of 0.6) to 50% of all images. During evaluation, preprocessing only involved image normalization, i.e. subtraction and division by [0.5, 0.5, 0.5].

Loss Functions For models trained without knowledge distillation, the loss function was given by a cross-entropy loss between image labels and model predictions (logits). For models trained with knowledge distillation, the loss function was given by a weighted average of two loss functions [68]. The first loss function with a weight 100 minimized the cross-entropy between the output probability distributions (soft targets) of the distilled and the emsemble model. The soft targets for both those models were computed using temperature 5. The second loss function with a weight 5 minimized the cross-entropy between image labels and the distilled model predictions (logits).

Optimization For optimization, we used Stochastic Gradient Descent with momentum 0.9, a weight decay 0.0005, and an initial learning rate 0.1. The learning rate was dynamically adjusted by dividing it by 10 whenever there is no improvement in validation loss for 5 consecutive epochs. All models were trained using a batch size of 128 images for 60 epochs.

5.5. Supplementary Materials

Table 5.2. **Absolute accuracies of ResNet18, standard VOneResNet18 and the seven additional variants on the 15 types of common image corruptions (averaged over perturbation severities).**

		Noise			Blur			
	Clean	Gaussian	Shot	Impulse	Defocus	Glass	Motion	Zoom
Model	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
ResNet18	**58.9**	19.8	23.2	21.9	14.5	20.0	20.5	16.6
VOneResNet18	52.3	28.7	32.7	26.5	16.9	19.8	22.3	18.8
Low Noise	54.4	27.1	31.5	25.6	16.9	20.3	22.4	18.6
No Noise	56.0	22.9	27.6	22.1	15.4	19.0	21.2	17.1
Low SF	39.0	31.2	32.8	**29.9**	**33.4**	**32.5**	**33.8**	**34.3**
Mid SF	49.3	31.1	**35.1**	28.1	29.2	28.7	**33.8**	33.7
High SF	54.6	24.9	29.0	24.8	13.9	18.0	18.8	15.5
Only Simple	52.8	**31.9**	34.9	29.1	19.3	21.2	24.5	21.4
Only Complex	47.9	23.7	27.0	22.3	12.8	15.7	16.3	14.1

	Weather				Digital			
	Snow	Frost	Fog	Bright.	Contrast	Elastic	Pixelate	JPEG
Model	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
ResNet18	24.7	26.0	**22.8**	28.6	**10.5**	39.1	33.6	25.6
VOneResNet18	26.0	25.3	17.7	26.9	6.2	34.3	37.1	28.7
Low Noise	27.0	27.1	20.1	29.0	7.7	36.1	38.5	29.1
No Noise	**27.7**	**28.0**	22.4	28.8	9.2	36.1	37.3	27.4
Low SF	19.9	19.9	13.9	22.4	4.7	34.4	33.8	33.3
Mid SF	26.5	27.5	19.5	**29.2**	7.1	**40.4**	**41.7**	**38.9**
High SF	24.9	24.8	18.1	26.8	6.7	34.5	35.1	25.5
Only Simple	26.4	27.5	18.8	28.5	6.8	35.5	38.7	30.9
Only Complex	16.8	16.1	13.8	20.5	4.6	29.3	31.8	22.8

Table 5.3. **Absolute accuracies of ResNet18, standard VOneResNet18, and the three ensemble models: Variants Ensemble, Seeds Ensemble and ResNet18 Seeds Ensemble (averaged over perturbation severities)**.

		Noise			Blur			
Model	Clean [%]	Gaussian [%]	Shot [%]	Impulse [%]	Defocus [%]	Glass [%]	Motion [%]	Zoom [%]
ResNet18	58.9	19.8	23.2	21.9	14.5	20.0	20.5	16.6
VOneResNet18	52.3	28.7	32.7	26.5	16.9	19.8	22.3	18.8
Variants Ensemble	59.3	**33.8**	**38.3**	**31.7**	**24.3**	**26.5**	**30.3**	**26.9**
Seeds Ensemble	56.8	32.9	37.0	30.7	19.5	22.8	24.7	21.8
ResNet18 Seeds Ensemble	**62.9**	21.6	25.4	24.6	16.2	21.5	22.3	18.3

	Weather				Digital			
Model	Snow [%]	Frost [%]	Fog [%]	Bright. [%]	Contrast [%]	Elastic [%]	Pixelate [%]	JPEG [%]
ResNet18	24.7	26.0	22.8	28.6	10.5	39.1	33.6	25.6
VOneResNet18	26.0	25.3	17.7	26.9	6.2	34.3	37.1	28.7
Variants Ensemble	**31.8**	**31.7**	**22.9**	**33.4**	8.3	**42.9**	**44.8**	**37.0**
Seeds Ensemble	30.0	29.0	20.0	30.6	7.0	38.8	41.6	32.5
ResNet18 Seeds Ensemble	27.5	28.7	24.6	31.2	**11.1**	42.4	36.4	28.2

Table 5.4. **Absolute accuracies of ResNet18, VOneResNet18, No Noise variant, and the three distillation models: VOneResNet18-Distilled, No Noise-Distilled and ResNet18-Distilled (averaged over perturbation severities)**.

		Noise			Blur			
Model	Clean [%]	Gaussian [%]	Shot [%]	Impulse [%]	Defocus [%]	Glass [%]	Motion [%]	Zoom [%]
ResNet18	**58.9**	19.8	23.2	21.9	14.5	20.0	20.5	16.6
VOneResNet18	52.3	28.7	32.7	26.5	16.9	19.8	22.3	18.8
VOneResNet18-Distilled	51.9	**29.1**	**32.9**	**27.0**	**17.4**	20.4	22.8	**19.3**
No Noise	56.0	22.9	27.6	22.1	15.4	19.0	21.2	17.1
No Noise-Distilled	57.8	25.0	30.0	24.1	16.2	**20.6**	**23.0**	18.1
ResNet18-Distilled	58.5	20.2	23.8	22.5	15.5	19.2	20.8	17.5

	Weather				Digital			
Model	Snow [%]	Frost [%]	Fog [%]	Bright. [%]	Contrast [%]	Elastic [%]	Pixelate [%]	JPEG [%]
ResNet18	24.7	26.0	22.8	28.6	**10.5**	39.1	33.6	25.6
VOneResNet18	26.0	25.3	17.7	26.9	6.2	34.3	37.1	28.7
VOneResNet18-Distilled	26.4	25.8	18.0	27.0	6.5	34.9	37.5	29.3
No Noise	27.7	28.0	22.4	28.8	9.2	36.1	37.3	27.4
No Noise-Distilled	**30.1**	**30.3**	**24.2**	**32.0**	**10.5**	**39.4**	**40.6**	**30.2**
ResNet18-Distilled	24.9	25.7	22.1	27.8	9.9	38.8	33.2	26.0

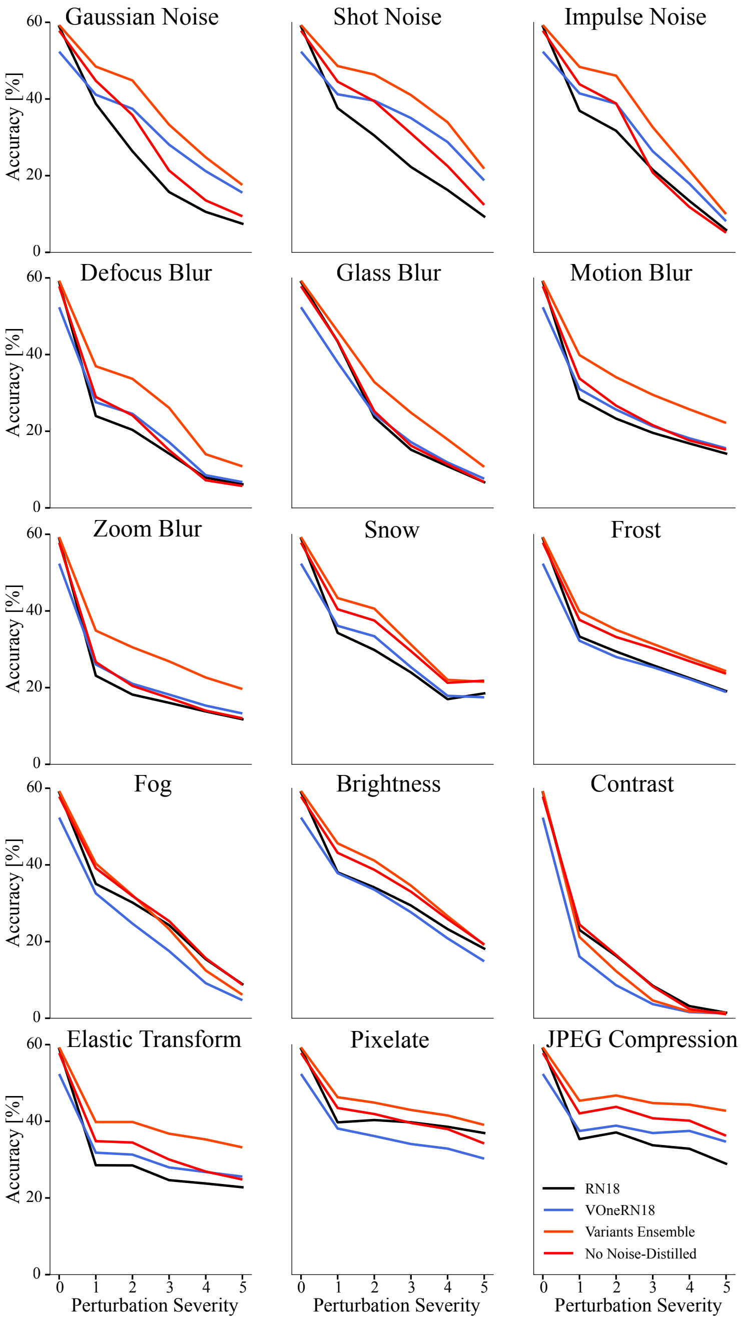


FIGURE 5.5. **Variants Ensemble and distilled VOneResNet18 No Noise show consistent improvements in robustness** Absolute accuracies for ResNet18, VOneResNet18, Variants Ensemble, and distilled VOneResNet18 without stochasticity for the 15 corruption types at all perturbation severity levels. Accuracy represents top-1. Perturbation 0 corresponds to clean images.

CHAPTER 6

Conclusion

This dissertation explores the intersection of machine learning (ML) and biology, demonstrating how these fields can synergize to address complex challenges. It comprises a collection of three studies that investigate the application of reinforcement learning (RL) in analyzing reward-based learning in animals, the application of ML in deciphering the mechanisms of SARS-CoV-2, and the application of biological principles to enhance CNNs. Each study contributes distinctively to the broader theme of merging ML techniques with biological knowledge.

We start by delving into the study of RL mechanisms in animals, specifically examining how neural circuits and neurotransmitter systems mediate reward-based learning processes. Through a combination of experimental observations and computational modeling, we shed light on the crucial roles played by the ventral striatum, the prefrontal cortex, and the ventral tegmental area in driving these intricate learning processes. By combining the choice-selective sequential activity observed in prefrontal cortical inputs to the striatum with known downstream circuitry, our models employ dopamine-driven learning mechanisms to either induce rapid synaptic plasticity or to alter neural dynamics. These models thus generate two hypotheses for the process by which animals learn to associate actions with delayed rewards. This research enhances our understanding of the neural basis of RL and illustrates how computational models can effectively fill the existing gaps in our understanding of brain functionality and learning processes.

Transitioning to the field of virology, the second part of this dissertation offers a detailed computational study of the SARS-CoV-2 virus, focusing particularly on how its spike protein interacts with host cells and evades the immune response. By employing molecular dynamics simulations and the advanced deep learning tool AlphaFold2, we thoroughly analyze the binding characteristics of key domains of the spike protein, including RBD-ACE2, RBD-antibody, FCD-Furin, and NTD-antibody. We conduct a comparative analysis of the Omicron variants against the wild type and Delta variants, offering significant insights into the structural factors that influence the virus's ability to infect and evade immune responses. Further, our research delves deeper into the FCD of SARS-CoV-2 and other related viruses, examining their interaction with the furin enzyme. Notably, our findings reveal that the Delta variant demonstrates the strongest binding affinity to the furin enzyme, and we identify crucial sequences, both observed and previously unobserved, that have similar binding potentials. Collectively, our research provides a detailed understanding of SARS-CoV-2's mechanisms, significantly contributing to the global scientific community's understanding of SARS-CoV-2 and aiding in the development of effective strategies to counteract the virus.

In the final part of this dissertation, we investigate how integrating biological insights from the primary visual cortex into CNNs can improve robustness against image corruptions. We particularly focus on VOneNet, a hybrid CNN model that draws inspiration from the V1 area of the primary visual cortex. Our approach demonstrates the effectiveness of combining various V1-inspired variants to mitigate performance trade-offs across various corruptions. By employing ensembling and knowledge distillation, we significantly enhance the model's overall robustness and successfully demonstrate how the knowledge from an ensemble of V1-inspired models can be compressed into a single, more efficient model. Through this work, we not only advance the field of computer vision but also highlight the potential of interdisciplinary research in leveraging biological principles to overcome challenges in ML.

Overall, this dissertation represents a comprehensive exploration at the intersection of ML and biology, offering new insights and methodologies that push the boundaries of both fields. Each study in this dissertation contributes to a broader narrative: that combining machine learning with biological understanding offers significant potential for deciphering the intricacies of both living organisms and computational systems, paving the way for innovative solutions to complex problems in these fields.

Bibliography

[1] P Apicella, T Ljungberg, E Scarnati, and W Schultz. Responses to reward in monkey dorsal and ventral striatum, 1991.

[2] M Cador, T W Robbins, and B J Everitt. Involvement of the amygdala in stimulus-reward associations: interaction with the ventral striatum. *Neuroscience*, 30(1):77–86, 1989.

[3] Regina M Carelli, Virginia C King, Robert E Hampson, and Sam A Deadwyler. Firing patterns of nucleus accumbens neurons during cocaine self-administration in rats, 1993.

[4] Julia Cox and Ilana B Witten. Striatal circuits for reward learning and decision-making. *Nature Reviews Neuroscience*, 20:482–494, 2019.

[5] P Di Ciano, R N Cardinal, R A Cowell, S J Little, and B J Everitt. Differential involvement of NMDA, AMPA/kainate, and dopamine receptors in the nucleus accumbens core in the acquisition and performance of pavlovian approach behavior. *J. Neurosci.*, 21(23):9471–9477, December 2001.

[6] B J Everitt, K A Morris, A O'Brien, and T W Robbins. The basolateral amygdala-ventral striatal system and conditioned place preference: further evidence of limbic-striatal interactions underlying reward-related processes. *Neuroscience*, 42(1):1–18, 1991.

[7] J A Parkinson, M C Olmstead, L H Burns, T W Robbins, and B J Everitt. Dissociation in effects of lesions of the nucleus accumbens core and shell on appetitive pavlovian approach behavior and the potentiation of conditioned reinforcement and locomotor activity by d-amphetamine. *J. Neurosci.*, 19(6):2401–2411, March 1999.

[8] G D Phillips, J Le Noury, G Wolterink, I Donselaar-Wolterink, T W Robbins, and B J Everitt. Cholecystokinin-dopamine interactions within the nucleus accumbens in the control over behaviour by conditioned reinforcement. *Behav. Brain Res.*, 55(2):223–231, June 1993.

[9] G D Phillips, T W Robbins, and B J Everitt. Mesoaccumbens dopamine-opiate interactions in the control over behaviour by a conditioned reinforcer. *Psychopharmacology*, 114(2):345–359, March 1994.

[10] T W Robbins, M Cador, J R Taylor, and B J Everitt. Limbic-striatal interactions in reward-related processes, 1989.

[11] Mitchell F Roitman, Robert A Wheeler, and Regina M Carelli. Nucleus accumbens neurons are innately tuned for rewarding and aversive taste stimuli, encode their predictors, and are linked to motor output, 2005.

[12] Barry Setlow, Geoffrey Schoenbaum, and Michela Gallagher. Neural encoding in ventral striatum during olfactory discrimination learning. *Neuron*, 38(4):625–636, May 2003.

[13] Garret D Stuber, Dennis R Sparta, Alice M Stamatakis, Wieke A van Leeuwen, Juanita E Hardjoprajitno, Saemi Cho, Kay M Tye, Kimberly A Kempadoo, Feng Zhang, Karl Deisseroth, and Antonello Bonci. Excitatory transmission from the amygdala to nucleus accumbens facilitates reward seeking. *Nature*, 475:377, June 2011.

[14] Janer Taylor and Trevorw Robbins. 6-hydroxydopamine lesions of the nucleus accumbens, but not of the caudate nucleus, attenuate enhanced responding with reward-related stimuli produced by intra-accumbens d-amphetamine, 1986.

[15] J S Brog, A Salyapongse, A Y Deutch, and D S Zahm. The patterns of afferent innervation of the core and shell in the "accumbens" part of the rat ventral striatum: immunohistochemical detection of retrogradely transported fluoro-gold. *J. Comp. Neurol.*, 338(2):255–278, December 1993.

[16] Fabricio H Do-Monte, Angélica Minier-Toribio, Kelvin Quiñones-Laracuente, Estefanía M Medina-Colón, and Gregory J Quirk. Thalamic regulation of sucrose seeking during unexpected reward omission. *Neuron*, 94(2):388–400.e4, April 2017.

[17] H J Groenewegen, N E Becker, and A H Lohman. Subcortical afferents of the nucleus accumbens septi in the cat, studied with retrograde axonal transport of horseradish peroxidase and bisbenzimid. *Neuroscience*, 5(11):1903–1916, 1980.

[18] Barbara J Hunnicutt, Bart C Jongbloets, William T Birdsong, Katrina J Gertz, Haining Zhong, and Tianyi Mao. A comprehensive excitatory input map of the striatum reveals novel functional organization. *Elife*, 5, November 2016.

[19] James M Otis, Vijay M K Namboodiri, Ana M Matan, Elisa S Voets, Emily P Mohorn, Oksana Kosyk, Jenna A McHenry, J Elliott Robinson, Shanna L Resendez, Mark A Rossi, and Garret D Stuber. Prefrontal cortex output circuits guide reward seeking through divergent cue encoding. *Nature*, 543(7643):103–107, March 2017.

[20] O T Phillipson and A C Griffiths. The topographic order of inputs to nucleus accumbens in the rat. *Neuroscience*, 16(2):275–296, October 1985.

[21] Jean-Francois Poulin, Giuliana Caronia, Caitlyn Hofer, Qiaoling Cui, Brandon Helm, Charu Ramakrishnan, C Savio Chan, Daniel A Dombeck, Karl Deisseroth, and Rajeshwar Awatramani. Mapping projections of molecularly defined dopamine neuron subtypes using intersectional genetic approaches. *Nat. Neurosci.*, 21(9):1260–1271, September 2018.

[22] Sean J Reed, Christopher K Lafferty, Jesse A Mendoza, Angela K Yang, Thomas J Davidson, Logan Grosenick, Karl Deisseroth, and Jonathan P Britt. Coordinated reductions in excitatory input to the nucleus accumbens underlie food consumption. *Neuron*, 99(6):1260–1273.e4, September 2018.

[23] L W Swanson. The projections of the ventral tegmental area and adjacent regions: a combined fluorescent retrograde tracer and immunofluorescence study in the rat. *Brain Res. Bull.*, 9(1-6):321–353, July 1982.

[24] C I Wright and H J Groenewegen. Patterns of convergence and segregation in the medial nucleus accumbens of the rat: relationships of prefrontal cortical, midline thalamic, and basal amygdaloid afferents. *J. Comp. Neurol.*, 361(3):383–403, October 1995.

[25] Yingjie Zhu, Carl F R Wienecke, Gregory Nachtrab, and Xiaoke Chen. A thalamic input to the nucleus accumbens mediates opiate dependence. *Nature*, 530(7589):219–222, February 2016.

[26] Nathan F Parker, Courtney M Cameron, Joshua P Taliaferro, Junuk Lee, Jung Yoon Choi, Thomas J Davidson, Nathaniel D Daw, and Ilana B Witten. Reward and choice encoding in terminals of midbrain dopamine neurons depends on striatal target. *Nat. Neurosci.*, 19(6):845–854, June 2016.

[27] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[28] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[29] Philip V'kovski, Annika Kratzel, Silvio Steiner, Hanspeter Stalder, and Volker Thiel. Coronavirus biology and replication: implications for sars-cov-2. *Nature Reviews Microbiology*, 19(3):155–170, 2021.

[30] Haitao Yang and Zihe Rao. Structural biology of sars-cov-2 and implications for therapeutic development. *Nature Reviews Microbiology*, 19(11):685–700, 2021.

[31] William T Harvey, Alessandro M Carabelli, Ben Jackson, Ravindra K Gupta, Emma C Thomson, Ewan M Harrison, Catherine Ludden, Richard Reeve, Andrew Rambaut, Sharon J Peacock, et al. Sars-cov-2 variants, spike mutations and immune escape. *Nature reviews microbiology*, 19(7):409–424, 2021.

[32] Chuan Tian, Koushik Kasavajhala, Kellon AA Belfon, Lauren Raguette, He Huang, Angela N Migues, John Bickel, Yuzhang Wang, Jorge Pincay, Qin Wu, et al. ff19sb: Amino-acid-specific protein backbone parameters trained against quantum mechanics energy surfaces in solution. *Journal of chemical theory and computation*, 16(1):528–552, 2019.

[33] Wendy D Cornell, Piotr Cieplak, Christopher I Bayly, Ian R Gould, Kenneth M Merz, David M Ferguson, David C Spellmeyer, Thomas Fox, James W Caldwell, and Peter A Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society*, 117(19):5179–5197, 1995.

[34] William L Jorgensen, Jayaraman Chandrasekhar, Jeffry D Madura, Roger W Impey, and Michael L Klein. Comparison of simple potential functions for simulating liquid water. *The Journal of chemical physics*, 79(2):926–935, 1983.

[35] J. Jumper, R. Evans, and A. et al. Pritzel. Highly accurate protein structure prediction with alphafold. *Nature*, 2021.

[36] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv:1312.6199 [cs]*, February 2014. arXiv: 1312.6199.

[37] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[38] Wieland Brendel, Jonas Rauber, Matthias Kümmerer, Ivan Ustyuzhaninov, and Matthias Bethge. Accurate, reliable and fast robustness evaluation. *Advances in neural information processing systems*, 32, 2019.

[39] Samuel Dodge and Lina Karam. A Study and Comparison of Human and Deep Learning Recognition Performance Under Visual Distortions. *arXiv:1705.02498 [cs]*, May 2017. arXiv: 1705.02498.

[40] Robert Geirhos, Carlos R. Medina Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. Generalisation in humans and deep neural networks. In *NeurIPS*, pages 1–13, 2018.

[41] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *7th International Conference on Learning Representations, ICLR 2019*, pages 1–16, 2019.

[42] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. *ICLR*, pages 1–15, 2020.

[43] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. Jacob steinhardt et justin gilmer. the many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020.

[44] Evgenia Rusak, Lukas Schott, Roland S Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 53–69. Springer, 2020.

[45] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D. Cubuk. Improving Robustness Without Sacrificing Accuracy with Patch Gaussian Augmentation. *arXiv:1906.02611 [cs, stat]*, June 2019. arXiv: 1906.02611.

[46] Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D. Cubuk, and Justin Gilmer. A Fourier Perspective on Model Robustness in Computer Vision. *arXiv:1906.08988 [cs, stat]*, September 2020. arXiv: 1906.08988.

[47] Jungkyu Lee, Taeryun Won, Tae Kwan Lee, Hyemin Lee, Geonmo Gu, and Kiho Hong. Compounding the Performance Improvements of Assembled Techniques in a Convolutional Neural Network. *arXiv:2001.06268 [cs]*, March 2020. arXiv: 2001.06268.

[48] Cristina Vasconcelos, Hugo Larochelle, Vincent Dumoulin, Nicolas Le Roux, and Ross Goroshin. An Effective Anti-Aliasing Approach for Residual Networks. *arXiv:2011.10675 [cs]*, November 2020. arXiv: 2011.10675.

[49] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, pages 1–22, 2019.

[50] Jan Laermann, Wojciech Samek, and Nils Strodthoff. Achieving Generalizable Robustness of Deep Neural Networks by Stability Training. *arXiv:1906.00735 [cs, stat]*, 11824:360–373, 2019. arXiv: 1906.00735.

[51] Joel Dapello, Tiago Marques, Martin Schrimpf, Franziska Geiger, David D. Cox, and James J. DiCarlo. Simulating a primary visual cortex at the front of CNNs improves robustness to image perturbations. *NeurIPS*, pages 1–30, 2020.

[52] Gaurav Malhotra, Benjamin D. Evans, and Jeffrey S. Bowers. Hiding a plane with a pixel: examining shape-bias in CNNs and the benefit of building in biological constraints. *Vision Research*, 174:57–68, September 2020.

[53] J. P. Jones and L. A. Palmer. The two-dimensional spatial structure of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58(6):1187–1211, 1987.

[54] Zhe Li, Wieland Brendel, Edgar Walker, Erick Cobos, Taliah Muhammad, Jacob Reimer, Matthias Bethge, Fabian Sinz, Zachary Pitkow, and Andreas Tolias. Learning from brains how to regularize machines. *Advances in neural information processing systems*, 32, 2019.

[55] Shahd Safarani, Arne Nix, Konstantin Willeke, Santiago A. Cadena, Kelli Restivo, George Denfield, Andreas S. Tolias, and Fabian H. Sinz. Towards robust vision by multi-task learning on monkey visual cortex. *arXiv:2107.14344 [cs]*, July 2021. arXiv: 2107.14344.

[56] Nicole C Rust, Odelia Schwartz, J. A. Movshon, and E. P. Simoncelli. Spatiotemporal Elements of Macaque V1 Receptive Fields. *Neuron*, 46:945–956, 2005.

[57] E.H. Adelson and J.R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A, Optics and image science*, 2(2):284–299, 1985. ISBN: 0740-3232 (Print).

[58] W. R. Softky and C. Koch. The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. *Journal of Neuroscience*, 13(1):334–350, 1993.

[59] Russell L. De Valois, Duane G. Albrecht, and Lisa G. Thorell. Spatial Frequency Selectivity of Cells in Macaque Visual Cortex. *Vision Research*, 22:545–559, 1982.

[60] Russell L. De Valois, E. W. Yund, and Norva Hepler. The orientation and direction selectivity of cells in macaque visual cortex. *Vision Research*, 22:531–544, 1982.

[61] Dario L Ringach. Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex. *Journal of neurophysiology*, 88(1):455–463, 2002.

[62] Thomas G. Dietterich. Ensemble Methods in Machine Learning. In *Multiple Classifier Systems*, pages 1–15, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[63] L.K. Hansen and P. Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, October 1990.

[64] Anders Krogh and Jesper Vedelsby. Neural network ensembles, cross validation, and active learning. *Advances in neural information processing systems*, 7, 1994.

[65] D. Opitz and R. Maclin. Popular Ensemble Methods: An Empirical Study. *Journal of Artificial Intelligence Research*, 11:169–198, August 1999.

[66] Lior Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, February 2010.

[67] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep Ensembles: A Loss Landscape Perspective. *arXiv:1912.02757 [cs, stat]*, June 2020. arXiv: 1912.02757.

[68] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv:1503.02531 [cs, stat]*, March 2015. arXiv: 1503.02531.

[69] Nathan F Parker, Avinash Baidya, Julia Cox, Laura M Haetzel, Anna Zhukovskaya, Malavika Murugan, Ben Engelhard, Mark S Goldman, and Ilana B Witten. Choice-selective sequences dominate in cortical relative to thalamic inputs to nac to support reinforcement learning. *Cell reports*, 39(7), 2022.

[70] Wael F Asaad, Peter M Lauro, János A Perge, and Emad N Eskandar. Prefrontal neurons encode a solution to the Credit-Assignment problem, 2017.

[71] Timothy M Gersch, Nicholas C Foley, Ian Eisenberg, and Jacqueline Gottlieb. Neural correlates of temporal credit assignment in the parietal lobe, February 2014.

[72] Richard S Sutton. Learning to predict by the methods of temporal differences. *Mach. Learn.*, 3(1):9–44, August 1988.

[73] Florentin Wörgötter and Bernd Porr. Temporal sequence learning, prediction, and control: a review of different models and their relation to biological mechanisms. *Neural Comput.*, 17(2):245–319, February 2005.

[74] Daphna Joel, Yael Niv, and Eytan Ruppin. Actor–critic models of the basal ganglia: new anatomical and computational perspectives, 2002.

[75] Simon D Fisher, Paul B Robertson, Melony J Black, Peter Redgrave, Mark A Sagar, Wickliffe C Abraham, and John N J Reynolds. Reinforcement determines the timing dependence of corticostriatal synaptic plasticity in vivo. *Nat. Commun.*, 8(1):334, August 2017.

[76] Charles R Gerfen and D James Surmeier. Modulation of striatal projection systems by dopamine. *Annu. Rev. Neurosci.*, 34:441–466, 2011.

[77] John N J Reynolds and Jeffery R Wickens. Dopamine-dependent plasticity of corticostriatal synapses. *Neural Netw.*, 15(4):507–521, June 2002.

[78] Scott J Russo, David M Dietz, Dani Dumitriu, John H Morrison, Robert C Malenka, and Eric J Nestler. The addicted synapse: mechanisms of synaptic and structural plasticity in nucleus accumbens, 2010.

[79] Jane X Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Demis Hassabis, and Matthew Botvinick. Prefrontal cortex as a meta-reinforcement learning system. *Nat. Neurosci.*, 21(6):860–868, June 2018.

[80] Ben Engelhard, Joel Finkelstein, Julia Cox, Weston Fleming, Hee Jae Jang, Sharon Ornelas, Sue Ann Koay, Stephan Y Thiberge, Nathaniel D Daw, David W Tank, and Ilana B Witten. Specialized coding of sensory, motor and cognitive variables in VTA dopamine neurons. *Nature*, 570(7762):509–513, June 2019.

[81] Michael Krumin, Julie J Lee, Kenneth D Harris, and Matteo Carandini. Decision and navigation in mouse parietal cortex. *Elife*, 7, November 2018.

[82] Matthew Lovett-Barron, Ritchie Chen, Susanna Bradbury, Aaron S Andalman, Mahendra Wagle, Su Guo, and Karl Deisseroth. Multiple overlapping hypothalamus-brainstem circuits drive rapid threat avoidance. *bioRxiv*, page 745075, 2019.

[83] Simon Musall, Matthew T Kaufman, Ashley L Juavinett, Steven Gluf, and Anne K Churchland. Single-trial neural dynamics are dominated by richly varied movements. *Nat. Neurosci.*, 22(10):1677–1686, October 2019.

[84] Il Memming Park, Miriam L R Meister, Alexander C Huk, and Jonathan W Pillow. Encoding and decoding in parietal cortex during sensorimotor decision-making, 2014.

[85] Lucas Pinto and Yang Dan. Cell-type-specific activity in prefrontal cortex during goal-directed behavior. *Neuron*, 87(2):437–450, July 2015.

[86] Bernardo L Sabatini. The impact of reporter kinetics on the interpretation of data gathered with fluorescent reporters. *BioRxiv*, page 834895, 2019.

[87] Nicholas A Steinmetz, Peter Zatka-Haas, Matteo Carandini, and Kenneth D Harris. Distributed coding of choice, action and engagement across the mouse brain. *Nature*, 576(7786):266–273, December 2019.

[88] Hessameddin Akhlaghpour, Joost Wiskerke, Jung Yoon Choi, Joshua P Taliaferro, Jennifer Au, and Ilana B Witten. Dissociated sequential activity and stimulus encoding in the dorsomedial striatum during spatial working memory. *Elife*, 5:e19507, 2016.

[89] Christopher D Harvey, Philip Coen, and David W Tank. Choice-specific sequences in parietal cortex during a virtual-navigation decision task. *Nature*, 484(7392):62–68, March 2012.

[90] Masashi Kondo, Kenta Kobayashi, Masamichi Ohkura, Junichi Nakai, and Masanori Matsuzaki. Two-photon calcium imaging of the medial prefrontal cortex and hippocampus without cortical invasion. *Elife*, 6, September 2017.

[91] Peter Dayan and Yael Niv. Reinforcement learning: the good, the bad and the ugly. *Curr. Opin. Neurobiol.*, 18(2):185–196, April 2008.

[92] John P O'Doherty, Peter Dayan, Karl Friston, Hugo Critchley, and Raymond J Dolan. Temporal difference models and reward-related learning in the human brain. *Neuron*, 38(2):329–337, April 2003.

[93] Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[94] J N Tsitsiklis and B Van Roy. An analysis of temporal-difference learning with function approximation, 1997.

[95] W Schultz, P Dayan, and P R Montague. A neural substrate of prediction and reward. *Science*, 275(5306):1593–1599, March 1997.

[96] Wolfram Schultz. Predictive reward signal of dopamine neurons. *J. Neurophysiol.*, 80(1):1–27, July 1998.

[97] M Kimura, M Kato, H Shimazaki, K Watanabe, and N Matsumoto. Neural information transferred from the putamen to the globus pallidus during learned movement in the monkey. *J. Neurophysiol.*, 76(6):3771–3786, December 1996.

[98] D E Oorschot. Total number of neurons in the neostriatal, pallidal, subthalamic, and substantia nigral nuclei of the rat basal ganglia: a stereological study using the cavalieri and optical disector methods. *J. Comp. Neurol.*, 366(4):580–599, March 1996.

[99] Jeremiah Y Cohen, Sebastian Haesler, Linh Vong, Bradford B Lowell, and Naoshige Uchida. Neuron-type-specific signals for reward and punishment in the ventral tegmental area. *Nature*, 482(7383):85–88, January 2012.

[100] Hao Li, Peter J Vento, Jeffrey Parrilla-Carrero, Dominika Pullmann, Ying S Chao, Maya Eid, and Thomas C Jhou. Three rostromedial tegmental afferents drive triply dissociable aspects of punishment learning and aversive valence encoding. *Neuron*, 104(5):987–999.e4, December 2019.

[101] Wulfram Gerstner, Marco Lehmann, Vasiliki Liakoni, Dane Corneil, and Johanni Brea. Eligibility traces and plasticity on behavioral time scales: Experimental support of NeoHebbian Three-Factor learning rules. *Front. Neural Circuits*, 12:53, July 2018.

[102] Hannah M Bayer and Paul W Glimcher. Midbrain dopamine neurons encode a quantitative reward prediction error signal, 2005.

[103] Ju Tian, Ryan Huang, Jeremiah Y Cohen, Fumitaka Osakada, Dmitry Kobak, Christian K Machens, Edward M Callaway, Naoshige Uchida, and Mitsuko Watabe-Uchida. Distributed and mixed information in monosynaptic inputs to dopamine neurons. *Neuron*, 91(6):1374–1389, September 2016.

[104] Matthew Botvinick, Sam Ritter, Jane X Wang, Zeb Kurth-Nelson, Charles Blundell, and Demis Hassabis. Reinforcement learning, fast and slow, 2019.

[105] Matthew Botvinick, Jane X Wang, Will Dabney, Kevin J Miller, and Zeb Kurth-Nelson. Deep reinforcement learning and its neuroscientific implications, 2020.

[106] Finale Doshi-Velez and George Konidaris. Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. *IJCAI*, 2016:1432–1440, July 2016.

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

[108] Hisham E Atallah, Dan Lopez-Paniagua, Jerry W Rudy, and Randall C O'Reilly. Separate neural substrates for skill learning and performance in the ventral and dorsal striatum. *Nat. Neurosci.*, 10(1):126–131, January 2007.

[109] Brian Lau and Paul W Glimcher. Value representations in the primate striatum during matching behavior. *Neuron*, 58(3):451–463, May 2008.

[110] John O'Doherty, Peter Dayan, Johannes Schultz, Ralf Deichmann, Karl Friston, and Raymond J Dolan. Dissociable roles of ventral and dorsal striatum in instrumental conditioning. *Science*, 304(5669):452–454, April 2004.

[111] Jocelyn M Richard, Frederic Ambroggi, Patricia H Janak, and Howard L Fields. Ventral pallidum neurons encode incentive value and promote Cue-Elicited instrumental actions. *Neuron*, 90(6):1165–1173, June 2016.

[112] Ken-Ichiro Tsutsui, Fabian Grabenhorst, Shunsuke Kobayashi, and Wolfram Schultz. A dynamic code for economic object valuation in prefrontal cortex neurons. *Nat. Commun.*, 7:12554, September 2016.

[113] Marc W Howard and Howard Eichenbaum. The hippocampus, time, and memory across scales, 2013.

[114] Matthew I Leon and Michael N Shadlen. Representation of time by neurons in the posterior parietal cortex of the macaque, 2003.

[115] Jeremy J Day, Robert A Wheeler, Mitchell F Roitman, and Regina M Carelli. Nucleus accumbens neurons encode pavlovian approach behaviors: evidence from an autoshaping paradigm. *Eur. J. Neurosci.*, 23(5):1341–1351, March 2006.

[116] Jeremy J Day and Regina M Carelli. The nucleus accumbens and pavlovian reward learning, 2007.

[117] Xun Wan and Laura L Peoples. Firing patterns of accumbal neurons during a pavlovian-conditioned approach task. *J. Neurophysiol.*, 96(2):652–660, August 2006.

[118] Courtney M Cameron, Malavika Murugan, Jung Yoon Choi, Esteban A Engel, and Ilana B Witten. Increased cocaine motivation is associated with degraded spatial and temporal representations in IL-NAc neurons, 2019.

[119] Aldo Genovesio, Peter J Brasted, and Steven P Wise. Representation of future and previous spatial goals by separate neural populations in prefrontal cortex. *J. Neurosci.*, 26(27):7305–7316, July 2006.

[120] Chung-Hay Luk and Jonathan D Wallis. Choice coding in frontal cortex during stimulus-guided or action-guided decision-making. *J. Neurosci.*, 33(5):1864–1871, January 2013.

[121] Michael J Siniscalchi, Hongli Wang, and Alex C Kwan. Enhanced population coding for rewarded choices in the medial frontal cortex of the mouse. *Cereb. Cortex*, 29(10):4090–4106, January 2019.

[122] Jung Hoon Sul, Hoseok Kim, Namjung Huh, Daeyeol Lee, and Min Whan Jung. Distinct roles of rodent orbitofrontal and medial prefrontal cortex in decision making, 2010.

[123] Rudolf N Cardinal and Timothy H C Cheung. Nucleus accumbens core lesions retard instrumental learning and performance with delayed reinforcement in the rat. *BMC Neurosci.*, 6:9, February 2005.

[124] Anne L Collins, Tara J Aitken, I-Wen Huang, Christine Shieh, Venuz Y Greenfield, Harold G Monbouquette, Sean B Ostlund, and Kate M Wassum. Nucleus accumbens cholinergic interneurons oppose cue-motivated behavior. *Biol. Psychiatry*, 86(5):388–396, February 2019.

[125] Pepe J Hernandez, Kenneth Sadeghian, and Ann E Kelley. Early consolidation of instrumental learning requires protein synthesis in the nucleus accumbens. *Nat. Neurosci.*, 5(12):1327–1331, December 2002.

[126] A E Kelley, S L Smith-Roe, and M R Holahan. Response-reinforcement learning is dependent on N-methyl-D-aspartate receptor activation in the nucleus accumbens core. *Proc. Natl. Acad. Sci. U. S. A.*, 94(22):12174–12179, October 1997.

[127] Hoseok Kim, Jung Hoon Sul, Namjung Huh, Daeyeol Lee, and Min Whan Jung. Role of striatum in updating values of chosen actions. *J. Neurosci.*, 29(47):14701–14712, November 2009.

[128] J D Salamone, R E Steinpreis, L D McCullough, P Smith, D Grebel, and K Mahan. Haloperidol and nucleus accumbens dopamine depletion suppress lever pressing for food but increase free food consumption in a novel food choice procedure, 1991.

[129] Silvia Maggi, Adrien Peyrache, and Mark D Humphries. An ensemble code in medial prefrontal cortex links prior events to outcomes during learning. *Nat. Commun.*, 9(1):2204, June 2018.

[130] Silvia Maggi and Mark D Humphries. Independent population coding of the past and the present in prefrontal cortex during learning. *bioRxiv*, page 668962, 2020.

[131] Eva Pastalkova, Vladimir Itskov, Asohan Amarasingham, and György Buzsáki. Internally generated cell assembly sequences in the rat hippocampus. *Science*, 321(5894):1322–1327, September 2008.

[132] Satoshi Terada, Yoshio Sakurai, Hiroyuki Nakahara, and Shigeyoshi Fujisawa. Temporal and rate coding for discrete event sequences in the hippocampus. *Neuron*, 94(6):1248–1262.e4, June 2017.

[133] Takashi Kawai, Hiroshi Yamada, Nobuya Sato, Masahiko Takada, and Masayuki Matsumoto. Roles of the lateral habenula and anterior cingulate cortex in negative outcome monitoring and behavioral adjustment in nonhuman primates. *Neuron*, 88(4):792–804, November 2015.

[134] Christina K Kim, Li Ye, Joshua H Jennings, Nandini Pichamoorthy, Daniel D Tang, Ai-Chi W Yoo, Charu Ramakrishnan, and Karl Deisseroth. Molecular and circuit-dynamical identification of top-down neural mechanisms for restraint of reward seeking. *Cell*, 170(5):1013–1027.e14, August 2017.

[135] Michael A Long, Dezhe Z Jin, and Michale S Fee. Support for a synaptic chain model of neuronal sequence generation. *Nature*, 468(7322):394–399, November 2010.

[136] Bence P Ölveczky, Timothy M Otchy, Jesse H Goldberg, Dmitriy Aronov, and Michale S Fee. Changes in the neural control of a complex motor sequence during learning. *J. Neurophysiol.*, 106(1):386–397, July 2011.

[137] Michel A Picardo, Josh Merel, Kalman A Katlowitz, Daniela Vallentin, Daniel E Okobi, Sam E Benezra, Rachel C Clary, Eftychios A Pnevmatikakis, Liam Paninski, and Michael A Long. Population-level representation of a temporal sequence underlying song production in the zebra finch. *Neuron*, 90(4):866–876, May 2016.

[138] Jon T Sakata, Cara M Hampton, and Michael S Brainard. Social modulation of sequence and syllable variability in adult birdsong. *J. Neurophysiol.*, 99(4):1700–1711, April 2008.

[139] Naoyuki Matsumoto, Takafumi Minamimoto, Ann M Graybiel, and Minoru Kimura. Neurons in the thalamic CM-Pf complex supply striatal neurons with information about behaviorally significant sensory events, 2001.

[140] Paolo Campus, Ignacio R Covelo, Youngsoo Kim, Aram Parsegian, Brittany N Kuhn, Sofia A Lopez, John F Neumaier, Susan M Ferguson, Leah C Solberg Woods, Martin Sarter, and Shelly B Flagel. The paraventricular thalamus is a critical mediator of top-down control of cue-motivated behavior in rats. *Elife*, 8, September 2019.

[141] James M Otis, Manhua Zhu, Vijay M K Namboodiri, Cory A Cook, Oksana Kosyk, Ana M Matan, Rose Ying, Yoshiko Hashikawa, Koichi Hashikawa, Ivan Trujillo-Pisanty, Jiami Guo, Randall L Ung, Jose Rodriguez-Romaguera, E S Anton, and Garret D Stuber. Paraventricular thalamus projection neurons integrate cortical and hypothalamic signals for cue-reward processing. *Neuron*, 103(3):277–290.e6, 2019.

[142] Yingjie Zhu, Gregory Nachtrab, Piper C Keyes, William E Allen, Liqun Luo, and Xiaoke Chen. Dynamic salience processing in paraventricular thalamus gates associative learning. *Science*, 362(6413):423–429, October 2018.

[143] Anne G E Collins and Jeffrey Cockburn. Beyond dichotomies in reinforcement learning. *Nat. Rev. Neurosci.*, 21(10):576–586, October 2020.

[144] Bradley B Doll, Dylan A Simon, and Nathaniel D Daw. The ubiquity of model-based reinforcement learning. *Curr. Opin. Neurobiol.*, 22(6):1075–1081, December 2012.

[145] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for Model-Based deep reinforcement learning with Model-Free Fine-Tuning, 2018.

[146] Steindór Sæmundsson, Katja Hofmann, and Marc Peter Deisenroth. Meta reinforcement learning with latent variable gaussian processes. *arXiv preprint arXiv:1803.07551*, 2018.

[147] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[148] Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. RL ^{2}: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

[149] Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pages 5331–5340. PMLR, 2019.

[150] M S Fee and J H Goldberg. A hypothesis for basal ganglia-dependent reinforcement learning in the songbird. *Neuroscience*, 198:152–170, December 2011.

[151] Dezhe Z Jin, Naotaka Fujii, and Ann M Graybiel. Neural representation of time in cortico-basal ganglia circuits, 2009.

[152] Mayank Aggarwal, Brian I Hyland, and Jeffery R Wickens. Neural control of dopamine neurotransmission: implications for reinforcement learning. *Eur. J. Neurosci.*, 35(7):1115–1123, April 2012.

[153] Luis Carrillo-Reid, Fatuel Tecuapetla, Dagoberto Tapia, Arturo Hernández-Cruz, Elvira Galarraga, René Drucker-Colin, and José Bargas. Encoding network states by striatal cell assemblies. *J. Neurophysiol.*, 99(3):1435–1450, March 2008.

[154] Samuel J Gershman, Ahmed A Moustafa, and Elliot A Ludvig. Time representation in reinforcement learning models of the basal ganglia. *Front. Comput. Neurosci.*, 7:194, January 2014.

[155] Adam Ponzi and Jeff Wickens. Sequentially switching cell assemblies in random inhibitory networks of spiking neurons in the striatum. *J. Neurosci.*, 30(17):5894–5911, April 2010.

[156] Hoseok Kim, Daeyeol Lee, and Min Whan Jung. Signals for previous goal choice persist in the dorsomedial, but not dorsolateral striatum of rats. *J. Neurosci.*, 33(1):52–63, January 2013.

[157] Kenji Doya. Metalearning and neuromodulation. *Neural Netw.*, 15(4-6):495–506, June 2002.

[158] Thomas E Hazy, Michael J Frank, and Randall C O'Reilly. Neural mechanisms of acquired phasic dopamine responses in learning. *Neurosci. Biobehav. Rev.*, 34(5):701–720, April 2010.

[159] Makoto Ito and Kenji Doya. Parallel representation of Value-Based and finite State-Based strategies in the ventral and dorsal striatum. *PLoS Comput. Biol.*, 11(11):e1004540, November 2015.

[160] Wei-Xing Pan, Robert Schmidt, Jeffery R Wickens, and Brian I Hyland. Dopamine cells respond to predicted events during classical conditioning: evidence for eligibility traces in the reward-learning network. *J. Neurosci.*, 25(26):6235–6242, June 2005.

[161] Roland E Suri and W Schultz. Learning of sequential movements by neural network model with dopamine-like reinforcement signal, 1998.

[162] R E Suri and W Schultz. A neural network model with dopamine-like reinforcement signal that learns a spatial delayed response task. *Neuroscience*, 91(3):871–890, 1999.

[163] Ruidong Chen, Pavel A Puzerey, Andrea C Roeser, Tori E Riccelli, Archana Podury, Kamal Maher, Alexander R Farhang, and Jesse H Goldberg. Songbird ventral pallidum sends diverse performance error signals to dopaminergic midbrain. *Neuron*, 103(2):266–276.e4, 2019.

[164] Kevin T Beier, Elizabeth E Steinberg, Katherine E DeLoach, Stanley Xie, Kazunari Miyamichi, Lindsay Schwarz, Xiaojing J Gao, Eric J Kremer, Robert C Malenka, and Liqun Luo. Circuit architecture of VTA dopamine neurons revealed by systematic Input-Output mapping. *Cell*, 162(3):622–634, July 2015.

[165] Daniel Fürth, Thomas Vaissière, Ourania Tzortzi, Yang Xuan, Antje Märtin, Iakovos Lazaridis, Giada Spigolon, Gilberto Fisone, Raju Tomer, Karl Deisseroth, Marie Carlén, Courtney A Miller, Gavin Rumbaugh, and Konstantinos Meletis. An interactive framework for whole-brain maps at cellular resolution. *Nat. Neurosci.*, 21(1):139–149, January 2018.

[166] P Thévenaz, U E Ruttimann, and M Unser. A pyramid approach to subpixel registration based on intensity. *IEEE Trans. Image Process.*, 7(1):27–41, 1998.

[167] Eftychios A Pnevmatikakis and Andrea Giovannucci. NoRMCorre: An online algorithm for piecewise rigid motion correction of calcium imaging data. *J. Neurosci. Methods*, 291:83–94, November 2017.

[168] Pengcheng Zhou, Shanna L Resendez, Jose Rodriguez-Romaguera, Jessica C Jimenez, Shay Q Neufeld, Andrea Giovannucci, Johannes Friedrich, Eftychios A Pnevmatikakis, Garret D Stuber, Rene Hen, Mazen A Kheirbek, Bernardo L Sabatini, Robert E Kass, and Liam Paninski. Efficient and accurate extraction of in vivo calcium signals from microendoscopic video data. *Elife*, 7, February 2018.

[169] P W Kalivas, L Churchill, and M A Klitenick. GABA and enkephalin projection from the nucleus accumbens and ventral pallidum to the ventral tegmental area. *Neuroscience*, 57(4):1047–1060, December 1993.

[170] Mitsuko Watabe-Uchida, Lisa Zhu, Sachie K Ogawa, Archana Vamanrao, and Naoshige Uchida. Whole-brain mapping of direct inputs to midbrain dopamine neurons. *Neuron*, 74(5):858–873, June 2012.

[171] P R Montague, P Dayan, and T J Sejnowski. A framework for mesencephalic dopamine systems based on predictive hebbian learning. *J. Neurosci.*, 16(5):1936–1947, March 1996.

[172] S Yagishita, A Hayashi-Takagi, G C R Ellis-Davies, H Urakubo, S Ishii, and H Kasai. A critical time window for dopamine actions on the structural plasticity of dendritic spines, 2014.

[173] Masayuki Matsumoto and Okihide Hikosaka. Two types of dopamine neuron distinctly convey positive and negative motivational signals. *Nature*, 459(7248):837–841, June 2009.

[174] Benjamin T Saunders, Jocelyn M Richard, Elyssa B Margolis, and Patricia H Janak. Dopamine neurons create pavlovian conditioned stimuli with circuit-defined motivational properties. *Nat. Neurosci.*, 21(8):1072–1083, August 2018.

[175] Moonsang Seo, Eunjeong Lee, and Bruno B Averbeck. Action selection and action value in frontal-striatal circuits. *Neuron*, 74(5):947–960, June 2012.

[176] Lung-Hao Tai, A Moses Lee, Nora Benavidez, Antonello Bonci, and Linda Wilbrecht. Transient stimulation of distinct subpopulations of striatal neurons mimics changes in action value. *Nat. Neurosci.*, 15(9):1281–1289, September 2012.

[177] Richard H R Hahnloser, Alexay A Kozhevnikov, and Michale S Fee. An ultra-sparse code underliesthe generation of neural sequences in a songbird, 2002.

[178] Alexay A Kozhevnikov and Michale S Fee. Singing-related activity of identified HVC neurons in the zebra finch. *J. Neurophysiol.*, 97(6):4271–4283, June 2007.

[179] Shanglin Zhou, Sotiris C Masmanidis, and Dean V Buonomano. Neural sequences as an optimal dynamical regime for the readout of time. *Neuron*, 108(4):651–658.e5, November 2020.

[180] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

[181] Ethan S Bromberg-Martin, Masayuki Matsumoto, Simon Hong, and Okihide Hikosaka. A pallidus-habenula-dopamine pathway signals inferred stimulus values. *J. Neurophysiol.*, 104(2):1068–1076, August 2010.

[182] Rachel S Lee, Marcelo G Mattar, Nathan F Parker, Ilana B Witten, and Nathaniel D Daw. Reward prediction error does not explain movement selectivity in DMS-projecting dopamine neurons, 2019.

[183] David J Ottenheimer, Bilal A Bari, Elissa Sutlief, Kurt M Fraser, Tabitha H Kim, Jocelyn M Richard, Jeremiah Y Cohen, and Patricia H Janak. A quantitative reward prediction error signal in the ventral pallidum. *Nat. Neurosci.*, 23(10):1267–1276, October 2020.

[184] Jessica Tooley, Lauren Marconi, Jason Bondoc Alipio, Bridget Matikainen-Ankney, Polymnia Georgiou, Alexxai V Kravitz, and Meaghan C Creed. Glutamatergic ventral pallidal neurons modulate activity of the Habenula–Tegmental circuitry and constrain reward seeking, 2018.

[185] Jack F Webster and Christian Wozny. Behavior: Local lateral habenula interneurons mediate aggression. *Curr. Biol.*, 30(16):R954–R956, August 2020.

[186] Hongbin Yang, Johannes W de Jong, Yeeun Tak, James Peck, Helen S Bateup, and Stephan Lammel. Nucleus accumbens subnuclei regulate motivated behavior via direct inhibition and disinhibition of VTA dopamine subpopulations. *Neuron*, 97(2):434–449.e4, January 2018.

[187] Nicholas J Edwards, Hugo A Tejeda, Marco Pignatelli, Shiliang Zhang, Ross A McDevitt, Jocelyn Wu, Caroline E Bass, Bernhard Bettler, Marisela Morales, and Antonello Bonci. Circuit specificity in the inhibitory architecture of the VTA regulates cocaine-induced behavior. *Nat. Neurosci.*, 20(3):438–448, March 2017.

[188] Vasileios Boulougouris, Anna Castañé, and Trevor W Robbins. Dopamine D2/D3 receptor agonist quinpirole impairs spatial reversal learning in rats: investigation of D3 receptor involvement in persistent behavior. *Psychopharmacology*, 202(4):611–620, March 2009.

[189] D A DeSteno and C Schmauss. A role for dopamine D2 receptors in reversal learning, 2009.

[190] Christoph Eisenegger. Role of dopamine D2 receptors in human reinforcement learning, 2016.

[191] P J Kruzich, S H Mitchell, A Younkin, and D K Grandy. Dopamine D2 receptors mediate reversal learning in male C57BL/6J mice, 2006.

[192] Shinae Kwak, Namjung Huh, Ji-Seon Seo, Jung-Eun Lee, Pyung-Lim Han, and Min W Jung. Role of dopamine D2 receptors in optimizing choice strategy in a dynamic and uncertain environment, 2014.

[193] Payam Piray. The role of dorsal striatal d2-like receptors in reversal learning: a reinforcement learning viewpoint. *J. Neurosci.*, 31(40):14049–14050, October 2011.

[194] Brian F Sadacca, Joshua L Jones, and Geoffrey Schoenbaum. Midbrain dopamine neurons compute inferred and cached value prediction errors in a common framework. *Elife*, 5, March 2016.

[195] M Zaki Jawaid, A Baidya, R Mahboubi-Ardakani, Richard L Davis, and Daniel L Cox. Sars-cov-2 omicron spike simulations: broad antibody escape, weakened ace2 binding, and modest furin cleavage. *Microbiology Spectrum*, 11(5):e01213–22, 2023.

[196] Salim S. Abdool Karim and Quarraisha Abdool Karim. Omicron sars-cov-2 variant: a new chapter in the covid-19 pandemic. *The Lancet*, 2021.

[197] Ginger Tsueng, Julia L. Mullen, Manar Alkuzweny, Marco Cano, Benjamin Rush, Emily Haag, Alaa Abdel Latif, Xinghua Zhou, Zhongchao Qian, Emory Hufbauer, Mark Zeller, Kristian G. Andersen, Chunlei Wu, Andrew I. Su, Karthik Gangavarapu, and Laura D. Hughes and. Outbreak.info research library: A standardized, searchable platform to discover and explore COVID-19 resources. *bioRxiv*, January 2022.

[198] Alexandra C Walls, Young-Jun Park, M Alejandra Tortorici, Abigail Wall, Andrew T McGuire, and David Veesler. Structure, function, and antigenicity of the sars-cov-2 spike glycoprotein. *Cell*, 181(2):281–292. e6, 2020.

[199] Ruben J. G. Hulswit, Yifei Lang, Mark J. G. Bakkers, Wentao Li, Zeshi Li, Arie Schouten, Bram Ophorst, Frank J. M. van Kuppeveld, Geert-Jan Boons, Berend-Jan Bosch, Eric G. Huizinga, and Raoul J. de Groot. Human coronaviruses oc43 and hku1 bind to 9-*o*-acetylated sialic acids via a conserved receptor-binding site in spike protein domain a. *Proceedings of the National Academy of Sciences*, 116(7):2681–2690, 2019.

[200] Petra Mlcochova, Steven A. Kemp, Mahesh Shanker Dhar, Guido Papa, Bo Meng, Isabella A. T. M. Ferreira, Rawlings Datir, Dami A. Collier, Anna Albecka, Sujeet Singh, Rajesh Pandey, Jonathan Brown, Jie Zhou, Niluka Goonawardane, Swapnil Mishra, Charles Whittaker, Thomas Mellan, Robin Marwal, Meena Datta, Shantanu Sengupta, Kalaiarasan Ponnusamy, Venkatraman Srinivasan Radhakrishnan, Adam Abdullahi, Oscar Charles, Partha Chattopadhyay, Priti Devi, Daniela Caputo, Tom Peacock, Chand Wattal, Neeraj Goel,

Ambrish Satwik, Raju Vaishya, Meenakshi Agarwal, Himanshu Chauhan, Tanzin Dikid, Hema Gogia, Hemlata Lall, Kaptan Verma, Mahesh Shanker Dhar, Manoj K. Singh, Namita Soni, Namonarayan Meena, Preeti Madan, Priyanka Singh, Ramesh Sharma, Rajeev Sharma, Sandhya Kabra, Sattender Kumar, Swati Kumari, Uma Sharma, Urmila Chaudhary, Sridhar Sivasubbu, Vinod Scaria, J. K. Oberoi, Reena Raveendran, S. Datta, Saumitra Das, Arindam Maitra, Sreedhar Chinnaswamy, Nidhan Kumar Biswas, Ajay Parida, Sunil K. Raghav, Punit Prasad, Apurva Sarin, Satyajit Mayor, Uma Ramakrishnan, Dasaradhi Palakodeti, Aswin Sai Narain Seshasayee, K. Thangaraj, Murali Dharan Bashyam, Ashwin Dalal, Manoj Bhat, Yogesh Shouche, Ajay Pillai, Priya Abraham, Varsha Atul Potdar, Sarah S. Cherian, Anita Sudhir Desai, Chitra Pattabiraman, M. V. Manjunatha, Reeta S. Mani, Gautam Arunachal Udupi, Vinay Nandicoori, Karthik Bharadwaj Tallapaka, Divya Tej Sowpati, Ryoko Kawabata, Nanami Morizako, Kenji Sadamasu, Hiroyuki Asakura, Mami Nagashima, Kazuhisa Yoshimura, Jumpei Ito, Izumi Kimura, Keiya Uriu, Yusuke Kosugi, Mai Suganami, Akiko Oide, Miyabishara Yokoyama, Mika Chiba, Akatsuki Saito, et al. Sars-cov-2 b.1.617.2 delta variant replication and immune evasion. *Nature*, 599(7883):114–119, 2021.

[201] Thomas P Peacock, Daniel H Goldhill, Jie Zhou, Laury Baillon, Rebecca Frise, Olivia C Swann, Ruthiran Kugathasan, Rebecca Penn, Jonathan C Brown, Raul Y Sanchez-David, et al. The furin cleavage site in the sars-cov-2 spike protein is required for transmission in ferrets. *Nature microbiology*, 6(7):899–909, 2021.

[202] Akatsuki Saito, Takashi Irie, Rigel Suzuki, Tadashi Maemura, Hesham Nasser, Keiya Uriu, Yusuke Kosugi, Kotaro Shirakawa, Kenji Sadamasu, Izumi Kimura, Jumpei Ito, Jiaqi Wu, Kiyoko Iwatsuki-Horimoto, Mutsumi Ito, Seiya Yamayoshi, Samantha Loeber, Masumi Tsuda, Lei Wang, Seiya Ozono, Erika P. Butlertanaka, Yuri L. Tanaka, Ryo Shimizu, Kenta Shimizu, Kumiko Yoshimatsu, Ryoko Kawabata, Takemasa Sakaguchi, Kenzo Tokunaga, Isao Yoshida, Hiroyuki Asakura, Mami Nagashima, Yasuhiro Kazuma, Ryosuke Nomura, Yoshihito Horisawa, Kazuhisa Yoshimura, Akifumi Takaori-Kondo, Masaki Imai, Mika Chiba, Hirotake Furihata, Haruyo Hasebe, Kazuko Kitazato, Haruko Kubo, Naoko Misawa, Nanami Morizako, Kohei Noda, Akiko Oide, Mai Suganami, Miyoko Takahashi, Kana Tsushima, Miyabishara Yokoyama, Yue Yuan, Shinya Tanaka, So Nakagawa, Terumasa Ikeda, Takasuke Fukuhara, Yoshihiro Kawaoka, Kei Sato, and Consortium The Genotype to Phenotype Japan. Enhanced fusogenicity and pathogenicity of sars-cov-2 delta p681r mutation. *Nature*, 2021.

[203] Adam Vaughan. Delta to dominate world. *New Scientist*, 250(3341):9, 2021.

[204] Milot Mirdita, Sergey Ovchinnikov, and Martin Steinegger. Colabfold - making protein folding accessible to all. *bioRxiv*, page 2021.08.15.456425, 2021.

[205] Richard Evans, Michael O'Neill, Alexander Pritzel, Natasha Antropova, Andrew Senior, Tim Green, Augustin Žídek, Russ Bates, Sam Blackwell, Jason Yim, Olaf Ronneberger, Sebastian Bodenstein, Michal Zielinski, Alex Bridgland, Anna Potapenko, Andrew Cowie, Kathryn Tunyasuvunakool, Rishub Jain, Ellen Clancy, Pushmeet Kohli, John Jumper, and Demis Hassabis. Protein complex prediction with alphafold-multimer. *bioRxiv*, page 2021.10.04.463034, 2021.

[206] Sheh-Yi Sheu, Dah-Yen Yang, H. L. Selzle, and E. W. Schlag. Energetics of hydrogen bonds in peptides. *Proceedings of the National Academy of Sciences*, 100(22):12683–12687, 2003.

[207] M. Zaki Jawaid, A. Baidya, S. Jakovcevic, J. Lusk, R. Mahboubi-Ardakani, N. Solomon, G. Gonzalez, J. Arsuaga, M. Vazquez, R.L. Davis, and D.L. Cox. Computational study of the furin cleavage domain of SARS-CoV-2: delta binds strongest of extant variants. *bioRxiv*, January 2022.

[208] Edward C. Holmes, Stephen A. Goldstein, Angela L. Rasmussen, David L. Robertson, Alexander Crits-Christoph, Joel O. Wertheim, Simon J. Anthony, Wendy S. Barclay, Maciej F. Boni, Peter C. Doherty, Jeremy Farrar, Jemma L. Geoghegan, Xiaowei Jiang, Julian L. Leibowitz, Stuart J. D. Neil, Tim Skern, Susan R. Weiss, Michael Worobey, Kristian G. Andersen, Robert F. Garry, and Andrew Rambaut. The origins of sars-cov-2: A critical review. *Cell*, 184(19):4848–4856, 2021.

[209] Guido Papa, Donna L. Mallery, Anna Albecka, Lawrence G. Welch, Jérôme Cattin-Ortolá, Jakub Luptak, David Paul, Harvey T. McMahon, Ian G. Goodfellow, Andrew Carter, Sean Munro, and Leo C. James. Furin cleavage of SARS-CoV-2 spike promotes but is not essential for infection and cell-cell fusion. *PLOS Pathogens*, 17(1):e1009246, January 2021.

[210] Donald J. Benton, Antoni G. Wrobel, Pengqi Xu, Chloë Roustan, Stephen R. Martin, Peter B. Rosenthal, John J. Skehel, and Steven J. Gamblin. Receptor binding and priming of the spike protein of sars-cov-2 for membrane fusion. *Nature*, 588(7837):327–330, 2020.

[211] Luigi Genovese, Marco Zaccaria, Michael Farzan, Welkin E Johnson, and Babak Momeni. Investigating the mutational landscape of the sars-cov-2 omicron variant via ab initio quantum mechanical modeling. *bioRxiv*, page 2021.12.01.470748, 2021.

[212] Suresh Kumar, Thiviya S Thambiraja, Kalimuthu Karuppanan, and Gunasekaran Subramaniam. Omicron and delta variant of sars-cov-2: A comparative computational study of spike protein. *bioRxiv*, page 2021.12.02.470946, 2021.

[213] Cecylia Severin Lupala, Yongjin Ye, Hong Chen, Xiaodong Su, and Haiguang Liu. Mutations in rbd of sars-cov-2 omicron variant result stronger binding to human ace2 protein. *bioRxiv*, page 2021.12.10.472102, 2021.

[214] Natalia Teruel, Matthew Crown, Matthew Bashton, and Rafael Najmanovich. Computational analysis of the effect of sars-cov-2 variant omicron spike protein mutations on dynamics, ace2 binding and propensity for immune escape. *bioRxiv*, page 2021.12.14.472622, 2021.

[215] Soumya Lipsa Rath, Aditya Kumar Padhi, and Nabanita Mandal. Scanning the rbd-ace2 molecular interactions in omicron variant. *bioRxiv*, page 2021.12.12.472253, 2021.

[216] Mert Golcuk, Ahmet Yildiz, and Mert Gur. The omicron variant increases the interactions of sars-cov-2 spike glycoprotein with ace2. *bioRxiv*, page 2021.12.06.471377, 2021.

[217] Filip Fratev. The high transmission of sars-cov-2 omicron (b.1.1.529) variant is not only due to its hace2 binding: A free energy of perturbation study. *bioRxiv*, page 2021.12.04.471246, 2021.

[218] Maren Schubert, Federico Bertoglio, Stephan Steinke, Philip Alexander Heine, Mario Alberto Ynga-Durand, Fanglei Zuo, Likun Du, Janin Korn, Marko Milošević, Esther Veronika Wenzel, Henrike Maass, Fran Krstanović, Saskia Polten, Marina Pribanić-Matešić, Ilija Brizić, Antonio Piralla, Fausto Baldanti, Lennart Hammarström, Stefan Dübel, Alan Šustić, Harold Marcotte, Monika Strengert, Alen Protić, Qiang Pan-Hammarström, Luka Čičin Šain, and Michael Hust. Human serum from sars-cov-2 vaccinated and covid-19 patients shows reduced binding to the rbd of sars-cov-2 omicron variant. *medRxiv*, page 2021.12.10.21267523, 2021.

[219] Bo Meng, Isabella Ferreira, Adam Abdullahi, Steven A. Kemp, Niluka Goonawardane, Guido Papa, Saman Fatihi, Oscar Charles, Dami Collier, Citiid-Nihr BioResource COVID-19 Collaboration, Consortium The Genotype to Phenotype Japan, Jinwook Choi, Joo Hyeon Lee, Petra Mlcochova, Leo James, Rainer Doffinger, Lipi Thukral, Kei Sato, and Ravindra K. Gupta. Sars-cov-2 omicron spike mediated immune escape, infectivity and cell-cell fusion. *bioRxiv*, page 2021.12.17.473248, 2021.

[220] Zeng Cong, John P. Evans, Panke Qu, Julia Faraone, Yi-Min Zheng, Claire Carlin, Joseph S. Bednash, Tongqing Zhou, Gerard Lozanski, Rama Mallampalli, Linda J. Saif, Eugene M. Oltz, Peter Mohler, Kai Xu, Richard J. Gumina, and Shan-Lu Liu. Neutralization and stability of sars-cov-2 omicron variant. *bioRxiv*, page 2021.12.16.472934, 2021.

[221] MC Chan. Hkumed finds omicron sars-cov-2 can infect faster and better than delta in human bronchus but with less severe infection in lung. *Brazilian Journal of Implantology and Health Sciences*, 4(1):50–54, 2022.

[222] Bailey Lubinski, Javier A Jaimes, and Gary R Whittaker. Intrinsic furin-mediated cleavability of the spike s1/s2 site from sars-cov-2 variant b. 1.1. 529 (omicron). *BioRxiv*, pages 2022–04, 2022.

[223] Thomas P. Peacock, Daniel H. Goldhill, Jie Zhou, Laury Baillon, Rebecca Frise, Olivia C. Swann, Ruthiran Kugathasan, Rebecca Penn, Jonathan C. Brown, Raul Y. Sanchez-David, Luca Braga, Maia Kavanagh Williamson, Jack A. Hassard, Ecco Staller, Brian Hanley, Michael Osborn, Mauro Giacca, Andrew D. Davidson, David A. Matthews, and Wendy S. Barclay. The furin cleavage site of sars-cov-2 spike protein is a key determinant for transmission due to enhanced replication in airway cells. *bioRxiv*, page 2020.09.30.318311, 2020.

[224] Bo Meng, Adam Abdullahi, Isabella A. T. M. Ferreira, Niluka Goonawardane, Akatsuki Saito, Izumi Kimura, Daichi Yamasoba, Pehuén Pereyra Gerber, Saman Fatihi, Surabhi Rathore, Samantha K. Zepeda, Guido Papa, Steven A. Kemp, Terumasa Ikeda, Mako Toyoda, Toong Seng Tan, Jin Kuramochi, Shigeki Mitsunaga, Takamasa Ueno, Kotaro Shirakawa, Akifumi Takaori-Kondo, Teresa Brevini, Donna L. Mallery, Oscar J. Charles, Stephen Baker, Gordon Dougan, Christoph Hess, Nathalie Kingston, Paul J. Lehner, Paul A. Lyons, Nicholas J. Matheson, Willem H. Ouwehand, Caroline Saunders, Charlotte Summers, James E. D. Thaventhiran, Mark Toshner, Michael P. Weekes, Patrick Maxwell, Ashley Shaw, Ashlea Bucke, Jo Calder, Laura Canna, Jason Domingo, Anne Elmer, Stewart Fuller, Julie Harris, Sarah Hewitt, Jane Kennet, Sherly Jose, Jenny Kourampa, Anne Meadows, Criona O'Brien, Jane Price, Cherry Publico, Rebecca Rastall, Carla Ribeiro, Jane Rowlands, Valentina Ruffolo, Hugo Tordesillas, Ben Bullman, Benjamin J. Dunmore, Stefan Gräf, Josh Hodgson, Christopher Huang, Kelvin Hunter, Emma Jones, Ekaterina Legchenko, Cecilia Matara, Jennifer Martin, Federica

Mescia, Ciara O'Donnell, Linda Pointon, Joy Shih, Rachel Sutcliffe, Tobias Tilly, Carmen Treacy, Zhen Tong, Jennifer Wood, Marta Wylot, Ariana Betancourt, Georgie Bower, Chiara Cossetti, Aloka De Sa, Madeline Epping, Stuart Fawke, Nick Gleadall, Richard Grenfell, Andrew Hinch, Sarah Jackson, Isobel Jarvis, Ben Krishna, Francesca Nice, Ommar Omarjee, Marianne Perera, Martin Potts, Nathan Richoz, Veronika Romashova, Luca Stefanucci, Mateusz Strezlecki, Lori Turner, et al. Altered tmprss2 usage by sars-cov-2 omicron impacts infectivity and fusogenicity. *Nature*, 603(7902):706–714, 2022.

[225] Georg Jocher, Vincent Grass, Sarah K Tschirner, Lydia Riepler, Stephan Breimann, Tuğberk Kaya, Madlen Oelsner, M Sabri Hamad, Laura I Hofmann, Carl P Blobel, Carsten B Schmidt-Weber, Ozgun Gokce, Constanze A Jakwerth, Jakob Trimpert, Janine Kimpel, Andreas Pichlmair, and Stefan F Lichtenthaler. Adam10 and adam17 promote sars-cov-2 cell entry and spike protein-mediated lung cell fusion. *EMBO reports*, 23(6):e54305, 2022.

[226] Tyler N. Starr, Allison J. Greaney, Sarah K. Hilton, Daniel Ellis, Katharine H. D. Crawford, Adam S. Dingens, Mary Jane Navarro, John E. Bowen, M. Alejandra Tortorici, Alexandra C. Walls, Neil P. King, David Veesler, and Jesse D. Bloom. Deep mutational scanning of sars-cov-2 receptor binding domain reveals constraints on folding and ace2 binding. *Cell*, 182(5):1295–1310.e20, 2020.

[227] N. Chapeshamano. Discovery health, south africa's largest private health insurance administrator, releases at-scale, real-world analysis of omicron outbreak based on 211 000 covid-19 test results in south africa, including collaboration with the south africa, 2021.

[228] Jun Lan, Jiwan Ge, Jinfang Yu, Sisi Shan, Huan Zhou, Shilong Fan, Qi Zhang, Xuanling Shi, Qisheng Wang, Linqi Zhang, and Xinquan Wang. Structure of the sars-cov-2 spike receptor-binding domain bound to the ace2 receptor. *Nature*, 581(7807):215–220, 2020.

[229] Yu Guo, Lisu Huang, Guangshun Zhang, Yanfeng Yao, He Zhou, Shu Shen, Bingqing Shen, Bo Li, Xin Li, Qian Zhang, Mingjie Chen, Da Chen, Jia Wu, Dan Fu, Xinxin Zeng, Mingfang Feng, Chunjiang Pi, Yuan Wang, Xingdong Zhou, Minmin Lu, Yarong Li, Yaohui Fang, Yun-Yueh Lu, Xue Hu, Shanshan Wang, Wanju Zhang, Ge Gao, Francisco Adrian, Qisheng Wang, Feng Yu, Yun Peng, Alexander G. Gabibov, Juan Min, Yuhui Wang, Heyu Huang, Alexey Stepanov, Wei Zhang, Yan Cai, Junwei Liu, Zhiming Yuan, Chen Zhang, Zhiyong Lou, Fei Deng, Hongkai Zhang, Chao Shan, Liang Schweizer, Kun Sun, and Zihe Rao. A sars-cov-2 neutralizing antibody with extensive spike binding coverage and modified for optimal therapeutic outcomes. *Nature Communications*, 12(1):2623, 2021.

[230] Sarah A. Clark, Lars E. Clark, Junhua Pan, Adrian Coscia, Lindsay G.A. McKay, Sundaresh Shankar, Rebecca I. Johnson, Anthony Griffiths, and Jonathan Abraham. Molecular basis for a germline-biased neutralizing antibody response to sars-cov-2. *bioRxiv*, page 2020.11.13.381533, 2020.

[231] Jiandong Huo, Yuguang Zhao, Jingshan Ren, Daming Zhou, Helen M. E. Duyvesteyn, Helen M. Ginn, Loic Carrique, Tomas Malinauskas, Reinis R. Ruza, Pranav N. M. Shah, Tiong Kit Tan, Pramila Rijal, Naomi Coombes, Kevin R. Bewley, Julia A. Tree, Julika Radecke, Neil G. Paterson, Piyada Supasa, Juthathip Mongkolsapaya,

Gavin R. Screaton, Miles Carroll, Alain Townsend, Elizabeth E. Fry, Raymond J. Owens, and David I. Stuart. Neutralization of sars-cov-2 by destruction of the prefusion spike. *Cell Host & Microbe*, 28(3):445–454.e6, 2020.

[232] Xiangyang Chi, Renhong Yan, Jun Zhang, Guanying Zhang, Yuanyuan Zhang, Meng Hao, Zhe Zhang, Pengfei Fan, Yunzhu Dong, Yilong Yang, Zhengshan Chen, Yingying Guo, Jinlong Zhang, Yaning Li, Xiaohong Song, Yi Chen, Lu Xia, Ling Fu, Lihua Hou, Junjie Xu, Changming Yu, Jianmin Li, Qiang Zhou, and Wei Chen. A neutralizing human antibody binds to the n-terminal domain of the spike protein of sars-cov-2. *Science*, 369(6504):650–655, 2020.

[233] AJ Venkatakrishnan, Praveen Anand, Patrick Lenehan, Rohit Suratekar, Bharathwaj Raghunathan, Michiel J Niesen, and Venky Soundararajan. Omicron variant of sars-cov-2 harbors a unique insertion mutation of putative viral or human genomic origin, Dec 2021.

[234] Elmar Krieger and Gert Vriend. Yasara view—molecular graphics for all devices—from smartphones to workstations. *Bioinformatics*, 30(20):2981–2982, 2014.

[235] James A Maier, Carmenza Martinez, Koushik Kasavajhala, Lauren Wickstrom, Kevin E Hauser, and Carlos Simmerling. ff14sb: improving the accuracy of protein side chain and backbone parameters from ff99sb. *Journal of chemical theory and computation*, 11(8):3696–3713, 2015.

[236] Junmei Wang, Romain M Wolf, James W Caldwell, Peter A Kollman, and David A Case. Development and testing of a general amber force field. *Journal of computational chemistry*, 25(9):1157–1174, 2004.

[237] Araz Jakalian, David B Jack, and Christopher I Bayly. Fast, efficient generation of high-quality atomic charges. am1-bcc model: Ii. parameterization and validation. *Journal of computational chemistry*, 23(16):1623–1641, 2002.

[238] Viktor Hornak, Robert Abel, Asim Okur, Bentley Strockbine, Adrian Roitberg, and Carlos Simmerling. Comparison of multiple amber force fields and development of improved protein backbone parameters. *Proteins: Structure, Function, and Bioinformatics*, 65(3):712–725, 2006.

[239] Ulrich Essmann, Lalith Perera, Max L Berkowitz, Tom Darden, Hsing Lee, and Lee G Pedersen. A smooth particle mesh ewald method. *The Journal of chemical physics*, 103(19):8577–8593, 1995.

[240] Elmar Krieger and Gert Vriend. New ways to boost molecular dynamics simulations. *Journal of computational chemistry*, 36(13):996–1007, 2015.

[241] Elmar Krieger, Roland L Dunbrack, Rob WW Hooft, and Barbara Krieger. Assignment of protonation states in proteins and ligands: Combining pk a prediction with hydrogen bonding network optimization. In *Computational Drug Discovery and Design*, pages 405–421. Springer, 2012.

[242] TJ Hou, JM Wang, YY Li, and W Wang. Assessing the performance of the mm/pbsa and mm/gbsa methods: I. the accuracy of binding free energy calculations based on molecular dynamics simulations. *Journal of Chemical Information & Modeling*, 51:69–82, 2001.

[243] Huiyong Sun, Youyong Li, Sheng Tian, Lei Xu, and Tingjun Hou. Assessing the performance of mm/pbsa and mm/gbsa methods. 4. accuracies of mm/pbsa and mm/gbsa methodologies evaluated by various simulation protocols using pdbbind data set. *Physical Chemistry Chemical Physics*, 16:16719–16729, 2014.

[244] F Chen, H Liu, HY Sun, PC Pan, YY Li, D Li, and TJ Hou. Assessing the performance of the mm/pbsa and mm/gbsa methods. 6. capability to predict protein-protein binding free energies and re-rank binding poses generated by protein-protein docking. *Physical Chemistry Chemical Physics*, 18:22129–22139, 2016.

[245] G Q Weng, E C Wang, Z Wang, H Liu, D Li, F Zhu, and T J Hou. Hawkdock: a web server to predict and analyze the structures of protein-protein complexes based on computational docking and mm/gbsa. *Nucleic Acids Research*, 47:W322–W330, 2019.

[246] Sven O. Dahms, Kornelia Hardes, Torsten Steinmetzer, and Manuel E. Than. X-ray structures of the proprotein convertase furin bound with substrate analogue inhibitors reveal substrate specificity determinants beyond the s4 pocket. *Biochemistry*, 57(6):925–934, 2018.

[247] Karthik Gangavarapu, Alaa Abdel Latif, Julia L Mullen, Manar Alkuzweny, Emory Hufbauer, Ginger Tsueng, Emily Haag, Mark Zeller, Christine M Aceves, Karina Zaiets, et al. Outbreak. info genomic reports: scalable and dynamic surveillance of sars-cov-2 variants and mutations. *Nature Methods*, 20(4):512–522, 2023.

[248] M Zaki Jawaid, A Baidya, S Jakovcevic, J Lusk, R Mahboubi-Ardakani, N Solomon, G Gonzalez, J Arsuaga, M Vazquez, RL Davis, et al. Computational study of the furin cleavage domain of sars-cov-2: delta binds strongest of extant variants. *bioRxiv*, pages 2022–01, 2022.

[249] Edward C Holmes, Stephen A Goldstein, Angela L Rasmussen, David L Robertson, Alexander Crits-Christoph, Joel O Wertheim, Simon J Anthony, Wendy S Barclay, Maciej F Boni, Peter C Doherty, et al. The origins of sars-cov-2: A critical review. *Cell*, 184(19):4848–4856, 2021.

[250] Yiran Wu and Suwen Zhao. Furin cleavage sites naturally occur in coronaviruses. *Stem Cell Research*, 50:102115, 2021.

[251] Panita Decha, Thanyada Rungrotmongkol, Pathumwadee Intharathep, Maturos Malaisree, Ornjira Aruksakunwong, Chittima Laohpongspaisan, Vudhichai Parasuk, Pornthep Sompornpisut, Somsak Pianwanit, Sirirat Kokpol, and Supot Hannongbua. Source of high pathogenicity of an avian influenza virus h5n1: Why h5 is better cleaved by furin. *Biophysical Journal*, 95(1):128–134, 2008.

[252] Stefan Henrich, Angus Cameron, Gleb P. Bourenkov, Reiner Kiefersauer, Robert Huber, Iris Lindberg, Wolfram Bode, and Manuel E. Than. The crystal structure of the proprotein processing proteinase furin explains its stringent specificity. *Nature Structural & Molecular Biology*, 10(7):520–526, 2003.

[253] Magdalena M. Kacprzak, Juan R. Peinado, Manuel E. Than, Jon Appel, Stefan Henrich, Gregory Lipkind, Richard A. Houghten, Wolfram Bode, and Iris Lindberg. Inhibition of furin by polyarginine-containing peptides: Nanomolar inhibition by nona-¡span class=”small”¿d¡/span¿-arginine *. *Journal of Biological Chemistry*, 279(35):36788–36794, 2004.

[254] Naveen Vankadari. Structure of furin protease binding to sars-cov-2 spike glycoprotein and implications for potential targets and virulence. *Journal of Physical Chemistry Letters*, 11(16):6655–6663, 2020.

[255] Sun Tian, Qingsheng Huang, Ying Fang, and Jianhua Wu. Furindb: A database of 20-residue furin cleavage site motifs, substrates and their associated drugs. *International Journal of Molecular Sciences*, 12(2), 2011.

[256] S. Khare, C. Gurry, L. Freitas, M.B. Schultz, Bach G., A. Diallo, N. Akite, Lee R.T.C., W. Yeo, GISAID Core Curation Team, and S. Maurer-Stroh. Gisaid's role in pandemic response. *China CDC Weekly*, 3:1049–1051, 2021.

[257] Stefan Elbe and Gemma Buckland-Merrett. Data, disease and diplomacy: Gisaid's innovative contribution to global health. *Global Challenges*, 1(1):33–46, 2017.

[258] Yuelong Shu and John McCauley. Gisaid: Global initiative on sharing all influenza data – from vision to reality. *Eurosurveillance*, 22(13):30494, 2017.

[259] Eric W. Sayers, Evan E. Bolton, J. Rodney Brister, Kathi Canese, Jessica Chan, Donald C Comeau, Ryan Connor, Kathryn Funk, Chris Kelly, Sunghwan Kim, Tom Madej, Aron Marchler-Bauer, Christopher Lanczycki, Stacy Lathrop, Zhiyong Lu, Francoise Thibaud-Nissen, Terence Murphy, Lon Phan, Yuri Skripchenko, Tony Tse, Jiyao Wang, Rebecca Williams, Barton W Trawick, Kim D Pruitt, and Stephen T Sherry. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, page gkab1112, 2021.

[260] Elisabeth Braun and Daniel Sauter. Furin-mediated protein processing in infectious diseases and cancer. *Clinical & Translational Immunology*, 8(8):e1073, 2019.

[261] Bailey Lubinski, Javier A. Jaimes, and Gary R. Whittaker. Intrinsic furin-mediated cleavability of the spike s1/s2 site from sars-cov-2 variant b.1.529 (omicron). *bioRxiv*, page 2022.04.20.488969, 2022.

[262] Y Shu and J McCauley. Gisaid: Global initiative on sharing all influenza data – from vision to reality. *Euro-Surveillance*, 22, 2017.

[263] Fabian Sievers and Desmond G Higgins. Clustal omega for making accurate alignments of many protein sequences. *Protein Science*, 27:135–145, 2018.

[264] J D Thompson, D G Higgins, and T J Gibson. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22:4673–4680, 1994.

[265] Robert C. Edgar. Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32:1792–1797, 2004.

[266] F Wu, S Zhao, B Yu, Y M Chen, W Wang, Z G Song, Y Hu, Z W Tao, J H Tian, Y Y Pei, M L Yuan, Y L Zhang, F H Dai, Y Liu, Q M Wang, J J Zhengand L Xu, E C Holmes, and Y Z Zhang. A new coronavirus associated with human respiratory disease in china. *Nature*, 579:265–269, 2020.

[267] Alexandra C. Walls, Young-Jun Park, M. Alejandra Tortorici, Abigail Wall, Andrew T. McGuire, and David Veesler. Structure, function, and antigenicity of the sars-cov-2 spike glycoprotein. *Cell*, 181(2):281–292.e6, 2020.

[268] Avinash Baidya, Joel Dapello, James J DiCarlo, and Tiago Marques. Combining different v1 brain model variants to improve robustness to image corruptions in cnns. *arXiv preprint arXiv:2110.10645*, 2021.

[269] Alex Krizhevsky, Ilya Sutskever, and Hinton Geoffrey E. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, pages 1097–1105, 2012. ISSN: 10495258.

[270] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:1–9, 2015. ISBN: 9781467369640.

[271] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, pages 1–14, 2015. ISSN: 15352900.

[272] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pages 1–12, December 2016.

[273] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[274] Asa Cooper Stickland and Iain Murray. Diverse Ensembles Improve Calibration. *arXiv:2007.04206 [cs, stat]*, July 2020. arXiv: 2007.04206.

[275] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift. *arXiv:1906.02530 [cs, stat]*, December 2019. arXiv: 1906.02530.

[276] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *arXiv:1612.01474 [cs, stat]*, November 2017. arXiv: 1612.01474.

[277] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, pages 4970–4979. PMLR, 2019.

[278] Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*, 2020.

[279] Christian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model Compression. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[280] Zeyuan Allen-Zhu and Yuanzhi Li. Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning. *arXiv:2012.09816 [cs, math, stat]*, July 2021. arXiv: 2012.09816.

[281] Xu Lan, Xiatian Zhu, and Shaogang Gong. Knowledge distillation by on-the-fly native ensemble. *Advances in neural information processing systems*, 31, 2018.

[282] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. *Advances in neural information processing systems*, 30, 2017.

[283] Antonio Greco, Alessia Saggese, Mario Vento, and Vincenzo Vigilante. Effective training of convolutional neural networks for age estimation based on knowledge distillation. *Neural Computing and Applications*, April 2021.

[284] Jia Cui, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, Tom Sercu, Kartik Audhkhasi, Abhinav Sethy, Markus Nussbaum-Thom, and Andrew Rosenberg. Knowledge distillation across ensembles of multilingual models for low-resource languages. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4825–4829, 2017.

[285] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Noise as a Resource for Learning in Knowledge Distillation. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3128–3137, Waikoloa, HI, USA, January 2021. IEEE.

[286] Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. Hyperparameter Ensembles for Robustness and Uncertainty Quantification. *arXiv:2006.13570 [cs, stat]*, January 2021. arXiv: 2006.13570.

[287] Sheheryar Zaidi, Arber Zela, Thomas Elsken, Chris Holmes, Frank Hutter, and Yee Whye Teh. Neural Ensemble Search for Uncertainty Estimation and Dataset Shift. *arXiv:2006.08573 [cs, stat]*, June 2021. arXiv: 2006.08573.

[288] Dario L Ringach, Robert M Shapley, and Michael J Hawken. Orientation selectivity in macaque V1: diversity and laminar dependence. *The Journal of Neuroscience*, 22(13):5639–5651, 2002. ISBN: 1529-2401 (Electronic)\n0270-6474 (Linking).

[289] Yang Liu, Sheng Shen, and Mirella Lapata. Noisy Self-Knowledge Distillation for Text Summarization. *arXiv:2009.07032 [cs]*, July 2021. arXiv: 2009.07032.

[290] Martin Schrimpf, Jonas Kubilius, Ha Hong, Najib J. Majaj, Rishi Rajalingham, Elias B. Issa, Kohitij Kar, Pouya Bashivan, Jonathan Prescott-Roy, Kailyn Schmidt, Daniel L. K. Yamins, and James J. DiCarlo. Brain-Score: Which Artificial Neural Network for Object Recognition is most Brain-Like? *bioRxiv*, 2018.

[291] Tiago Marques, Martin Schrimpf, and James J Dicarlo. Multi-scale hierarchical neural network models that bridge from single neurons in the primate primary visual cortex to object recognition behavior. *bioRxiv*, 2021.

[292] Rishi Rajalingham, Elias B. Issa, Pouya Bashivan, Kohitij Kar, Kailyn Schmidt, and James J. DiCarlo. Large-scale, high-resolution comparison of the core visual object recognition behavior of humans, monkeys, and state-of-the-art deep artificial neural networks. *The Journal of Neuroscience*, 38(33):7255–7269, 2018. ISBN: 0011-4162.