Using network control theory to study the dynamics of the structural connectome

Linden Parkes*,[1,2,3] Jason Z. Kim*,[4] Jennifer Stiso,[1] Julia K. Brynildsen,[1] Matthew Cieslak,[2,5,6] Sydney Covitz,[2,5,6] Raquel E. Gur,[5,6] Ruben C. Gur,[5,6] Fabio Pasqualetti,[7] Russell T. Shinohara,[8,9,10] Dale Zhou,[1] Theodore D. Satterthwaite,[2,5,6,9] and Dani S. Bassett[1,6,11,12,13,14]

[1]*Department of Bioengineering, University of Pennsylvania, PA 19104, USA*
[2]*Lifespan Informatics and Neuroimaging Center (PennLINC), Department of Psychiatry, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, USA*
[3]*Department of Psychiatry, Rutgers University, Piscataway, NJ 08854, USA*
[4]*Department of Physics, Cornell University, Ithaca, NY 14853, USA*
[5]*Penn/CHOP Lifespan Brain Institute, Perelman School of Medicine, Children's Hospital of Philadelphia Research Institute, Philadelphia, PA 19104, USA*
[6]*Department of Psychiatry, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, USA*
[7]*Department of Mechanical Engineering, University of California, Riverside, Riverside, CA 92521, USA*
[8]*Department of Biostatistics, Epidemiology, and Informatics, Perelman School of Medicine, Philadelphia, PA 19104, USA*
[9]*Center for Biomedical Image Computation and Analytics, University of Pennsylvania, Philadelphia, PA 19104, USA*
[10]*Penn Statistics in Imaging and Visualization Endeavor (PennSIVE), Center for Clinical Epidemiology and Biostatistics, Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA 19104, USA*
[11]*Department of Neurology, Perelman School of Medicine, Philadelphia, PA 19104, USA*
[12]*Department of Electrical and Systems Engineering, University of Pennsylvania, PA 19104, USA*
[13]*Department of Physics and Astronomy, University of Pennsylvania, PA 19104, USA*
[14]*Santa Fe Institute, Santa Fe, NM 87501, USA*
* *These authors contributed equally*

Network control theory (NCT) is a simple and powerful tool for studying how network topology informs and constrains dynamics. Compared to other structure-function coupling approaches, the strength of NCT lies in its capacity to predict the patterns of external control signals that may alter dynamics in a desired way. We have extensively developed and validated the application of NCT to the human structural connectome. Through these efforts, we have studied (i) how different aspects of connectome topology affect neural dynamics, (ii) whether NCT outputs cohere with empirical data on brain function and stimulation, and (iii) how NCT outputs vary across development and correlate with behavior and mental health symptoms. In this protocol, we introduce a framework for applying NCT to structural connectomes following two main pathways. Our primary pathway focuses on computing the *control energy* associated with transitioning between specific neural activity states. Our second pathway focuses on computing *average controllability*, which indexes nodes' general capacity to control dynamics. We also provide recommendations for comparing NCT outputs against null network models. Finally, we support this protocol with a Python-based software package called *network control theory for python (nctpy)*.

I. INTRODUCTION

Network neuroscience is principally concerned with studying the connectome [1], the description of whole brain connectivity. This connectome is often encoded as a graph of nodes interconnected by edges that can be defined across multiple scales, species, and data modalities [2, 3]. In any case, these descriptions of brain connectivity give rise to complex topology—including hubs, modules, small-worldness, and core-periphery structure [4]—and understanding how this topology shapes and constrains the brain's rich repertoire of dynamics is a central goal of network neuroscience.

Network control theory (NCT) provides an approach to studying these dynamics that yields insights into how they emerge from the topology of the underlying structural connectome [5–8]. The application of NCT has revolutionized both the understanding and design of complex networks in contexts as diverse as space and terrestrial exploration, as well as modeling of financial markets, airline networks, and fire-control systems. Briefly, NCT assumes that inter-nodal communication follows a linear model of diffusion, where activity from one set of nodes (i.e., an initial *state*) spreads across the network over time along a series of fronts [4, 9]. Then, upon this dynamical system, NCT models a set of external control signals that are designed to guide these diffusing activity patterns towards a chosen target *state*. This choice can be informed by a measurement of activity evoked by behavior, spontaneous activity, or the type of brain system. These control signals are found by minimizing the total magnitude of their input over a given time horizon; that is, they are designed to achieve a desired state transition with the lowest amount of *control energy*. Once modeled, these control signals can be examined to determine to what extent, and how, they were constrained by topology, thus allowing researchers to study how the connectome might be leveraged to control dynamics.

Recently, we have developed and tested the application of NCT to brain network data across multiple contexts, scales, and definitions of connectivity [10–20]. Here, we present a protocol for applying NCT to two different structural connectomes: one defined using undirected connectivity estimated in the human brain [21, 22] and the other using directed connectivity estimated in the mouse brain [23–25]. Briefly, we detail two common applications of NCT that we—as well as other groups [26–32]—have deployed that focus on (i) quantifying the amount of energy that is required to complete transitions between specific brain states (Figure 1) and (ii) examining regions' general capacity to control dynamics (Figure 2). The former approach is useful for researchers interested in examining how dynamics can be controlled to move from one place on the network to another, while the latter is useful for researchers interested in analyzing topographic maps of control. Additionally, we provide recommendations for visualization of model outputs, and discuss the use of null network models to examine which topological features affect model outputs.

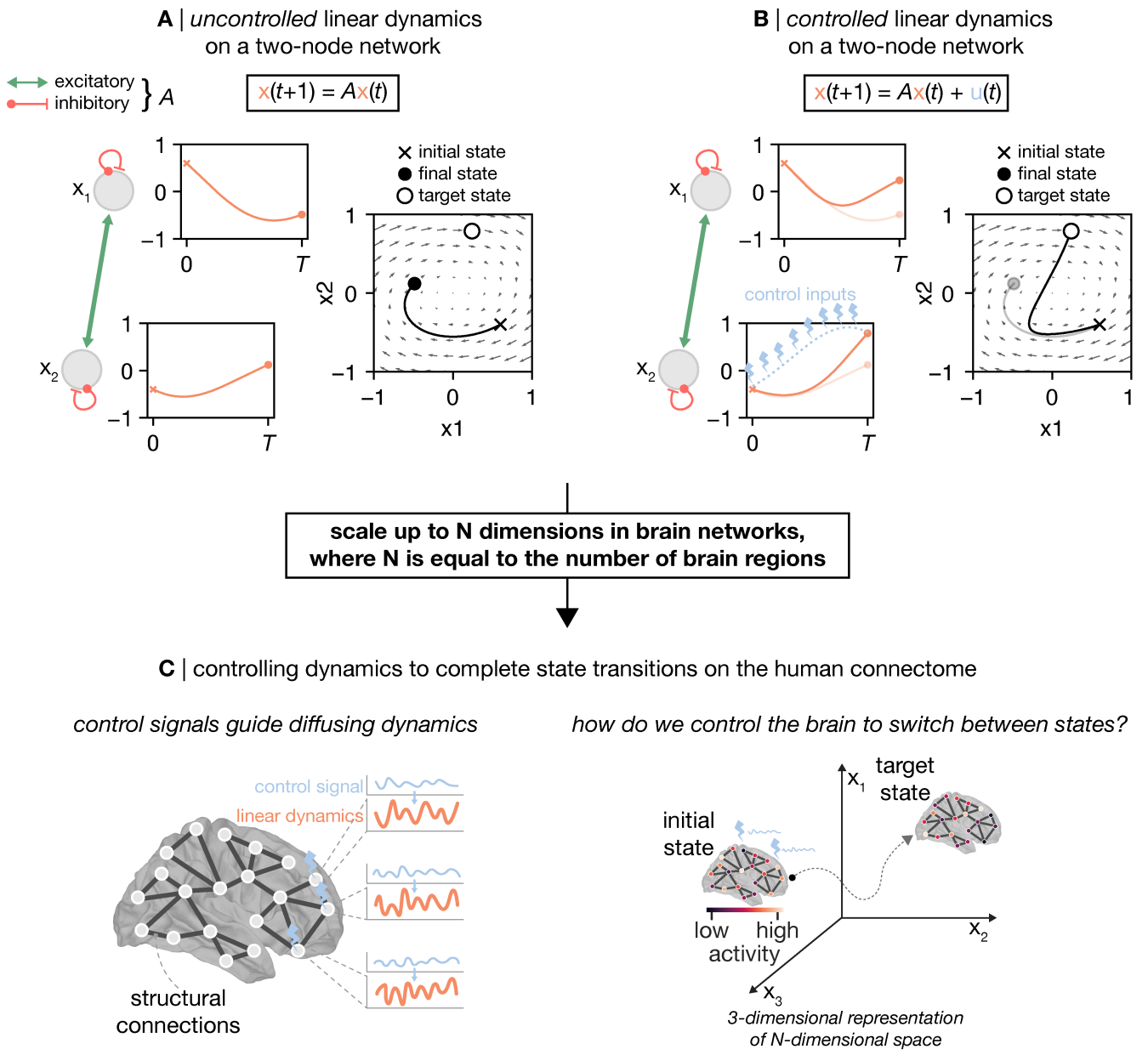

FIG. 1

Modeling the *control energy* required to complete a state transition. Network control theory (NCT) finds the control signals that, when injected into a networked system, will guide simulated neural activity from an *initial state* to a *target state*. Here, we show a two-node toy network (x_1, x_2) that illustrates the difference between neural activity (solid orange lines) in the absence (**A**) and presence (**B**) of a control signal (dashed blue line). **A**, Uncontrolled linear dynamics on a two-node network. Given an initial state where $x_1 = 0.3$ and $x_2 = -0.2$, as well as coupling between nodes encoded by A, uncontrolled neural activity unfolds as shown on the left. These dynamic trajectories can also be represented in two dimensions as a vector field as shown on the right. Under this uncontrolled regime, the state of the system culminates in $x_1 = -0.24$ and $x_2 = 0.06$ at time T. **B**, Controlled linear dynamics on a two-node network. By contrast, when we introduce a control signal to x_2, the trajectory changes to now culminate in $x_1 = 0.12$ and $x_2 = 0.39$ at time T. Thus, NCT has found the control signal that drove our system from our *initial state* [$x_1 = 0.3, x_2 = -0.2$] to our desired *target state* [$x_1 = 0.12, x_2 = 0.39$]. **C**, NCT applied to the human connectome. The above model can be extended to the scale of N brain regions that constitute a whole-brain connectome (left). In doing so, we can model and examine the control signals required to transition the brain between various states of interest (right).

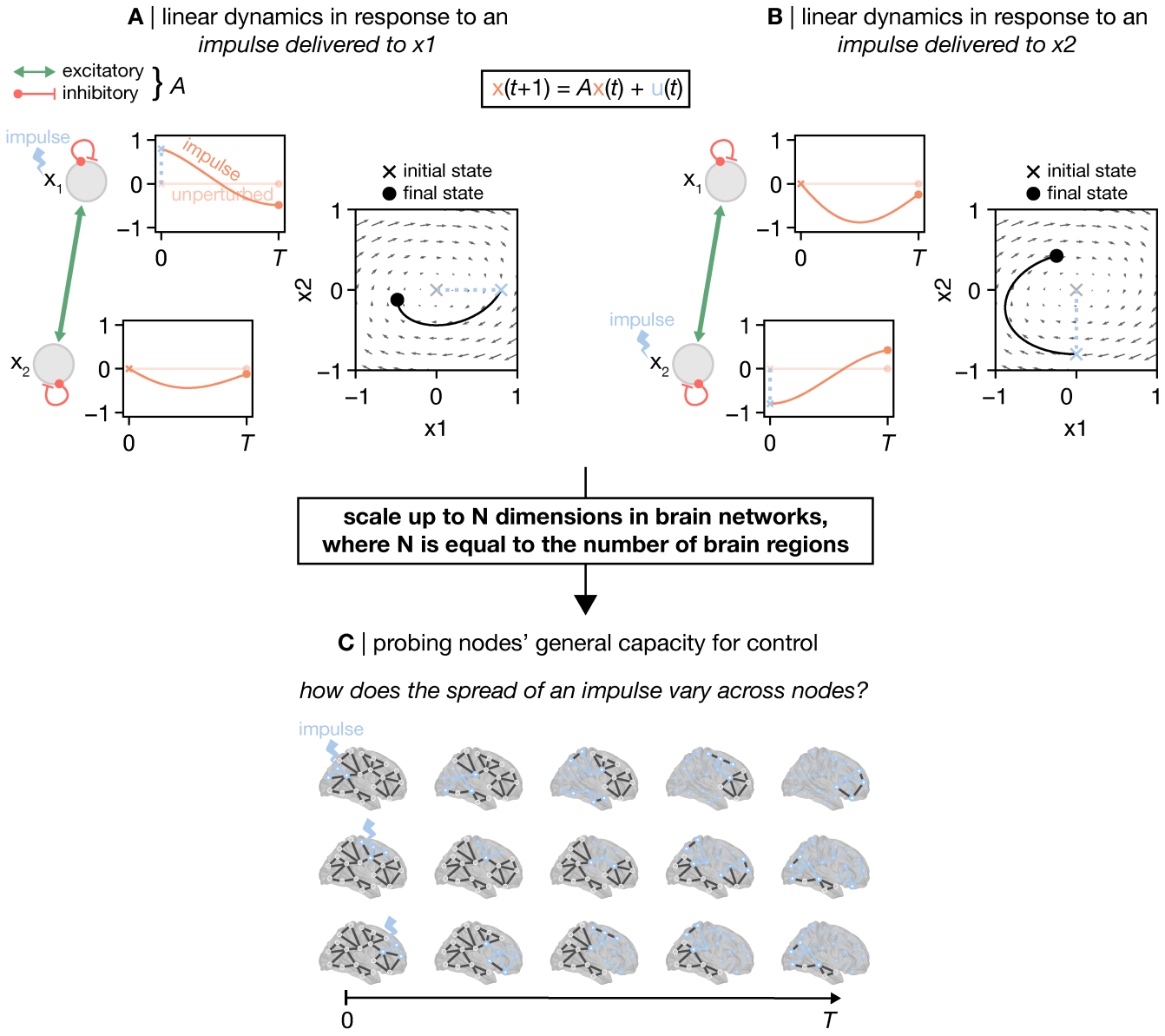

FIG. 2
Average controllability: modeling the impulse response of the system from each node. Network control theory (NCT) can be used to probe regions' general capacity to control dynamics. This is achieved by studying how the system responds to an impulse delivered to each node. Here, we show a two-node toy network (x_1, x_2) coupled by A. Upon this network, we demonstrate how neural activity (solid orange lines) unfolds when an impulse (dashed blue line) is delivered to x_1 (**A**) and x_2 (**B**). **A**, An impulse is delivered to x_1 that sets the initial state of the system to $[x_1 = 0.4, x_2 = 0]$. **B**, An impulse is delivered to x_2 that sets the initial state of the system to $[x_1 = 0, x_2 = -0.4]$. In each case, the impulse response of the system is quantified as the area under the squared curves of the two orange traces. Intuitively, this measurement corresponds to the amount of activity propagated throughout the system over time. We refer to this measure as the *average controllability*. Thus, for a given time horizon (T), a region with higher *average controllability* is better able to broadcast an impulse. **C**, Impulse response modeled for the human connectome. The above model can be extended to the scale of N brain regions that constitute a whole-brain connectome. In doing so, we can compare each region's capacity to broadcast an impulse across the whole brain.

II. DEVELOPMENT OF THE PROTOCOL

The methods that underlie NCT are based on the established fields of control theory and dynamical systems theory. Dating back to at least the 19th century [33], control theory is primarily concerned with engineering perturbations to achieve desired behaviors in the states of a system, and specifically the evolution of such states over *time*. Hence, one of the most natural ways to formulate theories of control is through *differential* and *difference* equations that mathematically define the next state of a

system given its current state. A common example of a control system is an inverted pendulum on a cart: the system states are the positions and velocities of the cart and pendulum, the differential equations are determined through the governing Newtonian physics, and the control task is to perturb the cart so that the cart and pendulum end up in a desired state. For example, one might want to push the cart back and forth in such a way as to stabilize the pendulum so that it remains upright [34].

From one perspective, the inverted pendulum is not unlike the brain, where the system states are the activities of neural units (e.g., brain regions), the differential equations are determined through the diffusion of activity through structural connections between those units, and the control task is to perturb the brain to steer it to a desired state. There is a rich history of such modeling of the brain as a dynamical system using differential equations, ranging from biophysical models of single neurons [35], to phenomenological [36] and coarse-grained [37] models of neural populations. In tandem, there is a very practical translational need to understand how to control brain dynamics [38], for example to compensate for abnormal dynamics that may be present in neurological and neuropsychiatric disorders.

However, there are also many ways in which the brain is not like an inverted pendulum. First is the dimensionality and complexity of the brain. Understanding how the topology of the structural connectome gives rise to brain function is a difficult task that has motivated a large body of work in the last two decades. This research has revealed that structure-function coupling is not one-to-one, varies spatially across the cortex [39–43], and is stronger when indirect structural pathways are accounted for under multiple models of network communication [44, 45]. Second is the distributed nature of brain states for human function. While some brain regions may be thought of as supporting specific functions (e.g., the fusiform face area), carrying out complex human functions typically requires the recruitment of a network of brain regions to a distributed brain state [46]. Finally, biology imposes relatively tight operating constraints. To support complex human function, the brain needs to optimize for efficient signaling while balancing the need to minimize wiring cost within the spatial constraints of the cranial cavity. Hence, there is a need to express the unique complexities and constraints of controlling brain structure-function coupling in the quantitative formalism of dynamics and control.

NCT emerges as a flexible framework that is methodologically based in optimal control theory [47], and can accommodate a wide range of theoretical and experimental hypotheses and constraints about structure-function coupling through a consistent mathematical framework [20, 48, 49]. Because NCT posits a model of neural dynamics at the level of individual neural interactions, it allows us to probe the role of the complex structural connectome on brain function at the level of those interactions [14, 50, 51]. Additionally, because NCT parameterizes which regions to control and how, as well as the precise patterns of initial and target neural activity, it can answer questions ranging from the importance of a single region for propagating activity [10] to the cost of transitioning between specific brain states [11]. Hence, the development of NCT has largely served to provide a simple, first-order biophysical model with the flexibility and power to study more advanced hypotheses of function.

The modeling framework of NCT comprises N nodes (e.g., neurons, brain regions) and m inputs, and stipulates that the state of each node, $x_i(t)$, evolves in time as a weighted sum of the state of all upstream nodes, $x_j(t)$, and any inputs, $u_m(t)$. If the evolution of the states can be framed in terms of *states*—where the activity of upstream nodes determines the state of downstream nodes at the next *discrete* point in time—then the model takes the form of a difference equation,

$$\boldsymbol{x}(t+1) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t), \tag{1}$$

where $\boldsymbol{x}(t) = [x_1(t), x_2(t), \cdots, x_n(t)]^\top$ is the vector of neural states, A is the $N \times N$ connectome, $\boldsymbol{u}(t) = [u_1(t), u_2(t), \cdots, u_m(t)]^\top$ is the vector of independent control signals, and B is the $N \times m$ matrix that quantifies how each input affects the nodes. If instead the evolution of the states can be framed in terms of *rates*—where the activity of upstream nodes affects the *continuous* rate at which the state of downstream nodes change—then the model takes the form of the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t). \tag{2}$$

While these two models appear similar at first glance, their definition, properties, and behavior differ substantially. In turn, the interpretation of the model parameters and outputs can vary dramatically between them.

This protocol will discuss two common operationalizations of NCT that can be derived from either of these models. The first employs a time-varying perturbation, $\boldsymbol{u}(t)$, to drive the neural activity, $\boldsymbol{x}(t)$, from an initial state, $x0$, to a target state, xf, given a balance of constraints on the magnitude of both the neural states and the perturbations. The magnitude of these perturbations are summarized as the *control energy*, which we interpret as the amount of effort that the model system must exert to complete a given state transition. The second is *average controllability*, which is the magnitude of the neural activity, $\boldsymbol{x}(t)$, in response to an impulse stimulus delivered to a single node; a node with higher *average controllability* is better able to leverage graph topology to spread an impulse throughout the system. Note, *average controllability* is only one example of a node-level NCT metric that falls within the broader category of *controllability statistics*. This category encompasses NCT outputs that describe different ways in which the nodes of the system may control its dynamics. While we have used other controllability statistics in our previous work (e.g., *modal controllability*, see section III), we focus on *average controllability* in this protocol owing to its simple intuition and broad appeal.

We focus on these two operationalizations—*control energy* and *average controllability*—because they encompass two common sets of questions about the brain. The first set of questions stems from advances in neuroimaging that allow us to empirically

measure neural states in many forms such as functional Magnetic Resonance Imaging (fMRI), electrophysiology, and calcium imaging [52]. Given these state-level empirical data, a natural question is "how does the brain reach or switch between these states using regimes of internal or external control?" Optimal control theory provides a powerful and flexible set of tools to explore these questions under various constraints and at different spatio-temporal scales. The second set of questions stems from empirical evidence demonstrating that individual and groups of brain regions may be important for (i) enabling specific functions, such as visual processing [53], motor processing [54], and cognition [55, 56]; (ii) supporting critical functional processes in the brain, such as segregation and integration [57, 58]; and (iii) may be disproportionately impacted by diseases processes [59, 60]. Given these data, a natural question to ask is "what is the contribution of these sets of regions to the control of brain activity?" Average controllability measures the magnitude of propagation of stimulation along neural tracts, thereby providing a coarse-grained understanding of this contribution.

III. APPLICATIONS OF THE METHOD

Analyzing brain data using a network representation is increasingly popular in neuroscience, and researchers have used a wide range of connectomic data to perform NCT analysis. For example, the availability of multimodal neuroimaging data in large cohorts accompanied by clinical and cognitive data [21, 22, 61–63]—as well as indices of neurobiology not measurable *in vivo* (e.g., high-resolution histology [64], gene expression [65], etc.)—enable researchers to validate NCT against brain function and biology as well as examine individual differences. Indeed, we have applied NCT in our research with a view towards achieving these goals. Here, we briefly review selections of this work to show how our protocol may be applied to study the brain. Specifically, we discuss how model outputs from NCT (A) link to network topology; (B) explain individual differences in mental health symptoms, cognition, and age; (C) predict effects of neurostimulation; (D) explain switching between functional task states; and (E) link to neuroanatomy.

A. Understanding the Influence of Topology

In our early work, we began by contextualizing nodal controllability statistics against what we know about connectome topology from graph theory. Specifically, Gu *et al.* [10] examined how nodal control properties–specifically *average controllability* (see Pathway B, Section IX B) and *modal controllability*–correlated with nodes' strength (the sum of a node's edge weights). Gu *et al.* [10] found that nodes' strength correlated strongly positively and negatively with average and modal controllability, respectively. These relations were conserved across both humans and macaques. Collectively, these results indicate that a node's local topological importance predicts its capacity to control the dynamics of a system.

We have also examined how connectome topology influences the *control energy* associated with specific state transitions. Betzel *et al.* [49] found that nodes' topological importance predicted their capacity to facilitate transitions between eight canonical brain states (seven resting-state cortical networks as well as a subcortical network [66]). Specifically, Betzel *et al.* found that target states that intersected with the brain's rich club [58]—a set of highly interconnected nodes that form the connectome's core—exhibited low transition energy. This result demonstrates that the rich club is well positioned in the network to act as an efficient target state to which a diverse set of initial states can transition with low *control energy*. Thus, the topology of the human connectome may be optimized to guide dynamics toward the rich club, bolstering the idea that these nodes support functional integration [57, 67, 68].

Given these advances in understanding how connectome topology contributes to control, we subsequently analyzed what the underlying control equations could tell us about network topology. Starting from the NCT equations, Kim *et al.* [50] derived the features of network architecture that were the most important to determining *control energy*. Kim *et al.* [50] discovered that a strong and diverse set of connections from stimulated nodes to unstimulated nodes were the leading-order contributors to the control cost. Using this discovery, the authors reduced the cost of controlling connectomes in the *Drosophila*, mouse, and human by virtually resecting edges, and developed a method to meaningfully compare the control cost between different species and connectomes. These results provide simple and quantitative knowledge about the most important features of topology according to NCT.

B. Individual Differences

While the strong correlation between *average controllability* and strength reported by Gu *et al.* [10] may seem to imply redundancy between nodal controllability statistics and measures from graph theory incorporating weighted degree, we note that this correlation was spatial (i.e., across the brain) rather than between subjects. In subsequent work examining individual differences, Parkes *et al.* [69] compared the capacity of *average controllability* and strength to predict psychosis spectrum symptoms using out-of-sample testing. Parkes *et al.* [69] found that *average controllability* significantly outperformed strength in this predictive

task, and demonstrated that this improved performance was concentrated in higher-order default mode cortex [70]. These results show that while high *average controllability* may depend upon high strength, there exists unique inter-individual variation between the metrics, and that this variance in *average controllability* couples more tightly to mental health symptoms.

We have also shown that *average controllability* exhibits robust developmental and sex effects. *Average controllability* increases between the ages of 8 and 22 years [12] and is higher in females in the cortex while higher in males in subcortex [19]. Furthermore, Tang *et al.* [12] showed that age effects were strongest in nodes with higher controllability, underscoring the developmental importance of nodes that are well positioned in the network to control dynamics. When examining *control energy*, Cui *et al.* [71] demonstrated that the amount of energy required to activate the fronto-parietal system—a brain network thought to support executive function [72]—was negatively correlated with both age and executive function in the same sample. This result suggests that the developmental emergence of executive function is associated with increased efficiency of neural signaling within the human connectome.

C. Predicting Stimulation Effects

An application of NCT that has clear translational impact is modeling the relationship between brain structure and function. To this end, we have examined whether NCT can predict the brain's functional response to neurostimulation from its structural connectome. For example, in patients with epilepsy, Stiso *et al.* [14] found that NCT was able to predict electrophysiological neuronal responses (measured with electrocorticography) following direct electrical stimulation. This result shows that our model, wherein neural activity is simulated upon the structural connectome, explains variance in experimentally-manipulated empirical changes in brain state.

We have also examined NCT in the context of non-invasive neurostimulation techniques. In a pair of studies, Medaglia *et al.* [16, 17] delivered transcranial magnetic stimulation (TMS) to the left inferior frontal gyrus (LIFG) in between repeated sessions of a set of language tasks. Across both studies, the authors found that NCT metrics extracted from the LIFG explained variance in changes to task performance before and after TMS. These results demonstrate that NCT can be used to probe the network mechanisms that underpin how neurostimulation elicits changes in behavior.

D. Modeling Switches Between Functional Brain States

In addition to predicting the effects of neurostimulation, NCT can be used to investigate how the topology of the structural connectome supports transitions between empirically-observed functional brain states. Our group has studied this process using brain states derived from fMRI. Cornblath *et al.* [20] clustered resting-state fMRI (rs-fMRI) into brain states representing instantaneous co-activations among canonical brain networks, and used NCT to model the energy required to transition between those states. Using a series of null network models (see section X), Cornblath *et al.* found that the topology of the structural connectome was wired to support efficient switching between brain states. This result demonstrates that the topology of the connectome is optimized to support dynamic fluctuations in resting-state activity.

Subsequent work by Braun *et al.* [73] examined transitions between brain states elicited by a working memory task. Braun *et al.* [73] found that transitioning from a 0-back brain state to the more cognitively demanding 2-back brain state required more energy than the reverse transition, demonstrating an asymmetry in *control energy*. Braun *et al.* also found that this energy asymmetry was more pronounced in patients with schizophrenia compared to healthy controls. Thus, while connectome topology may be setup to enable low-cost fluctuations in resting-state [20], activating cognitively demanding brain states may require more control effort. Furthermore, this increased control effort appears to scale with within-task differences in cognitive demand and is further elevated in psychopathology.

E. Biologically informed NCT

Neuroscience is increasingly moving toward a multi-scale approach that seeks to understand how features of the brain observed at one scale link to properties observed at another, and *vice versa* [3, 74–82]. Recently, we have applied this multi-scale approach to NCT by examining how dynamics within the model are influenced by variations to regions' cellular composition. Specifically, we [51] examined how regions' profiles of cytoarchitecture impacted the energy associated with state transitions that spanned the cortical hierarchy (i.e., the sensory-fugal axis [83]). We found that state transitions traversing bottom-up along the cortical hierarchy of cytoarchitecture required lower *control energy* to complete compared to their top-down counterparts, and observed that nodes' position along this hierarchy predicted their importance in facilitating these transitions. This result shows that spatial variations in cortical microstructure constrains macroscopic connectome topology; this effect is consistent with work from neuroanatomy that describes a precise relationship between regions' profiles of cytoarchitecture and their extrinsic connectivity [84].

In recent work from outside our group, Luppi *et al.* [32] characterized the *control energy* associated with a large set of activity maps derived from NeuroSynth [85] related to cognition. Additionally, the authors examined how these transition energies varied when they utilized a broad range of neurotransmitter density maps to modify the control weights. This work ties together switching between functional brain states and biologically informed connectome analysis to provide the field with a comprehensive "look-up table" of how regions' diverse biology impacts *control energy*.

IV. COMPARISON WITH OTHER METHODS

We consider NCT with respect to other models that also seek to understand how communication unfolds within a structurally interconnected complex system. Specifically, for practicing neuroscientists, we view NCT as complementing both more complex biophysical models of dynamics as well as graph-theoretic measures of inter-nodal communication. While both of these approaches model communication, they differ in their biological plausability and complexity.

Biophysical models aim to capture neuronal communication by distilling the various biophysical processes necessary for functional activity into separate model parameters. These parameters are tuned to simulate biologically plausible non-linear dynamics within and between neurons at multiple scales. For example, at the scale of single neurons, the Hodgkin-Huxley model is concerned with modeling neuronal spiking activity [86], and is based on parameterizing the flow of sodium and potassium ions across the cell membrane. At the next scale up, mean-field models focus on the collective activity patterns of co-located populations of neurons [87, 88]. Coupling multiple mean-field models together—where each model represents distinct neuronal populations—enables researchers to study how non-linear dynamics emerge from brain structure at the macroscale [87]. In turn, this approach gives rises to a wide range of complex dynamical behaviors, including synchronized oscillators [89], learning [90, 91], large-scale traveling brain waves [92], and structure-function coupling [93–95]. Broadly, NCT trades biophysical accuracy and the complexity of specific model behaviors for more power in designing and studying stimuli. For example, in lieu of studying state transitions that emerge from different models of associative memory [96] and context integration [97], NCT allows us to design specific stimuli to transition the model system to states that are known to be important for memory and cognitive control under specific constraints [49].

By contrast, graph-theoretic approaches instantiate relatively simple models of inter-nodal communication that rely on assumptions such as shortest-path routing, spatial proximity, random walks, and diffusion processes [4, 9]. While these assumptions are an oversimplification of brain dynamics—and are thus less biologically plausible—their simplicity confers greater analytic tractability and scalability, which are both desirable features when studying the human brain. This benefit compounds when the goal of a given study is to examine inter-individual differences, wherein dynamical models may be fit to thousands of participants. As such, despite their relative simplicity, graph-theoretic approaches have deepened our insights into large-scale brain organization [55, 56, 98–101], improved our understanding of the link between the brain and mental disorders [59, 102–104], and helped elucidate the link between structure and function [40, 44, 45, 105–107].

We consider NCT as situated between these two modeling approaches. As discussed in Section II, NCT is essentially a model of two parts, *dynamics* and *control*. For the former component, NCT models dynamics according to a diffusion process; thus, like graph theory, NCT makes simplifying assumptions of inter-nodal communication, which confers the advantages of analytic tractability and scalability. However, the second component, *control*, adds an additional layer of model parameterization that allows researchers to probe how the system might behave under different contexts (e.g., in response to task manipulations, cognitive control, or neurostimulation protocols). This added flexibility brings NCT closer to biophysical modeling, insofar as they both seek to understand how the dynamics of a system respond to external perturbation. Indeed, we have shown that NCT can be used to predict changes in the dynamics of coupled Wilson-Cowan oscillators following simulated stimulation [18], suggesting that NCT can explain some of the behaviors engendered by non-linear biophysical models.

V. EXPERIMENTAL DESIGN

The goal of an NCT analysis, as it is conceptualized in this protocol, is to understand how the topology of the structural connectome supports and constrains spreading dynamics, and to what extent those dynamics can be controlled. Thus, core to this analysis is the acquisition of one or more structural connectomes from a model organism. These connectomes can be obtained in numerous ways that each depend on the model organism under study and the available imaging modality. In humans, for example, structural connectomes are typically extracted from diffusion-weighted imaging (DWI) sequences obtained using MRI. Tractography algorithms are applied to DWI scans to model the white matter pathways intersecting pairs of brain regions, which are then used to populate connectome edges with the number of those pathways (e.g., the streamline count) [4]. This example constitutes a weighted undirected connectome upon which NCT can be conducted. Note that while the application of NCT is not restricted to weighted undirected connectomes (edges can be weighted or unweighted, directed or undirected), the edge weight definition determines what types of questions can be addressed using this protocol (for example, see Ref. [48] for analysis of an unweighted directed *Caenorhabditis elegans* connectome).

Given that connectomes are central to the application of NCT, any artifacts present in the connectomes will be reflected in model outputs. For example, connectomes populated by DWI estimates of connectivity are known to contain false positives and false negatives, which may be partly mitigated by the use of thresholding techniques [108, 109]. In-scanner head motion is well known to spuriously impact these estimates of connectivity as well [110, 111]. Thus, the accurate generation and rigorous quality control of connectomes are both crucial considerations for experimental design. For human connectomes, we recommend researchers consult the extant literature on the processing and quality control of DWI scans [108, 112, 113] (see also https://qsiprep.readthedocs.io/).

Another consideration for connectome estimation is the brain parcellation used to define system nodes. If, as mentioned above, structural connectivity is determined by streamline count, then variations in the size of regions across the parcellation will bias connectome edge weights; larger brain regions will intersect with more white matter pathways and thus show higher connectivity to the remaining regions. As with any analysis of graph topology, this bias will effect the outputs of NCT; for example, larger regions may show higher *average controllability* just by virtue of being more directly connected to the system. It is for this reason that we recommend researchers reproduce their results using several different parcellation definitions and resolutions. Doing so ensures that their results are not driven by a specific parcellation choice.

Beyond the core requirement of a connectome, the flexibility of NCT makes it applicable to a broad range of experimental designs (see Section III); the most critical component is that researchers have hypotheses that pertain to studying the control of brain dynamics. However, in the case of *control energy*, where researchers will study the control signals, $\boldsymbol{u}(t)$, there are some additional considerations. First, differences in brain states' magnitude will impact *control energy*, potentially necessitating the normalization of state magnitude. For example, if researchers are examining transitions between patterns of brain activity (e.g., using functional data as in [20, 73]), then differences between states' mean activity will impact *control energy*; assuming a common initial state, target states with higher activity will require more energy to transition to compared to target states with lower activity. This effect generalizes to binary brain states—where a brain state is encoded as a set of nodes being "on" while the rest of the nodes are "off"—as well. In this case, differences in state size (i.e., the number of "on" regions) constitute differences in state magnitude; transitioning to larger target states will require more energy. If there are differences in state magnitude, we recommend normalizing states before computing *control energy* (see Section IX A, step 3). Note, the need for this normalization will depend upon researchers' analyses. For example, if researchers are studying individual differences in the energy associated with a single transition, then normalization may not be necessary so long as state definition is consistent across subjects. What is critical is that researchers consider what comparisons they want to make and whether variations in state definition would confound those comparisons.

A second consideration is how researchers define their control set, B. As discussed in Section II, the $N \times m$ control set defines the extent to which the nodes of the system can affect changes in its dynamics. In turn, the definition of B determines the dimensions of $\boldsymbol{u}(t)$; the greater the number of control nodes, the more independent control signals will be generated. In our work, we have often deployed a *uniform full control set*, which means that all of the nodes of the system are designated as controllers (*full*) and all are given equivalent control over dynamics (*uniform*). In this case, $m = N$. Intuitively, this approach assumes that the entire brain is being controlled—either internally or externally—when completing a state transition. However, depending on a researcher's hypotheses, this assumption may not be appropriate. Instead, researchers may want to define only a subset of nodes as controllers (e.g., [49, 50]), or assign variable weights to control nodes (e.g., [30–32, 51]), or both. Note that assigning variable control weights serves to give some nodes more control over system dynamics than others. In any case, it is critical that researchers check whether their designated control set was able to complete the associated state transition (see Section IX A, step 5); successful completion of a state transition is not guaranteed in the model, and completion is less likely when transitions are driven by a small control set.

VI. EXPERTISE NEEDED TO IMPLEMENT THE PROTOCOL

We provide open-source and broadly accessible tools that implement *optimal control* and *average controllability* in a Python-based software package called *network control theory for python (nctpy)*. In *nctpy*, we provide a flexible implementation that enables researchers to make model assumptions that best fit their research question. As a result, while a full understanding of linear systems and optimal control theory are not required, the researcher must have enough expertise to make key modeling decisions that best represent the data, which we explicitly mark in the protocol.

The first piece of expertise needed is to understand the differences between (and implications of) discrete-time systems and continuous-time systems (see Section II). This difference is not merely a conceptual one, because discrete-time systems display a fundamentally different set of behaviors than continuous-time systems. That is, a discrete-time system is not simply a temporally coarse-grained version of a continuous-time system. Instead, each system exhibits different dynamics. As a simple example, consider a 1-dimensional, discrete-time system that evolves according to $x(t+1) = -x(t)$. Starting at 1, this system will alternate between -1 and 1 over time. There does not exist an equivalent continuous-time system that evolves according to $\frac{\mathrm{d}}{\mathrm{d}t}x(t) = ax(t)$ (where a is a real number) that oscillates in this way.

The second piece of expertise needed is to understand the nature of Pathway A (*control energy*, Section IX A) and Pathway

B (*average controllability*, Section IX B) in order to interpret the outputs. Pathway A is solving an optimization problem. Specifically, first we provide a model of the dynamics (i.e. A, B), the initial and target states, and some optimization parameters (see Section IX A). Then, we solve for the control signals, $\boldsymbol{u}(t)$, that minimize the cost. Hence, all interpretations of $\boldsymbol{u}(t)$ should be made with the understanding that they were determined by the user-defined optimization parameters. Pathway B is not solving an optimization problem, and thus does not receive any optimization parameters. Rather, it measures the magnitude of the neural states over time as a result of an impulse stimulation. Because Pathways A and B use the same dynamics but output different quantities through different means, more expertise in linear systems and optimal control is needed to meaningfully compare and contrast the two pathways.

VII. LIMITATIONS AND ONGOING DEVELOPMENT

Network control theory has the ability to flexibly accommodate many scientific questions and to generate concise knowledge from a simple model. However, NCT also possesses several limitations that are faced by many in the study of high-dimensional complex systems, such as numerical stability of algorithms, validation at the scale of microscopic states, approximations of complex interactions, and interpretation of model parameters.

A. Numerical Stability of Optimization

One limitation is the numerical stability of Pathway A under certain parameter conditions, which arises from ill-conditioned matrices that are built while solving for the control signals. This issue occurs most frequently when using a relatively small control set—a small m in the $N \times m$ matrix B—to control a network with large N. It is intuitive that precisely controlling the initial and target states of the whole brain from only a few nodes is difficult. In light of this limitation, it is crucial that the researcher carefully study the generated trajectories of the neural activity to ensure that the desired initial and target states are reached, and that the numerical integrator does not generate a warning of numerically ill-conditioned matrices. In the event that the control set must be small for the purposes of the research question, one solution may be to extend the control set by heavily weighting the desired control nodes and lightly weighting the remaining nodes [14]. Another option is to use Pathway B to study the *average controllability* of the control set.

B. Validation at the Scale of Microscopic States

A second limitation is the validation of the model at the level of individual neural states. Phrased another way: given a connectome, A, and stimulations $\boldsymbol{u}(t)$ delivered to brain regions B starting at neural activations $\boldsymbol{x}(0)$, does experimentally measured neural activity agree with the simulated trajectory $\boldsymbol{x}(t)$? The challenges associated with addressing this sort of question extend far beyond NCT and to a significant portion of systems and network neuroscience. Microstate validation between neural structure and activity is most evident in small systems of neural circuits [114], but how to perform similar validations for large-scale systems such as the human brain remains an open area of research. Challenges include (i) the multiple possible scales of constructing brain networks [2, 115]; (ii) differing measures of inter-areal connectivity [108, 116]; (iii) multiple definitions of simulated neural activity [87, 117–119]; and (iv) the diverse spatial and temporal resolutions at which we can record whole-brain activity [52]. Along this active area of research, recent work has demonstrated that linear models outperform non-linear and kernel-based models in both 1-step prediction and model complexity for both fMRI and EEG data [120], as well as correspondence between *control energy* and local metabolism [121].

Closely linked to ideas of validation is ongoing work on the implementation of the connectome, A. NCT stipulates a simple but mechanistic model for the evolution of neural states, $\boldsymbol{x}(t)$, given a stimulus $\boldsymbol{u}(t)$. Hence, the interpretation of the model outputs is bounded by the interpretability of the model inputs. In the majority of applications, the matrix A is taken to be the structural connectome, with the justification that regions must be able to transmit information along structural connections. However, the distribution of edge weights can vary significantly across different pre-processing pipelines, which implies equally significant variations in the strength of interactions between regions. Hence, it is important for the interpretation of the results to critically assess whether the edge from node i to node j of the implemented adjacency matrix is a reasonable measure of the strength of activity diffusion from node i to node j.

C. Linear Dynamics

A third limitation is the assumption of linear dynamics, which enables the calculation of powerful measures such as optimal control trajectories, but hinders the biophysical realism of the framework. More sophisticated non-linear models capture complex

dynamics from individual neurons [86] to neural populations [37], thereby enabling the study of fine-grained experimental behavior [122] and complex non-linear phenomena [123]. These models make fewer simplifying assumptions to capture non-linear behaviors of biological systems such as complex memory landscapes [124]. While prior work has shown that linear models outperform many classes of non-linear models in describing and predicting brain-wide neural activity [120], extensions of NCT to non-linear systems will enable greater flexibility to accommodate and explore the impact of non-linear biophysical constraints. While the theory of non-linear control is an active area of research [125], there are immediate applications of NCT to non-linear systems, and many exciting potential extensions of NCT to capture more biophysical realism.

Broadly speaking, the linear dynamics assumed by NCT can be thought of as being valid for a non-linear system within small deviations of an operating state [7]. Hence, the most immediate application of NCT to non-linear systems is to linearize the model about an operating point, such as the upright position of an inverted pendulum [126]. Along these lines, the next immediate generalization to NCT is to linear time-varying systems [47], where the model is linearized not about a point, but about a trajectory. While methods to implement control for linearized and time-varying systems are well-established in the control community, a biophysically meaningful implementation and interpretation of the parameters—namely $A(t)$ and $B(t)$—remains an area of active work [127]. Another approximation that is particularly relevant for high-dimensional neural systems is at the limit of weakly coupled oscillators [128, 129], whereby a high-dimensional system of oscillators with weak interactions can be reduced to a low-dimensional phase response curve, allowing for the potential linearization of the system about phase-locked states.

In addition to linearizing dynamics about points and trajectories, NCT can also meaningfully be applied to non-linear dynamical systems that can be made linear through a non-linear change of variables. One such example is by using finite-dimensional *Koopman subspaces*, which allow for the recasting of non-linear systems with single fixed points as higher-dimensional linear systems [130], and closely-related methods in dynamic mode decomposition [131]. Further, advances in non-linear control enable us to probe important coarse-grained questions such as the control set necessary to push non-linear systems between attracting states [132]. Other control strategies take advantage of the ability of non-linear systems to access states that lie outside of their linearization [133].

VIII. MATERIALS

As discussed above, we split our protocol into two pathways (Figure 3). The primary pathway of our protocol focuses on computing *control energy* (Pathway A, Section IX A). This pathway is illustrated in Figures 3A, 3B, and 3C. Briefly, Figure 3A outlines the inputs required to run our protocol, Figure 3B outlines the model outputs, and Figure 3C outlines some of the variations to model inputs that we have discussed thus far. The second pathway of our protocol focuses on computing nodes' *average controllability* (Pathway B, Section IX B; Figure 3D).

Equipment

- A computer with *Python* (tested on version 3.9) and `nctpy` installed alongside its dependencies. This protocol has been tested on Mac OS running on Intel Core i5/i7/i9 processors as well as on Apple Silicon. We have also tested this protocol on Linux Ubuntu running on Intel processors. RAM requirements will vary depending on researchers' data and analyses, but we recommend at least 16 GB. Finally, we recommend installing `nctpy` inside a virtual environment managed by Anaconda (`https://www.anaconda.com/`). The following core dependencies are required to run `nctpy`:
  - numpy (`https://numpy.org/`). Tested on version 1.24.3.
  - scipy (`https://scipy.org/`). Tested on version 1.10.1.
  - tqdm (`https://github.com/tqdm/tqdm`). Tested on version 4.65.

  Additionally, there are some functions in `nctpy.plotting` and `nctpy.utils` that require the following:
  - statsmodels (`https://www.statsmodels.org/`). Tested on version 0.13.5.
  - matplotlib (`https://matplotlib.org/`). Tested on version 3.7.1.
  - seaborn (`https://seaborn.pydata.org/`). Tested on version 0.12.2.
  - nibabel (`https://nipy.org/nibabel/`). Tested on version 5.1.
  - nilearn (`https://nilearn.github.io/`). Tested on version 0.10.1.

  Finally, the following optional packages were used to run the analyses illustrated in this protocols paper:
  - (optional) pandas (`https://pandas.pydata.org/`). Tested on version 1.5.3.
  - (optional) scikit-learn (`https://scikit-learn.org/`). Tested on version 1.2.2.

See https://github.com/BassettLab/nctpy for more details. Creating a *Python* environment using Anaconda and installing the above dependencies should take no longer than 30 minutes.

Input Data

- CRITICAL **Adjacency matrix, A (used in all steps)**: The adjacency matrix, A (Figure 3A, left), is the primary input to our protocol. In A, the N nodes of the system are stored on the rows and columns, and the $N \times N$ edge values are stored in the entries. As discussed above, both the nodes and the edges of A can be defined in numerous ways and their definition depends on the acquired data as well as the research question. For example, the nodes of the system may be defined as single neurons in organisms such as the *Caenorhabditis elegans* [48, 134] or as brain regions of varying size and definition in organisms such as the mouse [76], Drosophila [50], macaque [135] and human [4]. The edges of A may be defined as either the directed or undirected structural connectivity between nodes (Figure 3C). Note, effective functional connectivity between nodes may also be used to define edges [15, 136, 137]. Critically, our model assumes that A_{ij} encodes the strength of diffusion of activity along the edge connecting *node j to node i*. In other words, our model assumes that the columns of A store the source nodes (i.e., projections from node j) while the rows store the target nodes (i.e., projections to node i). While this distinction is irrelevant for undirected connectomes where $A_{ij} = A_{ji}$, it is crucial for directed connectomes. Thus, researchers must ensure that their directed A matrix conforms to the above assumptions. In any case, NCT can be applied to either participant-level or group-averaged connectomes. Although we focus on a group-averaged undirected structural connectome in this protocol, we also include an example of our protocol applied to a directed structural connectome.

- **Brain states, $x0$ and xf (used in steps 3-6)**: In order to analyze state transitions, researchers also need to provide a pair of brain states (Figure 3A, middle; initial state, $x0$; target state, xf) relevant to their hypotheses. Providing these states allows NCT to find the control signals, $\boldsymbol{u}(t)$, that are required to transition between them and to summarize those control signals as *control energy* (Figure 3B); here, *control energy* estimates the amount of effort the model has to exert to complete the transition. Brain states can be defined in a number of ways. The simplest approach is to define each brain state as a binary vector (Figure 3A, middle), where nodes that are within a given state are assigned an arbitrary constant value (e.g., 1) and any remaining nodes are assigned a value of 0. In this setup, NCT is tasked with transitioning the brain between actuating different sets of node to a constant arbitrary level of neural activity. We commonly adopt this approach in our work. An alternative approach is to allow brain states to represent a variable pattern of activity (Figure 3C, middle). As mentioned above, Cornblath *et al.* [20] modeled the energy required to transition between brain states derived from clustering of fMRI data, while Braun *et al.* [73] used task activation maps extracted from an fMRI contrast. These approaches allowed the authors to generate state vectors that encode non-zero activity across all nodes of the system. In this protocol, we illustrate examples using both binary and non-binary brain states. Note that brain states are not required for Pathway B (Figure 3D).

- **Control set, B (used in steps 3-6)**: In addition to brain states, researchers also need to designate a control set; these are the nodes that NCT will use to complete state transitions. Intuitively, the control nodes are where time-varying control signals, $\boldsymbol{u}(t)$, will be injected into the system in order to transition the system between states. Each column k of B indicates the impact of input u_k on the network nodes. Here, the simplest approach is to define a *uniform full control set* (Figure 3A, right), which means that all nodes of the system are designated as control nodes and that they are all given the same degree of control over system dynamics. Alternatively, as discussed above, researchers may also designate partial control sets or control sets with variable weights (Figure 3C, right). For the latter, variable weights can either be derived *a priori* and input directly into the model (e.g., [30–32]) or via data-driven optimization approaches (e.g., [51]). We illustrate examples of using such control sets in this protocol. Note that a control set is not required for Pathway B (Figure 3D).

Example Dataset

- We primarily used undirected structural connectomes derived from DWI performed on the human brain. We obtained these connectomes from the Philadelphia Neurodevelopmental Cohort (PNC) [21, 22], a community-based study of brain development in youths aged 8 to 22 years. The neuroimaging sample of the PNC consists of 1,601 participants. From this original sample, we retained 253 typically developing participants who had no medical co-morbidity or radiological abnormalities, and who were not taking psychoactive medications at the time of assessment. Additionally, these participants' T1-weighted, DWI, and rs-fMRI scans all passed stringent quality control procedures [113, 138, 139].

  - Structural connectome reconstruction was performed using QSIprep 0.14.2 [112], which is based on Nipype 1.6.1 [140]. Connectomes were extracted using the 200-node variant of the Schaefer parcellation [115], ordered according to 7

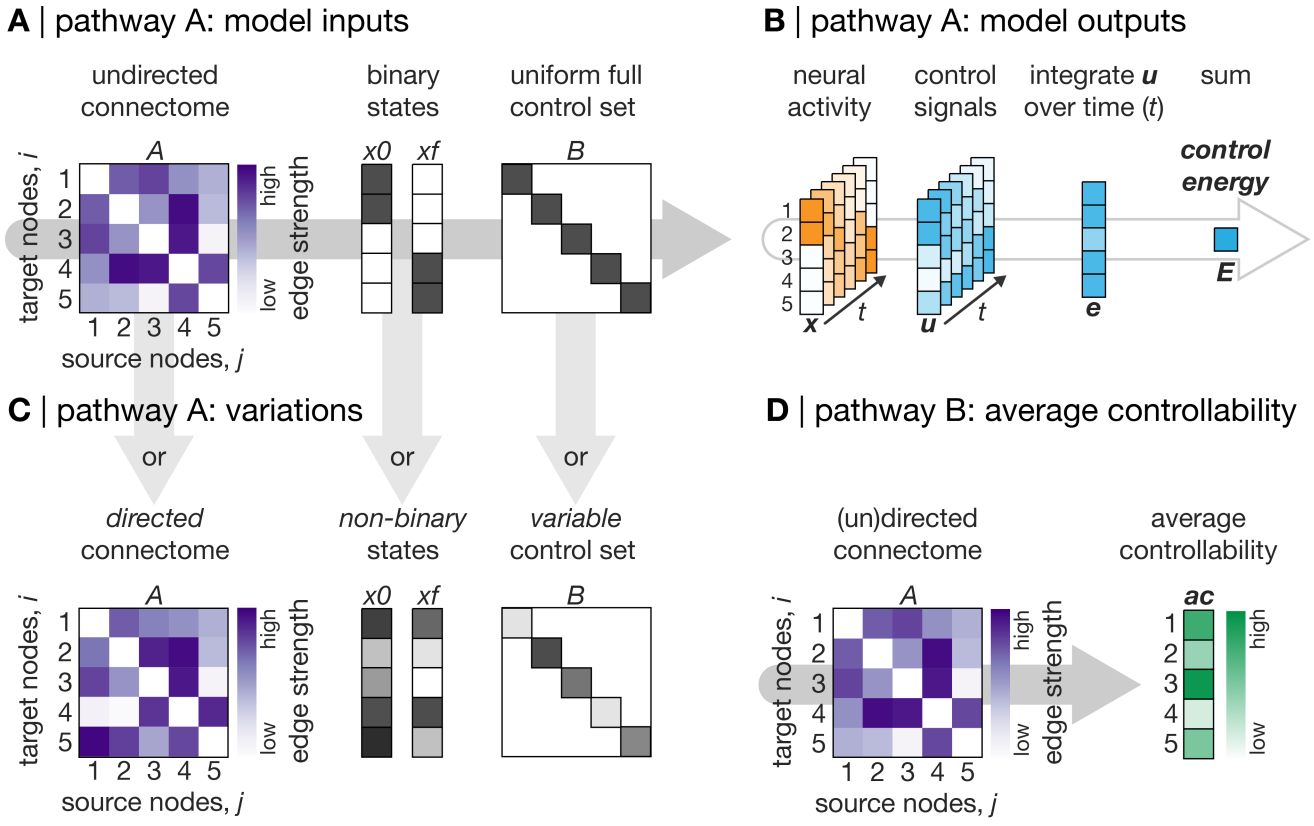

FIG. 3
Schematic representation of the network control theory (NCT) protocol. Our protocol is split into two pathways. Primarily, our protocol focuses on modeling the *control energy* associated with user-defined control tasks. We refer to this part of our protocol as Pathway A (**A**, **B**, and **C**). Pathway A will be of interest to researchers who seek to study specific state transitions. We also outline a brief protocol for estimating nodes' *average controllability*. We refer to this part of our protocol as Pathway B (**D**). Pathway B will be of interest to researchers who want to examine nodes' general capacity to control system dynamics. **A**, Inputs required for Pathway A. To compute *control energy*, researchers must provide a structural connectome (A), an initial state ($x0$), and a target state (xf), and must also define a control set (B). **B**, Model outputs from Pathway A. Given these inputs, our protocol will output the state trajectory (neural activity, $\boldsymbol{x}(t)$) and the control signals ($\boldsymbol{u}(t)$). Once inspected, the control signals can be integrated over time to obtain node-level energy ($\boldsymbol{e}$), which in turn can be summed over nodes to get the *control energy* ($\boldsymbol{E}$). **C**, Variations to Pathway A. Pathway A can handle a diverse range of inputs, including but not limited to undirected and directed connectomes (left), binary and non-binary brain states (middle), and control sets with uniform or variable weights (right). **D**, Pathway B: *average controllability*. Pathway B only requires a structural connectome (A) as input and will return the *average controllability* of each node. This metric quantifies the impulse response of the system from a given node. Higher *average controllability* indicates that a node is better positioned in the network to propagate dynamics.

canonical brain systems [66]. The strength of inter-regional connectivity was summarized using the number of streamlines that intersected each pair of parcels. Connectomes were averaged over subjects. This group-averaged connectome was thresholded by retaining the edges that were present in at least 60% of participants' connectomes [109]. This process resulted in a final connectome with 98% edge density.

- rs-fMRI was also obtained from the same 253 PNC participants [21]. These data were used to generate empirical brain activity states (see Figure 6). The eXtensible Connectivity Pipeline (XCP-D) [139, 141] was used to post-process the outputs of fMRIPrep version 20.2.3 [142]. XCP was built with Nipype 1.7.0 [140]. Processed rs-fMRI time series were extracted from the same 200-node parcellation mentioned above [115].

- We also studied a directed structural connectome obtained from the Allen Mouse Brain Connectivity Atlas.

- Whole-brain structural connectomes were constructed with 2×10^5 voxels at a spatial resolution of 100 µm. Voxels were assigned to regions (coarse structures) according to a 3-D Allen Mouse Brain Reference Atlas [23]. Isocortex was further divided into 6 systems (auditory, lateral, medial, prefrontal, somatomotor, and visual) based on prior work that applied community detection to identify stable modules [143]. Connection strengths were modeled for all source

and target voxels using data from 428 anterograde tracing experiments in wild type C57BL/6J mice [144]. Normalized connection strengths were obtained by dividing the connection strengths by the source and target region sizes. Here, we retained only the 43 isocortical regions. This process resulted in a fully-connected directed structural connectome.

In all of the below code, we assume the existence of a *Python* environment with `nctpy` installed alongside its dependencies. First, we import all the functions we need to run our protocol:

```
# import
import os
import numpy as np
import pandas as pd
import scipy as sp
from scipy import stats
from scipy.spatial import distance
from sklearn.cluster import KMeans
from tqdm import tqdm

# import plotting libraries
import matplotlib.pyplot as plt
import seaborn as sns
from nilearn import datasets
from nilearn import plotting

# import nctpy functions
from nctpy.energies import integrate_u, get_control_inputs
from nctpy.pipelines import ComputeControlEnergy, ComputeOptimizedControlEnergy
from nctpy.metrics import ave_control
from nctpy.utils import (
    matrix_normalization,
    convert_states_str2int,
    normalize_state,
    normalize_weights,
    get_null_p,
    get_fdr_p,
)
from nctpy.plotting import roi_to_vtx, null_plot, surface_plot, add_module_lines
from null_models.geomsurr import geomsurr
```

Note that depending on their goals, researchers may only need a subset of this import call. Next, we will load a structural connectome as our adjacency matrix:

```
# directory where data is stored
datadir = '/path/to/data'
adjacency_file = 'structural_connectome.npy'

# load adjacency matrix
adjacency = np.load(os.path.join(datadir, adjacency_file))
n_nodes = adjacency.shape[0]
print(adjacency.shape)

# check for self-connections
print(np.any(np.diag(adjacency) > 0))

# get density including self connections
density = np.count_nonzero(np.triu(adjacency, k=0)) / (n_nodes**2 / 2)
print(density)
```

```
(200, 200)
True
0.9768
```

The above code demonstrates that our connectome comprises 200 nodes, includes self-connections (i.e., $A_{ij} > 0$), and has an edge density of 98%. See Figure S1 for *control energy* plotted as a function of edge density.

IX. PROCEDURE

Core Steps

Discrete-time versus continuous-time dynamical system

1. CRITICAL **Define a time system**. The first step is to determine whether to model the linear dynamical system in discrete- or continuous-time. In a discrete-time system, the states of the system, $\boldsymbol{x}(t)$, evolve forward in time according to a set of discrete steps ($\boldsymbol{x}(t) \rightarrow \boldsymbol{x}(t+1)$). In a continuous-time system, the states of the system are continuously changing in time ($\boldsymbol{x}(t) \rightarrow \boldsymbol{x}(t) + \dot{\boldsymbol{x}}(t)\mathrm{d}t$). The choice of time system will depend upon the research question and affects all subsequent analyses owing to differences in the mathematical implementation of NCT under each system. We refer the reader to Karrer *et al.* [8], Kim *et al.* [7], Hespanha [47], and other texts in linear systems theory for extended discussion.

2. CRITICAL **Normalize adjacency matrix** (Timing: <1 second). Once a time system has been determined, the first practical step in this protocol is to normalize the adjacency matrix, A. Normalizing A prior to modeling the dynamical system ensures stability and that the system will not grow to infinity over time. We include a function that normalizes A appropriately depending on the researcher's chosen time system.

   A) Normalizing for discrete-time systems:

```
# normalize adjacency matrix
system = "discrete"
adjacency_norm = matrix_normalization(adjacency, system, c=1)
```

   The above call will normalize A according to the following equation:

$$A_{norm} = \frac{A}{|\lambda(A)|_{max} + c}.$$

   Here, $|\lambda(A)|_{max}$ denotes the largest absolute eigenvalue of the system. Additionally, c is a user-defined input parameter that determines the rate of decay of system dynamics. We set $c = 1$ by default, which ensures that all modes of the system decay and thus that activity goes to zero over time (note, this is true of any positive c value). This normalization ensures that internal dynamics decay in a manner that is necessary for the stabilization of the system. Specifically, the largest absolute value of a matrix's eigenvalues is called the *spectral radius*, and this normalization ensures that the spectral radius is less than 1: a condition known as *Schur stability*. Intuitively, a discrete-time system given by Eq. 1 with no input (i.e. $\boldsymbol{u}(t) = 0$) will evolve as $\boldsymbol{x}(n) = A^n \boldsymbol{x}(0)$, and the most unstable eigenmode of the system will evolve as $\lambda(A)^n_{max}$. To ensure that this mode does not grow infinitely with n, it must have a magnitude less than 1.

   B) Normalizing for continuous-time systems:

```
# normalize adjacency matrix
system = "continuous"
adjacency_norm = matrix_normalization(adjacency, system, c=1)
```

   The above call will normalize A according to the following equation:

$$A_{norm} = \frac{A}{|\lambda(A)|_{max} + c} - I.$$

   Here, I denotes the identity matrix of size $N \times N$. As above, we normalize such that the spectral radius is less than one, but we take the additional step of subtracting the identity. This step exists because a continuous-time system given by Eq. 2 with no input will evolve as $\boldsymbol{x}(t) = e^{At}\boldsymbol{x}(0)$, and the eigenmodes of the system as $e^{\lambda_i t}$. Hence, for the system to decay, all λ_i must have a negative real component, which is achieved through the subtraction of I.

Irrespective of the chosen time system, the above step outputs A_{norm}, which contains the structural connectome as a normalized adjacency matrix that is ready for NCT analysis. Importantly, there are several different approaches to normalizing the matrix. Irrespective of approach, the key properties to ensure are the stability of the system as well as the preservation of as much structural information as possible. In all of the code and results shown below, A_{norm} was produced for a continuous-time system.

A. Protocol Pathway A: Control Energy

3. **Define a control task: binary brain states** (Timing: <1 second). To calculate the *control energy* required to complete a state transition, researchers must first define a control task. A control task comprises an initial state, $x0$, a target state, xf,

and a control set, B. In other words, a control task involves defining a specific set of control nodes that will be used by the model to transition from an initial state to a target state.

Here, we illustrate an example control task where NCT is used to transition between a pair of *binary brain states* controlled by a *uniform full control set*. As mentioned above, our A_{norm} is ordered according to 7 canonical brain systems [115]. We leverage this grouping to define a state transition between the visual system and the default mode network (DMN). To begin, we set up a vector, `states`, that stores integer values denoting which brain system each node belongs to. That is, `states == 0` represents nodes that belong to system 1, `states == 1` represents nodes that belong to system 2, *etcetera*. We create `states` from a list of strings that groups nodes into the aforementioned canonical brain systems (this file can be found here):

```
# load node-to-system mapping
system_labels = list(
    np.loadtxt(os.path.join(datadir, "pnc_schaefer200_system_labels.txt"), dtype=str)
)

print(len(system_labels))
print(system_labels[:20])
```

```
200
['Vis', 'Vis', 'Vis', 'Vis', 'Vis', 'Vis', 'Vis', 'Vis', 'Vis', 'Vis',
'Vis', 'Vis', 'Vis', 'Vis', 'SomMot', 'SomMot', 'SomMot', 'SomMot', 'SomMot', 'SomMot']
```

In `nctpy`, we include a function called `convert_states_str2int` that will convert this list of strings for us:

```
# use list of system names to create states
states, state_labels = convert_states_str2int(system_labels)

print(type(state_labels), len(state_labels), state_labels)
print(type(states), states.shape, states)
```

```
<class 'list'> 7 ['Cont', 'Default', 'DorsAttn', 'Limbic', 'SalVentAttn', 'SomMot', 'Vis']
<class 'numpy.ndarray'> (200,) [6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 5 5 5 5 5 5 5 5 5 5 5 5 5 5
5 5 2 2 2 2 2 2 2 2 2 2 2 2 2 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 5
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 4 4 4 4 4 4 4 4 4 4 3 3
3 3 3 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
```

As can be seen from the above `print` commands, our `system_labels` variable comprises a string variable for every node in our system that denotes which brain system that node belongs to. `convert_states_str2int` takes that list of strings and returns an array of integers, `states`, with a corresponding list of labels, `state_labels`. Below, we extract $x0$ and xf using the integers that correspond to the visual system (`'Vis'`) and the default mode system (`'Default'`):

```
# extract initial state
initial_state = states == state_labels.index('Vis')

# extract target state
target_state = states == state_labels.index('Default')
```

`initial_state` and `target_state` will be Boolean vectors (`[True, False]`), wherein `True` encodes the nodes that belong to a given state. Next, we normalize the state magnitude using our included function, `normalize_state`:

```
# normalize state magnitude
initial_state = normalize_state(initial_state)
target_state = normalize_state(target_state)
```

This process will convert `initial_state` and `target_state` from Boolean entries to floating point numbers that have been normalized using the Euclidean norm of the vector. This normalization constrains state magnitude to a unit sphere (see Section V for more details). Finally, we define our control set. Unlike the initial and target states, the control set is encoded along the diagonal of an $N \times N$ matrix, B. Here, we use a *uniform full control set*; thus, we can define our control set as the identity matrix:

```
# specify a uniform full control set: all nodes are control nodes
# and all control nodes are assigned equal control weight
control_set = np.eye(n_nodes)
```

4. **Compute control signals and state trajectory** (Timing: <1 second). Following the definition of a control task, the next step is to find the control signals, $\boldsymbol{u}(t)$, that drive the system to transition between $x0$ and xf. $\boldsymbol{u}(t)$ will be an $m \times T$ matrix of m time-varying signals injected into the control nodes over a specified time horizon, T. Here, owing to our use of a full control set, $m = N$. Critically, injecting these control signals into a system whose initial state is encoded by $x0$ should result in a system whose final state is encoded by xf at time T. Alongside the control signals, we also extract the state trajectory, $\boldsymbol{x}(t)$. The state trajectory, which will be an $N \times T$ matrix, is the time-varying pattern of simulated neural activity that unfolds as the system traverses between $x0$ and xf. In this protocol, we find $\boldsymbol{u}(t)$ and $\boldsymbol{x}(t)$ using the `get_control_inputs` function:

```
# set parameters
time_horizon = 1  # time horizon (T)
rho = 1  # mixing parameter for state trajectory constraint
trajectory_constraints = np.eye(n_nodes)  # nodes in state trajectory to be constrained

# get the state trajectory, x(t), and the control signals, u(t)
state_trajectory, control_signals, numerical_error = get_control_inputs(
    A_norm=adjacency_norm,
    T=time_horizon,
    B=control_set,
    x0=initial_state,
    xf=target_state,
    system=system,
    rho=rho,
    S=trajectory_constraints,
)
```

By default, we set `time_horizon=1`. Note that this value is arbitrary and does not correspond to any real-world time units (e.g., seconds). Importantly, `get_control_inputs` utilizes a cost function that includes both the magnitude of the control signals and the magnitude of the state trajectory. The input parameter `rho` allows researchers to tune the mixture of these two costs while finding the input $\boldsymbol{u}(t)$ that achieves the state transition. Specifically, `rho=1` places equal cost over the magnitude of the control signals and the state trajectory. Reducing `rho` below 1 increases the extent to which the state trajectory adds to the cost function alongside the control signals. Conversely, increasing `rho` beyond 1 reduces the state trajectory contribution, thus increasing the relative prioritization of the control signals. Lastly, `S` takes in an $N \times N$ matrix whose diagonal elements define which nodes' activity will be constrained in the state trajectory. In summary, `S` designates which nodes' neural activity will be constrained while `rho` determines by how much it will be constrained, relative to the control signals. Here, by setting `rho=1` and `S=np.eye(n_nodes)`, we are implementing what we refer to as *optimal control* [11]. Alternatively, researchers may choose to constrain only a subset of the state trajectory by defining partial constraint sets. If `S` is set to an $N \times N$ matrix of zeros, then the state trajectory is completely unconstrained; we refer to this setup as *minimum control* [20, 51]. In this case, `rho` is ignored. See here for a notebook outlining different use cases of `get_control_inputs`.

In addition to `state_trajectory` and `control_signals`, `get_control_inputs` also outputs `numerical_error`, which stores two forms of numerical error. The first error is the *inversion* error, which measures the conditioning of the optimization problem. If this error is small, then solving for the control signals was well-conditioned (see Section VII). The second error is the *reconstruction* error, which is a measure of the distance between xf and $\boldsymbol{x}(T)$. If this error is small, then the state transition was successfully completed; that is, the neural activity at the end of the simulation was equivalent to the neural activity encoded by xf. We consider errors $< 1^{-8}$ as adequately small:

```
# print errors
thr = 1e-8

# the first numerical error corresponds to the inversion error
print(
    "inversion error = {:.2E} (<{:.2E}={:})".format(
        numerical_error[0], thr, numerical_error[0] < thr
    )
)

# the second numerical error corresponds to the reconstruction error
print(
    "reconstruction error = {:.2E} (<{:.2E}={:})".format(
        numerical_error[1], thr, numerical_error[1] < thr
    )
)
```

```
inversion error = 1.36E-15 (<1.00E-08=True)
reconstruction error = 5.16E-14 (<1.00E-08=True)
```

5. **Visualize state trajectory and control signals** (Timing: $<$ 1 second). Once $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ have been derived, they should be visualized before computing *control energy*. Visualization provides intuition regarding how the model is behaving and is helpful for confirming that the state transition was completed successfully. As noted in Section V, completion of a state transition is not guaranteed by the model, and an incomplete transition may necessitate revising either the control set (e.g., to provide more control over the system if a partial control set was used) or the time horizon (e.g., to provide more time for the model to complete the transition). We suggest the following simple plot:

```
f, ax = plt.subplots(3, 2, figsize=(7, 7))
# plot control signals for initial state
ax[0, 0].plot(control_signals[:, initial_state != 0], linewidth=0.75)
ax[0, 0].set_title("A | control signals, x0")
# plot state trajectory for initial state
ax[0, 1].plot(state_trajectory[:, initial_state != 0], linewidth=0.75)
ax[0, 1].set_title("B | neural activity, x0")

# plot control signals for target state
ax[1, 0].plot(control_signals[:, target_state != 0], linewidth=0.75)
ax[1, 0].set_title("C | control signals, xf")
# plot state trajectory for target state
ax[1, 1].plot(state_trajectory[:, target_state != 0], linewidth=0.75)
ax[1, 1].set_title("D | neural activity, xf")

# plot control signals for bystanders
ax[2, 0].plot(
    control_signals[:, np.logical_and(initial_state == 0, target_state == 0)],
    linewidth=0.75,
)
ax[2, 0].set_title("E | control signals, bystanders")
# plot state trajectory for bystanders
ax[2, 1].plot(
    state_trajectory[:, np.logical_and(initial_state == 0, target_state == 0)],
    linewidth=0.75,
)
ax[2, 1].set_title("F | neural activity, bystanders")

for cax in ax.reshape(-1):
    cax.set_ylabel("activity")
    cax.set_xlabel("time (a.u.)")
    cax.set_xticks([0, state_trajectory.shape[0]])
    cax.set_xticklabels([0, time_horizon])

f.tight_layout()
plt.show()
```

This plot (Figure 4) shows the control signals (left column) alongside the state trajectory (i.e., neural activity; right column) separately for nodes within the initial (top row) and the target state (middle row), as well as the bystanders (bottom row). Note, we define bystanders as nodes that are outside both the initial and target states. We choose this division of nodes as it provides several simple intuitions about model behavior. First, we can see that the model is driving negative time-varying control signals into the nodes of the initial state (Figure 4A), which drives their activity to 0 over time (Figure 4B). Second, we can see that the model is driving positive time-varying control signals into the nodes of the target state (Figure 4C), which drives their activity from 0 to $\sim$0.15 over time (Figure 4D). Note that $\sim$0.15 represents the maximum neural activity following state normalization for the states presented here; this maximum activity may vary depending on state definition. Finally, we can see that diverse time-varying control signals are being injected into the bystander nodes (Figure 4E), which are in turn guiding changes in these regions' activity (Figure 4F). In other words, Figure 4 shows that the model is performing a combination of "turning off" the initial state, "turning on" the target state, as well as guiding diffusing activity toward the target state via the bystanders. Figure 4 also provides a simple visual way to check whether the state transition completed successfully; at the end of the simulation, it is apparent that activity in the target state is maximal while activity in the initial state and bystanders is 0, which accords with our definition of xf. This behavior explains the low reconstruction error shown above. Additionally, this plot allows researchers to visualize how model behavior varies under different control sets (see Section 6 below as well as Figures S2, S3, S4, and S5) and time horizons (see Figures S6, S7, and S8). Note that while we view Figure 4 as the simplest way to plot initial model outputs, it is only one of many options. Researchers may choose to plot $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ as heatmaps or on the brain's surface, which would facilitate visualization of spatial patterning (see step 3a in Section 6 for an example).

6. **Compute control energy** (Timing: $<$ 1 second). The final step is to summarize the control signals into *control energy*. This is done by numerically integrating the control signals over time. In this protocol, we use Simpson's rule—an extension

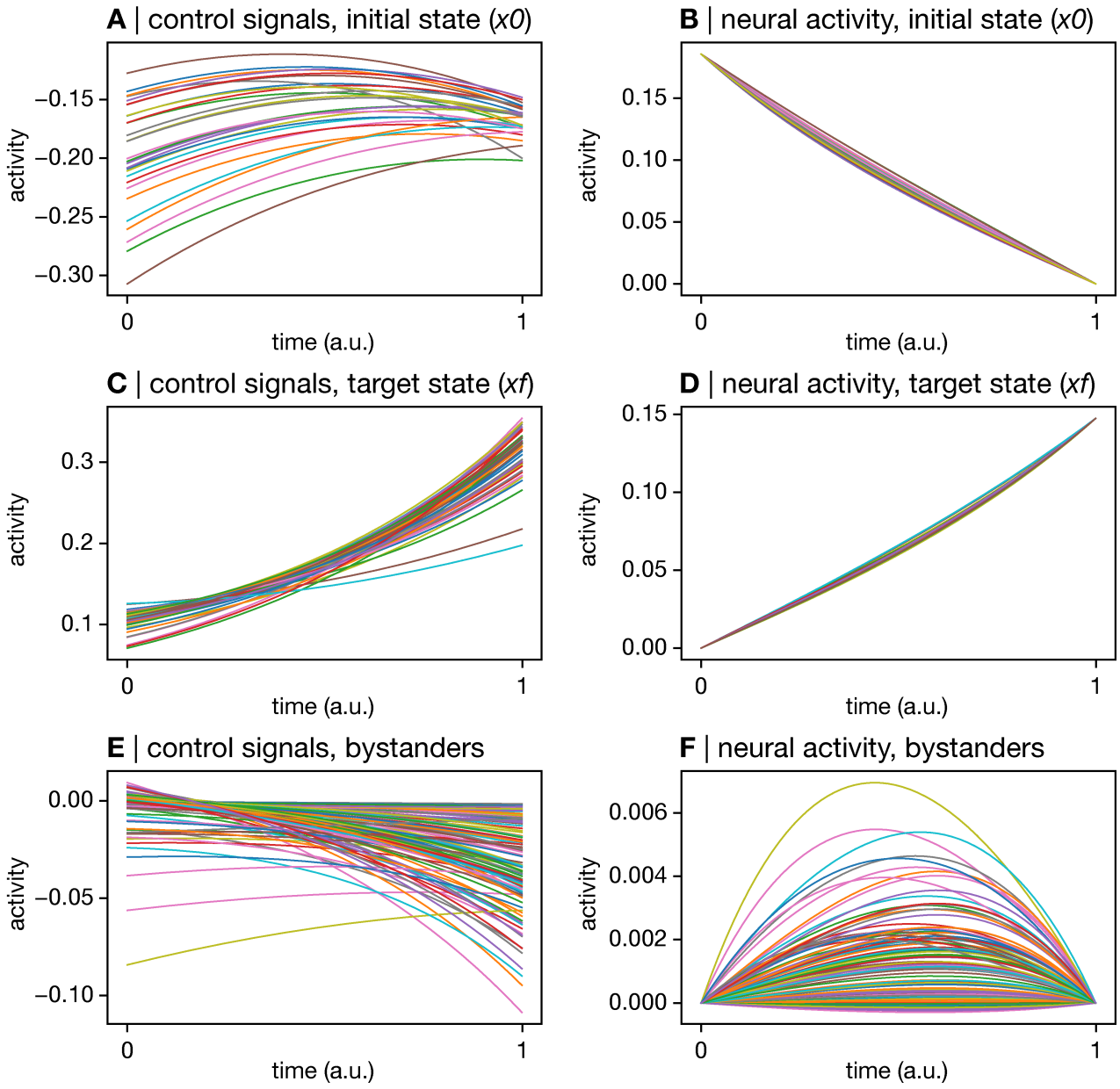

FIG. 4
Visualize the control signals and the state trajectory. For a given state transition, the control signals ($\boldsymbol{u}(t)$, left column) and the state trajectory ($\boldsymbol{x}(t)$, right column) should be visualized. Here, owing to our use of binary brain states, we group this visualization by nodes in the initial state ($x0$, top row), the target state (xf, middle row), and the remaining nodes (bystanders, bottom row). This plot provides intuition on model behavior by showing the kinds of control signals that are driving specific changes in neural activity. The top row shows that the model is driving negative time-varying control signals into the nodes of the initial state (**A**), which drives their activity to 0 over time (**B**). The middle row shows that the model is driving positive time-varying control signals into the nodes of the target state (**C**), which drives their activity from 0 to ∼0.15 over time (**D**). The bottom row shows that the model is driving diverse time-varying control signals into the bystander nodes (**E**), which are in turn guiding changes in these regions' activity (**F**).

of the trapezoidal rule that fits a polynomial through neighboring sets of points—to achieve this integration, yielding a vector of node-level energy:

```
# integrate control signals to get control energy
node_energy = integrate_u(control_signals)

print(node_energy.shape)
print(np.round(node_energy[:5], 2))
```

```
(200,)
[21.13 37.65 23.55 21.55 28.34]
```

Finally, these node-level energies can be summed to produce a single estimate of *control energy*:

```python
# summarize nodal energy
energy = np.sum(node_energy)

print(np.round(energy, 2))
```

```
2604.71
```

Wrapping Pathway A for ease of use

Steps 1-6 outline how to extract the *control energy* for a single control task, which we defined as completing a state transition between the visual system and the default mode system using control signals delivered to all system nodes (to view the above steps in a single notebook, see here). Alternatively, researchers may want to examine many control tasks within the context of a single study. Thus, in `nctpy` we include a *Python* class called `ComputeControlEnergy` that wraps all of the above steps (excluding step 5) and applies them over a list of control tasks. In addition to an adjacency matrix, `ComputeControlEnergy` expects a dictionary as input wherein each entry is a control task that includes: (i) an initial state, $x0$; (ii) a target stage, xf; (iii) a control set, B; (iv) a matrix of state trajectory constraints, S; and (v) a constraint parameter, *rho*. For example, using the `states` variable defined above, this dictionary could be created as follows:

```python
# initialize list of control tasks
control_tasks = []

# define control set using a uniform full control set
# note, here we use the same control set for all control tasks
control_set = np.eye(n_nodes)

# define state trajectory constraints
# note, here we constrain the full state trajectory equally for all control tasks
trajectory_constraints = np.eye(n_nodes)

# define mixing parameter
# note, here we use the same rho for all control tasks
rho = 1

# assemble control tasks
n_states = len(state_labels)
for initial_idx in np.arange(n_states):
    initial_state = normalize_state(states == initial_idx)  # initial state
    for target_idx in np.arange(n_states):
        target_state = normalize_state(states == target_idx)  # target state

        control_task = dict()  # initialize dict
        control_task["x0"] = initial_state  # store initial state
        control_task["xf"] = target_state  # store target state
        control_task["B"] = control_set  # store control set
        control_task["S"] = trajectory_constraints  # store state trajectory constraints
        control_task["rho"] = rho  # store rho
        control_tasks.append(control_task)
```

Note that for simplicity, the above section of code assumes the same control set (B), trajectory constraints (S), and ρ for all control tasks, but researchers may vary these parameters over transitions according to their needs. Next, we initialize and run `ComputeControlEnergy`:

```python
# compute control energy across all control tasks
compute_control_energy = ComputeControlEnergy(
    A=adjacency, control_tasks=control_tasks, system="continuous", c=1, T=1
)
compute_control_energy.run()
```

`ComputeControlEnergy` will perform matrix normalization internally, and hence here we input A rather than A_{norm}. Apart from the `control_tasks` variable, `ComputeControlEnergy` also requires that the user specify the time system (`system='continuous'`), normalization constant (`c=1`), and time horizon (`T=1`) as input arguments. Once instantiated, `ComputeControlEnergy.run()` will run steps 1-6 (excluding step 5). Once completed, a single estimate of *control energy* per control task will be stored in `ComputeControlEnergy.E`. Note that `ComputeControlEnergy` will not output the state trajectory, control signals, or node-level energy. These energy values can be trivially reshaped into a matrix and visualized (Figure 5):

```
# reshape energy into matrix
energy_matrix = np.reshape(compute_control_energy.E, (n_states, n_states))

# subtract lower triangle from upper to examine energy asymmetries
energy_matrix_delta = energy_matrix.transpose() - energy_matrix

f, ax = plt.subplots(1, 3, figsize=(7, 4))

# plot energy matrix
sns.heatmap(
    energy_matrix,
    ax=ax[0],
    square=True,
    linewidth=0.5,
    cbar_kws={"label": "energy", "shrink": 0.25},
)

# plot without self-transitions
# setup mask to exclude persistence energy (i.e., transitions where i==j)
mask = np.zeros_like(energy_matrix)
mask[np.eye(n_states) == 1] = True
sns.heatmap(
    energy_matrix,
    ax=ax[1],
    square=True,
    linewidth=0.5,
    cbar_kws={"label": "energy", "shrink": 0.25},
    mask=mask,
)

# plot energy asymmetries
mask = np.triu(np.ones_like(energy_matrix, dtype=bool))
sns.heatmap(
    energy_matrix_delta,
    ax=ax[2],
    square=True,
    linewidth=0.5,
    cbar_kws={"label": "energy (delta)", "shrink": 0.25},
    mask=mask,
    cmap="RdBu_r",
    center=0,
)

for cax in ax:
    cax.set_ylabel("initial state (x0)")
    cax.set_xlabel("target state (xf)")
    cax.set_yticklabels(state_labels, rotation=0, size=6)
    cax.set_xticklabels(state_labels, rotation=90, size=6)
f.tight_layout()
plt.show()
```

In Figure 5A, the diagonals represent the energy associated with control tasks that have identical initial and target states. We refer to these energies as persistence energy [20, 73, 145], which we interpret as the amount of *control energy* required to maintain a given brain state (see Figure S9 for a persistence energy example of Figure 4). Persistence energy is typically lower than the energy for control tasks wherein the initial and target states differ (i.e., the off-diagonal elements of `energy_matrix`). Thus, in order to better visualize the variance in energy across state transitions, we recommend also plotting `energy_matrix` excluding the diagonal elements (Figure 5B). Additionally, as the energy associated with transitioning from $x0$ to xf is not necessarily equivalent to that associated with the reverse direction, researchers may subtract the upper and lower triangles of `energy_matrix` to examine energy asymmetries (Figure 5C). Indeed, we have done this in our previous work (see [51, 73]).

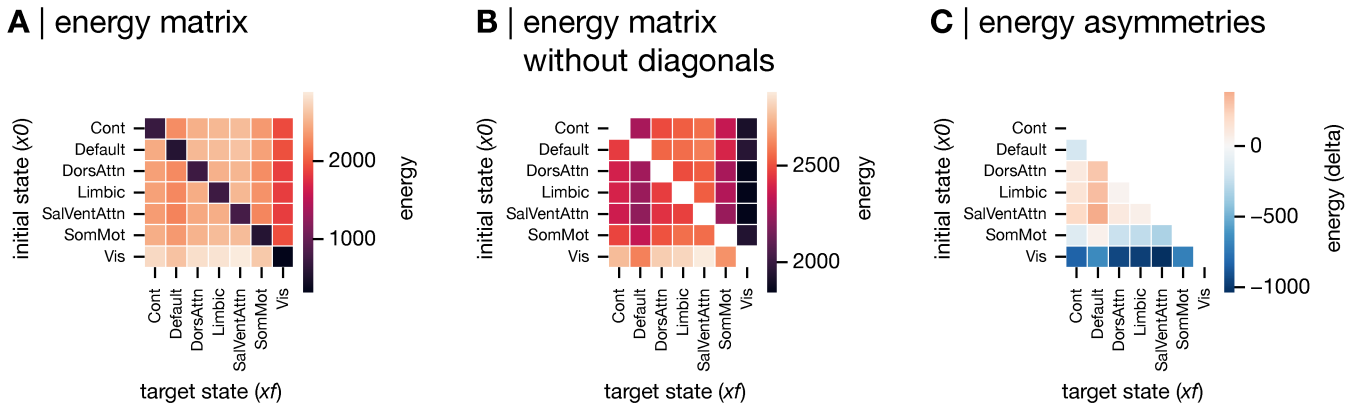

FIG. 5
Visualize control energy. In cases where multiple state transitions are considered, we recommend visualizing *control energy* using the following heatmaps. **A**, Full energy matrix. The full energy matrix shows energy for all state transitions, including those where the initial and target state are the same (diagonal elements). This form of energy is referred to as persistence energy and is interpreted as the amount of effort required to maintain neural activity in a given state. Persistence energy is typically lower than the energy associated with transitioning between different states. **B**, Between-state energy matrix. In order to visualize variance in *control energy* for transitions between states, we recommend plotting the energy matrix without the diagonal as well. **C**, Energy asymmetry matrix. Finally, we recommend subtracting the transpose of the energy matrix and visualizing the lower (or upper) triangle of the ensuing asymmetry matrix. Doing so allows researchers to see the asymmetries present in the *control energy*.

Variations to Pathway A

All of the above constitutes our complete protocol for calculating *control energy*. However, there are multiple variations to the above protocol that researchers may wish to consider depending on their research goals. In this section, we illustrate a selection of these variants that are likely to be of broad interest to the field of neuroscience (Figure 3C). These variations include (A) studying non-binary brain states; (B) implementing partial and non-uniform control sets; and (C) examining directed structural connectomes.

Non-binary brain states

In Section IX A, we illustrated a state transition between the visual system and the default mode system using a binary definition of brain states extracted from a canonical system-level grouping of brain regions [66]. Below, we provide an example of using non-binary brain states instead. This example draws on aforementioned work from Cornblath *et al.* [20], who modeled the energy required to transition between clusters of rs-fMRI activity.

3A) *Define a control task: non-binary brain states.* To define non-binary brain activity states, we cluster rs-fMRI data along the time dimension to extract co-activation states [20]. To achieve this goal, we first load the rs-fMRI data for 253 participants extracted from the same parcellation that defined our structural connectome. Then, we concatenate these time series end-to-end across subjects:

```
# load resting-state time series
rsfmri_file = 'pnc_schaefer200_rsts.npy'
rsfmri = np.load(os.path.join(datadir, rsfmri_file))

n_trs = rsfmri.shape[0]
n_nodes_rsfmri = rsfmri.shape[1]
n_subs = rsfmri.shape[2]
print('n_trs, {0}; n_nodes, {1}; n_subs, {2}'.format(n_trs, n_nodes_rsfmri, n_subs))

rsfmri_concat = np.zeros((n_trs * n_subs, n_nodes_rsfmri))
print(rsfmri_concat.shape)

for i in np.arange(n_subs):
    # z score and concatenate subject i's time series
    start_idx = i * n_trs
```

```
    end_idx = start_idx + n_trs
    rsfmri_concat[start_idx:end_idx, :] = sp.stats.zscore(rsfmri[:, :, i], axis=0)
```

```
n_trs, 120; n_nodes, 200; n_subs, 253
(30360, 200)
```

This process assigns empirical time series to each of the system nodes comprising 30,360 time points (120 TRs by 253 participants) of rs-fMRI data. Next, we cluster these data in time using K-means and visualize the corresponding co-activation states on the cortical surface. To generate this plot, we include a function called `surface_plot` that utilizes tools from Nilearn (https://nilearn.github.io/stable/index.html):

```
# extract 5 clusters of activity
n_clusters = 5
kmeans = KMeans(n_clusters=n_clusters, random_state=0).fit(rsfmri_concat)

# extract cluster centers. These represent dominant patterns of recurrent activity over time
centroids = kmeans.cluster_centers_
print(centroids.shape)

# plot centroids on brain surface
lh_annot_file = ("/path/to/schaefer/files/lh.Schaefer2018_200Parcels_7Networks_order.annot")
rh_annot_file = ("/path/to/schaefer/files/rh.Schaefer2018_200Parcels_7Networks_order.annot")
fsaverage = datasets.fetch_surf_fsaverage(mesh="fsaverage5")

for cluster in np.arange(n_clusters):
    f = surface_plot(
        data=centroids[cluster, :],
        lh_annot_file=lh_annot_file,
        rh_annot_file=rh_annot_file,
        fsaverage=fsaverage,
        order="lr",
        cmap="coolwarm",
    )
```

```
(5, 200)
```

Figure 6 illustrates the centroids for 2 of the 5 clusters we extracted using K-means. As we are focused on illustrating a single state transition for the purposes of this protocol, the remaining 3 centroids are not shown (see Cornblath *et al.* [20] for detailed discussion of all 5 clusters). These centroids represent patterns of activity that recur throughout our concatenated time series. The spatial patterning of these 2 centroids indicate activity concentrated in visual cortex (Figure 6A) and the default mode system (Figure 6B), respectively. Using the same functions outlined above, we can extract this pair of centroids as brain states and recompute the control energy:

```
# extract visual cluster is initial state
initial_state = centroids[1, :]
# extract default mode cluster as target state
target_state = centroids[4, :]

# normalize state magnitude
initial_state = normalize_state(initial_state)
target_state = normalize_state(target_state)

# get the state trajectory and the control signals
state_trajectory, control_signals, numerical_error = get_control_inputs(
    A_norm=adjacency_norm,
    T=time_horizon,
    B=control_set,
    x0=initial_state,
    xf=target_state,
    system=system,
    rho=rho,
    S=trajectory_constraints,
)

# get energy
node_energy = integrate_u(control_signals)
energy = np.sum(node_energy)
```

As stated above in step 5, the temporal unfolding of $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ can be visualized using line plots (Figure 6B top, C top). Alternatively, to illustrate the spatial patterning of the state transition, $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ can be visualized on the cortical surface for specific time points (Figure 6B bottom, C bottom):

```
timepoints_to_plot = np.arange(0, state_trajectory.shape[0], int(state_trajectory.shape[0] / 5))

for timepoint in timepoints_to_plot:
    f = surface_plot(
        data=state_trajectory[timepoint, :],
        lh_annot_file=lh_annot_file,
        rh_annot_file=rh_annot_file,
        fsaverage=fsaverage,
        order="lr",
        cmap="coolwarm"
    )
```

Partial and variable control sets

In Section IX A, we illustrated a state transition controlled via a *uniform full control set*. Such a control set amounts to assigning all nodes of the system the same degree of control over system dynamics. Below, we provide examples of using alternatives to this regime that involve using variable (instead of uniform) and partial (instead of full) control sets.

3B) *Define a control task: uniform **partial** control set*. Defining a *uniform partial control set* is straight forward. Instead of assigning the $N \times N$ identity matrix to B, we select specific diagonal elements to assign the value of 1, and assign 0 to the remaining diagonal (and non-diagonal) elements. Here, we illustrate the example of selecting the bystander regions (see Figure 4E, F) as our control set:

```
# specify a uniform partial control set: some nodes are control nodes
# and all control nodes are assigned equal control weight
bystanders = np.logical_and(
    initial_state == 0, target_state == 0
)  # use initial state and final state to find bystanders. note, this only works for binary
    states
control_set = np.zeros((n_nodes, n_nodes))  # initialize control nodes matrix
control_set[bystanders, bystanders] = 1  # set bystanders to control nodes
```

Figure S2 shows the results of generating $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ under the above control set. Note that the above code only works for binary brain states, because in this case bystanders can be trivially defined as nodes with no activity in either the initial or target states. This fact does not imply that partial control sets cannot be used for non-binary brain states. In Figure S2, we observe that the control signals are 0 for both $x0$ and xf, indicating that they received no control signals. Additionally, the control signals for the bystanders, as well as the neural activity of all nodes, has changed substantially compared to Figure 4. Notably, the control signals are several orders of magnitude greater for this *uniform partial control set* compared to the *uniform full control set* used above. In turn, although this state transition completes successfully, the energy we observe here is also several orders of magnitude greater (energy = 6.64×10^9). See Figures S3, S4, and S5 for more examples of *uniform partial control sets*, including some for which the state transition does not complete. While the code implementation of a *uniform partial control set* is straight forward, researchers must ensure that their control set is large enough to achieve numeric stability (see Section VII).

3C) *Define a control task: **a priori variable** full control set*. Instead of assigning all nodes of the system the same degree of control over dynamics, researchers may want to make statements about which nodes should have more or less control according to their hypotheses. One way that this can be achieved is by using node-level annotation maps [146, 147] to assign control weights. For a single annotation map—which we assume is stored in `'neuromap.npy'`—assigning weights can be achieved in the following way:

```
# helper func for printing descriptive stats
def print_stats(x):
    print(
        "min={:.2f}; max={:.2f}; mean={:.2f}; std={:.2f}; skew={:.2f}; kurt={:.2f}".format(
            np.min(x),
            np.max(x),
            np.mean(x),
            np.std(x),
            sp.stats.skew(x),
```

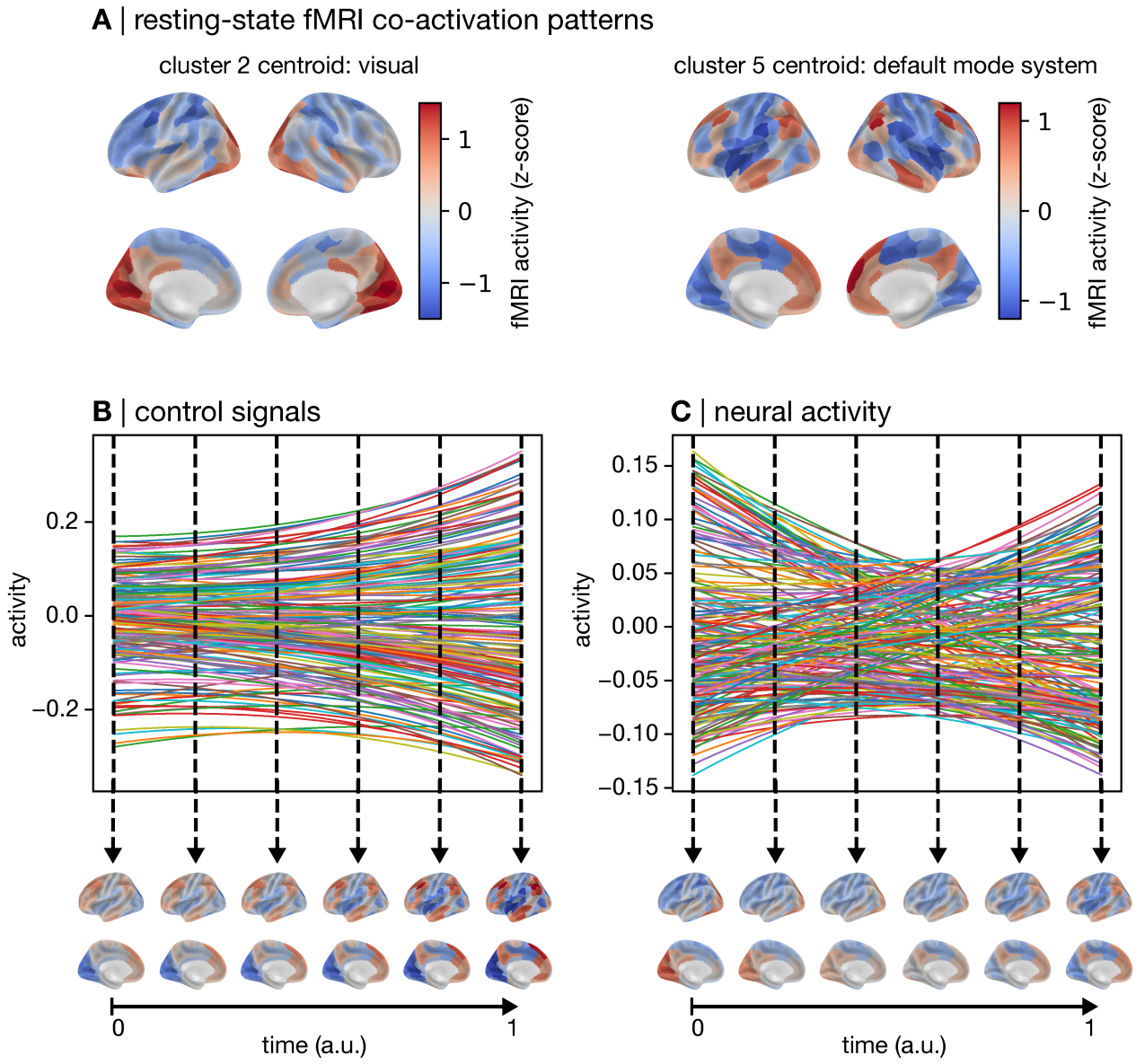

FIG. 6
Control signals and state trajectory for non-binary brain states derived from resting-state fMRI. We applied K-means clustering to resting-state fMRI time-series to extract patterns of co-activation. These patterns were used as non-binary brain states for network control theory (NCT) analysis. **A**, Resting-state fMRI clusters that represent visual system activity (left) and default mode activity (right). The control signals and the state trajectory were modeled by assigning the visual system to the initial state and the default mode system to the target state. **B**, Control signals visualized using line plots (top) and on the cortical surface for select time points (bottom). **C**, State trajectory (neural activity) visualized using line plots (top) and on the cortical surface for select time points (bottom).

```
            sp.stats.kurtosis(x),
        )
    )

neuromap_file = 'neuromap.npy'
neuromap = np.load(os.path.join(datadir, neuromap_file))
print(neuromap.shape)
print_stats(neuromap)

control_set = np.zeros((n_nodes, n_nodes))  # initialize B matrix
control_set[np.diag_indices(n_nodes)] = neuromap  # set weights using neuromap
```

```
(200,)
```

```
min=-0.41; max=0.42; mean=-0.04; std=0.20; skew=0.25; kurt=-0.54
```

The above code demonstrates that our annotations, and therefore our control set, include both positive and negative values. The sign of a given weight will determine how a positive control signal delivered to that node influences its neural state; a positive weight will cause a positive change in neural activity, while a negative weight will cause a negative change. As control signals can also carry positive and negative values over time (see Figure 4), this behavior has no impact on *control energy*; a positive control signal driven into a node using a positive control weight will cause the same change in neural state as a negative control signal driven via a negative control weight. However, it does impact the interpretation of $\boldsymbol{u}(t)$. Thus, to simplify the interpretation of the control weights, and of $\boldsymbol{u}(t)$, we suggest adding a constant (1) as well as the absolute minimum value to the annotation map:

```
# modify neuromap so that its minimum value is 1
neuromap += 1 + np.abs(np.min(neuromap))
print_stats(neuromap)

control_set = np.zeros((n_nodes, n_nodes))  # initialize B matrix
control_set[np.diag_indices(n_nodes)] = neuromap  # set weights using neuromap
```

```
min=1.00; max=1.84; mean=1.38; std=0.20; skew=0.25; kurt=-0.54
```

This process will create a set of weights with a minimum value of 1 and, in this case, a maximum value of 1.84 (see Figure S10 for a plot $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ derived from this control set). This fact simplifies our interpretation. For example, because all weights are positive, we can say that the highest weight node has 1.84 times more control over system dynamics compared to the lowest weight node. Additionally, whether a control signal drives a positive or negative change in neural state now depends solely on its own sign at a given point in time.

A recent study by Singleton *et al.* [30] utilized an *a priori variable full control set* by assigning control weights according to a range of seretonin receptor maps. Singleton *et al.* [30] found that the *control energy* associated with their state transition was lowest when using a 5-HT2a receptor map compared to 5-HT1a, 5-HT1b, 5-HT4, and 5-HTT maps. This result suggests that, compared to other receptors, the spatial patterning of 5-HT2a receptors yielded the most efficient state transition (i.e., by reducing *control energy* the most). The authors subsequently replicated their results using N,N-Dimethyltryptamine [31]. However, to draw this conclusion, researchers need to mindful of the following caveat. In our model, increasing the total amount of control necessarily reduces energy. This relation exists because the task of completing a given state transition is easier for the model when any node is granted a greater degree of control over dynamics than it had previously, leading to smaller amplitude control signals and thus lower energy. For example, if we compared energy between our *uniform full control set* and our above *variable full control set*, energy would be trivially lower for the latter. This difference occurs because all weights on the diagonal of B are 1 in our *uniform full control set*, while all but one of the weights in our *variable full control set* are > 1. This issue is pertinent for researchers who want to compare different *variable full control sets* (as in [30–32]), because any comparison of *control energy* across two different annotation maps needs to account for differences in those maps' distributions. The simplest solution to this problem is to take the rank of each annotation map and then rescale the ranks to be between 1 and 2. `normalize_weights` performs this normalization:

```
neuromap_file_1 = 'neuromap.npy'
neuromap_1 = np.load(os.path.join(datadir, neuromap_file_1))
neuromap_1_norm = normalize_weights(neuromap_1)

neuromap_file_2 = 'neuromap2.npy'
neuromap_2 = np.load(os.path.join(datadir, neuromap_file_2))
neuromap_2_norm = normalize_weights(neuromap_2)

print_stats(neuromap_1)
print_stats(neuromap_1_norm)
print_stats(neuromap_2)
print_stats(neuromap_2_norm)
```

```
min=-0.41; max=0.42; mean=-0.04; std=0.20; skew=0.25; kurt=-0.54
min=1.00; max=2.00; mean=1.50; std=0.29; skew=-0.00; kurt=-1.20
min=-0.10; max=0.16; mean=-0.00; std=0.07; skew=0.59; kurt=-0.78
min=1.00; max=2.00; mean=1.50; std=0.29; skew=-0.00; kurt=-1.20
```

Once annotation maps have been normalized in this manner, differences in energy can only be attributed to differences in the (rank) spatial patterning between maps. This independence occurs because both annotation maps now conform to a uniform distribution. Note, energy derived from this normalization approach will still be lower than our *uniform full control set*.

3D) *Define a control task:* ***data-driven variable*** *full control set*. Instead of assigning variable weights *a priori*, researchers may assign them in a data-driven manner. In our recent work, we developed an approach for achieving this goal using gradient descent (see [51] for more details). Briefly, starting with a *uniform full control set*, this approach involves perturbing control nodes' weight one at a time by a constant arbitrary amount and measuring the corresponding change in energy. This process results in N estimates of *perturbed control energy* for a given state transition. As mentioned above, each of these perturbations will necessarily reduce *control energy* compared to the baseline *uniform full control set*, creating negative Δs. In turn, differences in Δ magnitude encode the relative importance of each node to completing a specific state transition; nodes with more negative energy Δs are more important:

```
# container for perturbed energies
energy_perturbed = np.zeros(n_nodes)

for node in tqdm(np.arange(n_nodes)):
    # start with a uniform full control set
    control_set = np.eye(n_nodes)

    # add arbitrary amount of additional control to node
    control_set[node, node] += 0.1

    # get perturbed control signals (u_p)
    _, control_signals, _ = get_control_inputs(
        A_norm=adjacency_norm,
        T=time_horizon,
        B=control_set,
        x0=initial_state,
        xf=target_state,
        system=system,
        rho=rho,
        S=trajectory_constraints,
    )

    # integrate control signals to get control energy
    node_energy = integrate_u(control_signals)

    # summarize nodal energy
    energy_perturbed[node] = np.sum(node_energy)

# check if perturbed energy is lower than original energy. Should print True
print(np.all(energy_perturbed < energy))

# calculate energy delta. these values will all be negative,
# indicating reduced energy compared to control_set=np.eye(n_nodes)
energy_delta = energy_perturbed - energy
```

```
True
```

Once estimated, these Δs can be used as weights on B to obtain an optimized version of *control energy*. Note that here we draw a distinction between *optimal* and *optimized*. The former refers to constraining the magnitudes of $\boldsymbol{x}(t)$ and $\boldsymbol{u}(t)$ in the optimization problem, whereas the latter refers to finding the weights that create the most efficient transition, irrespective of this constraint:

```
# re-compute energy using energy deltas as weights
# we do this by taking a single step down the
# gradient created by the energy deltas
learning_rate = 0.01  # set a learning rate for gradient descent
control_set = np.zeros((n_nodes, n_nodes))  # initialize container for optimized weights
control_set[np.diag_indices(n_nodes)] = (1 - energy_delta * learning_rate)  # step down gradient
control_set = (
    control_set / sp.linalg.norm(control_set) * sp.linalg.norm(control_set)
)  # normalize
# normalization ensures that the optimized weights have the same
# norm as control_set=np.eye(n_nodes)

# get optimized control signals
_, control_signals, _ = get_control_inputs(
    A_norm=adjacency_norm,
    T=time_horizon,
    B=control_set,
    x0=initial_state,
```

```
    xf=target_state,
    system=system,
    rho=rho,
    S=trajectory_constraints,
)

# integrate control signals to get control energy
node_energy_optimized = integrate_u(control_signals)
print(np.round(node_energy_optimized[:5], 2))

# summarize nodal energy
energy_optimized = np.sum(node_energy_optimized)
print(np.round(energy_optimized, 2))
```

```
[20.56 34.94 22.77 20.92 26.96]
2429.68
```

The above code uses `energy_delta` to define a gradient that we step down one time using a learning rate of 0.01. Stepping down this gradient yields a set of optimized weights, B_o, that we then use to re-estimate *control energy*. As expected, this new estimate of energy (2429.68) is lower than the energy derived from our *uniform full control set* (2604.71). Thus, we have found a set of control weights that optimize (i.e., reduce) our *control energy* in a data-driven way. Additionally, setting up this algorithm using gradient descent allows researchers to optimize energy over multiple steps, wherein each new set of optimized weights is calculated from the previous set. In `nctpy` we include a *Python* class called `ComputeOptimizedControlEnergy` that wraps the above optimization steps and allows researchers to define their own learning rate and number of gradient steps:

```
control_task = dict()  # initialize dict
control_task["x0"] = initial_state  # store initial state
control_task["xf"] = target_state  # store target state
control_task["S"] = trajectory_constraints  # store state trajectory constraints
control_task["rho"] = rho  # store rho
compute_opt_control_energy = ComputeOptimizedControlEnergy(
    A=adjacency,
    control_task=control_task,
    system,
    c=1,
    time_horizon,
    n_steps=2,
    lr=learning_rate,
)
compute_opt_control_energy.run()
```

`ComputeOptimizedControlEnergy` is similar to `ComputeControlEnergy` but has some notable differences. Like `ComputeControlEnergy`, `ComputeOptimizedControlEnergy` will perform matrix normalization internally, thus we input A rather than A_{norm}. Additionally, `ComputeOptimizedControlEnergy` requires that users specify the time system (`system='continuous'`), normalization constant (`c=1`), and time horizon (`T=1`) as input arguments. The differences are as follows. First, `ComputeOptimizedControlEnergy` only accepts a single control task dictionary, rather than a list of tasks. Second, `ComputeOptimizedControlEnergy` requires that users also specify the number of gradient steps (`n_steps=2`) and the learning rate (`lr=0.01`) as inputs. Once instantiated, `ComputeOptimizedControlEnergy.run()` will run the above optimization steps. Once completed, optimized energy at each gradient step will be stored in `ComputeControlEnergy.E_opt` as a vector of length N_s, where N_s is the number of steps. The corresponding optimized control weights will be stored in `ComputeControlEnergy.B_opt` as an $N_s \times m$ matrix.

Directed structural connectome

In Section IX A, we performed NCT analysis using an undirected structural connectome derived from the human brain. However, our protocol is designed to work with directed connectomes as well. Thus, as a final variation on Pathway A, we present results from a directed connectome obtained in the mouse brain. As discussed in Section VIII, our model assumes that A_{ij} encodes the edge connecting *node j to node i*. Provided that this assumption is met, Pathway A can be run without modification.

Using the Allen Mouse Brain Connectivity Atlas [24, 25, 148], we extracted the ipsilateral directed connectivity from 43 regions in the mouse isocortex. These 43 regions were grouped into 6 systems: auditory, lateral, medial, prefrontal, somatomotor, and visual. Following [148], we combined the prefrontal and medial systems, as well as 3 regions from the somatomotor system, to create the mouse default mode system. Then, using `ComputeControlEnergy`, we estimated the energy required to transition from the default mode to the lateral, visual, and auditory systems and back again. Similar to the undirected human connectome (see Figure 5), this process yielded energy asymmetries (Figure 7A). We repeated this process using a symmetric version of the mouse connectome ($A_s = \frac{A+A^\top}{2}$; Figure 7B) and examined how energy asymmetries differed between the directed and undirected cases (Figure 7C); see here for *Python* code. For both the directed (Figure 7A) and undirected connectomes (Figure 7B), we found that *control energy* was lower when transitioning to the default mode compared to from the default mode. Critically, these observed energy asymmetries were larger for the directed connectome compared to the undirected connectome (Figure 7C); the lateral energy asymmetry was larger by 131 units (20% larger), the visual asymmetry was larger by 16 units (9% larger), and the auditory asymmetry was larger by 92 units (35% larger). Thus, the presence of directed edges increased the energy asymmetries observed in our model.

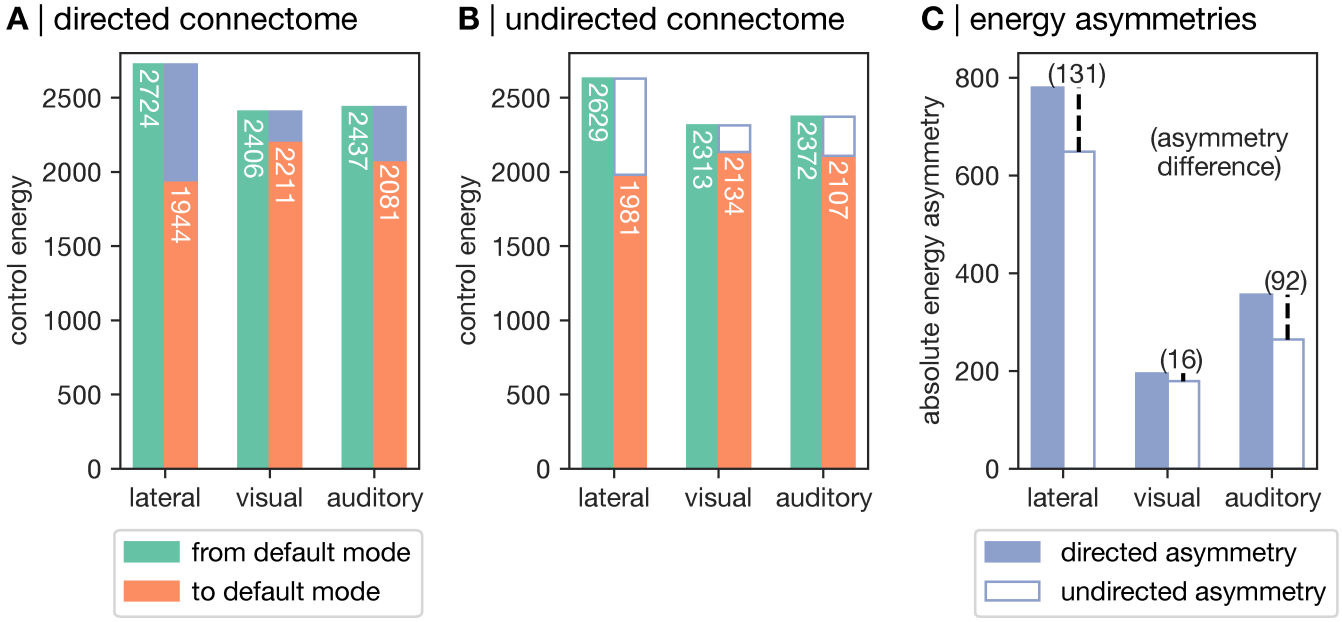

FIG. 7
Energy asymmetries are larger for the directed than the undirected mouse connectome. Our protocol can be applied to connectomes with directed edges as well as undirected edges. In the directed case, our protocol assumes that A_{ij} encodes the edge connecting *node j to node i*. Here, we estimated *control energy* using the Allen Mouse Brain Connectivity Atlas. **A**, *Control energy* associated with transitioning from the default mode (DMN) to the lateral, visual, and auditory systems (green) and back again (orange) in the directed mouse connectome. *Control energy* associated with transitioning to the default mode was lower than the reverse direction, indicating a clear asymmetry (blue). **B**, *Control energy* estimated in the undirected mouse connectome. We recomputed energy using a symmetrized version of the mouse connectome ($As = A + A^\top$) and observed the same set of energy asymmetries. **C**, Differences in energy asymmetries between the directed and undirected mouse connectome. We observed that the size of the energy asymmetries were larger for the directed than the undirected mouse connectome.

B. Protocol Pathway B: Average Controllability

Pathway A is the primary component of our protocol. Implementing these steps assumes that researchers are interested in studying a specific set of state transitions defined in accordance with their research questions and hypotheses. In the absence of such hypotheses, researchers may instead wish to examine nodes' general capacity to control a broad range of unspecified state transitions. To support these types of hypotheses, we present a complementary pathway to our protocol that yields estimates of *average controllability*, where higher values indicate that a region is better positioned in the network to control dynamics (see section II):

1. **Compute average controllability** (Timing: discrete time, <1 second for 200 nodes; continuous time, $10-20$ seconds for 200 nodes).

```
# compute average controllability
average_controllability = ave_control(adjacency_norm, system)
```

`average_controllability` will be a vector containing the *average controllability* of each node of the system.

2. **Visualize average controllability**. As *average controllability* is a regional metric, we can simply plot its distribution of values on the surface of the cortex (Figure 8).

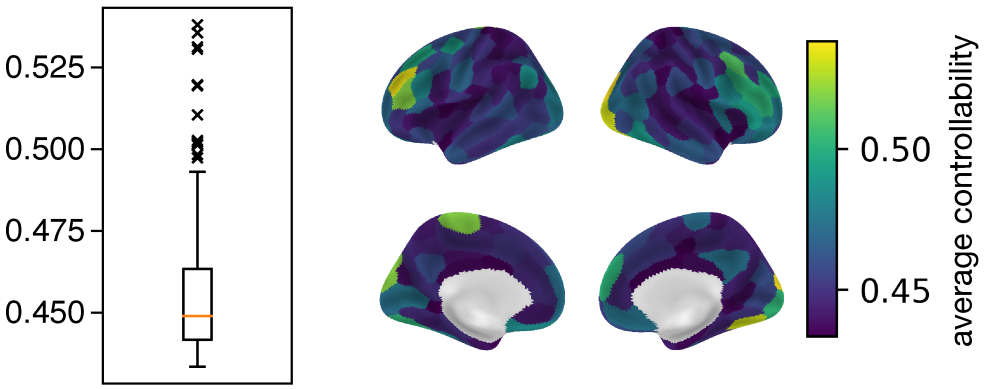

FIG. 8
Average controllability. Each system node receives an impulse of equal magnitude. Nodes with higher *average controllability* are able to broadcast that impulse throughout the system to a greater extent compared to nodes with lower *average controllability*. Thus, nodes with high *average controllability* are better positioned within the network to control dynamics. Distribution of *average controllability* values are displayed using a box plot (left) as well as projected onto the cortical surface (right). In the box plot, the orange line represents the median, the box spans the middle 50% of the data, the whiskers span 1.5 times the interquartile range on either side, and the crosses represent outliers beyond this limit.

X. ANTICIPATED RESULTS

The final outputs of our protocol will depend on whether researchers choose to follow Pathway A (see Section IX A) or Pathway B (see Section IX B). For the former, the output will be one estimate of *control energy* per control task, or one estimate per brain region per task if energy was not summarized across regions. This value will be positive and can be thought of as the amount of effort the model has to exert in order to complete a specific control task; higher energy corresponds to greater effort. For the latter, output will be one estimate of *average controllability* per brain region; a regional map of control over system dynamics. These regional values will also be positive. Greater *average controllability* indicates that regions are better positioned within the network's topology to broadcast an impulse, and as such may better orchestrate control of brain dynamics.

What can researchers do with these outputs? The answers to this question are diverse and depend heavily on the researcher's goals. As we discussed in Section III, we have used NCT to investigate a range of research questions that spanned from examining the influence of topology [10, 49, 50], to predicting state transitions observed in functional data [14, 73], to studying individual differences, including psychosis symptoms [69], executive function [71], and sex effects [19]. Providing detailed guidance on each of these applications is beyond the scope of this protocol. However, to conclude this protocol, we outline the use of null network models as an initial analysis that we believe is an essential step irrespective of researchers' study goals.

1. *Null network models*

Null models allow researchers to examine the extent to which different aspects of topology explain NCT model outputs. As discussed in Ref. [149], these null models take different forms, including edge rewiring, generative models of surrogate networks, and spatially-preserved node permutation. Of these different forms, the appropriate choice will depend on the research question. Here, in order to understand the extent to which topology informs *control energy*, we focus on null models that rewire an empirical adjacency matrix (see Ref. [69] for an example of spatially-preserved node permutation

used to compare maps of *average controllability* with other properties of network topology). In this case, a null network model involves randomly swapping the edges of the adjacency matrix n times subject to certain constraints—for example, while preserving the spatial embedding of the nodes as well as the degree or strength distribution [149, 150]—and recalculating energy upon each rewired matrix. This process generates an empirical null distribution that observed *control energy* can be compared against. For instance, *control energy* that is lower than expected under the null suggests that NCT was able to leverage properties of network topology, beyond those preserved by the null model, to complete a given state transition. In turn, by deploying a range of null models that each preserve different topological properties, researchers can systematically probe the aspects of topology that explain their observed outputs.

Using the undirected human connectome, we provide an example of the above approach using two of our binary state transitions: the default mode to visual transition and the default mode to ventral attention (VAN) transition. We chose these transitions as they represent two control tasks with strong but opposing energy asymmetries (see Figure 5). For each transition, we recompute the *control energy* for each direction under two null models. The first preserves the spatial embedding of the nodes as well as the strength distribution (strength-preserving). The second preserves spatial embedding and the strength sequence of the nodes (sequence-preserving). That is, the sequence-preserving null preserves the strength of each node as it was in the original connectome. By contrast, the strength-preserving null only preserves the distribution of strength across the network; the strength of each node is allowed to change. The sequence-preserving null is a more stringent test than the strength-preserving null as it preserves how strength is embedded in the connectome [150]. We begin by loading the coordinates of our nodes in 3 dimensions:

```
# null networks
centroids = pd.read_csv(
    os.path.join(datadir, "pnc_schaefer200_centroids.csv")
)  # load coordinates of nodes
centroids.set_index("node_names", inplace=True)
print(centroids.head())
```

```
             vox_x  vox_y  vox_z
node_names
LH_Vis_1       121    149     69
LH_Vis_2       123    174     65
LH_Vis_3       143    166     70
LH_Vis_4       107    164     74
LH_Vis_5       124    192     66
```

Then, we use those coordinates to define a distance matrix that encodes the physical distance between node pairs:

```
distance_matrix = distance.pdist(centroids, 'euclidean')  # get euclidean distances between nodes
distance_matrix = distance.squareform(distance_matrix)  # reshape to square matrix
```

Finally, we define our control task and compute our nulls using the included function, `geomsurr` [150]:

```
# extract initial state
initial_state = states == state_labels.index("Vis")  # 'Vis' or 'SalVentAttn'
initial_state = normalize_state(initial_state)  # normalize

# extract target state
target_state = states == state_labels.index("Default")
target_state = normalize_state(target_state)  # normalize

# compute true control energy
_, control_signals, _ = get_control_inputs(
    A_norm=adjacency_norm,
    T=time_horizon,
    B=control_set,
    x0=initial_state,
    xf=target_state,
    system=system,
    rho=rho,
    S=trajectory_constraints,
)
node_energy = integrate_u(control_signals)  # integrate control signals
energy = np.sum(node_energy)  # get energy

# run permutation
n_perms = 5000  # number of permutations

# containers for null distributions
```

```
energy_null_sp = np.zeros(n_perms)
energy_null_ssp = np.zeros(n_perms)

for perm in tqdm(np.arange(n_perms)):
    # rewire adjacency matrix using geomsurr
    _, Wsp, Wssp = geomsurr(W=adjacency, D=distance_matrix, seed=perm)
    # Wsp is the adjacency matrix rewired while preserving spatial embedding and the strength distribution
    # Wssp is the adjacency matrix rewired while preserving spatial embedding and the strength sequence
    # this python implementation is included with our toolbox, but if you use these nulls
    # in your own work, please cite:
    #       Roberts et al. NeuroImage (2016), doi:10.1016/j.neuroimage.2015.09.009

    # compute control energy for Wsp
    Wsp = matrix_normalization(Wsp, system)
    _, control_signals, _ = get_control_inputs(
        A_norm=Wsp,
        T=time_horizon,
        B=control_set,
        x0=initial_state,
        xf=target_state,
        system=system,
        rho=rho,
        S=trajectory_constraints,
    )
    node_energy = integrate_u(control_signals)
    energy_null_sp[perm] = np.sum(node_energy)

    # compute control energy for Wssp
    Wssp = matrix_normalization(Wssp, system)
    _, control_signals, _ = get_control_inputs(
        A_norm=Wssp,
        T=time_horizon,
        B=control_set,
        x0=initial_state,
        xf=target_state,
        system=system,
        rho=rho,
        S=trajectory_constraints,
    )
    node_energy = integrate_u(control_signals)
    energy_null_ssp[perm] = np.sum(node_energy)

# plot
f, ax = plt.subplots(1, 2, figsize=(7, 3))
null_plot(
    observed=energy,
    null=energy_null_sp,
    xlabel="strength-preserving",
    ax=ax[0],
)
null_plot(
    observed=energy,
    null=energy_null_ssp,
    xlabel="sequence-preserving",
    ax=ax[1],
)
f.tight_layout()
plt.show()
```

Figure 9 displays the energy associated with transitioning from the visual cortex to the DMN (Figure 9A) and back again (Figure 9B), as well from the VAN to the DMN (Figure 9C) and back again (Figure 9D). In each panel of Figure 9, the strength-preserving null is shown on the left and the sequence-preserving null is shown on the right. These results provide several insights. First, as mentioned above, both transitions show energy asymmetries but in opposite directions. The energy associated with transitioning from visual cortex to the DMN is larger (energy = 2605) compared to the reverse direction (energy = 1947). By contrast, the energy associated with transitioning from the VAN to the DMN is lower (energy = 2218) compared to the reverse direction (energy = 2601). Note that the former result represents an exception to the general finding that energy is lower when transitioning to the DMN than from it (see Figure 5). This pattern of findings

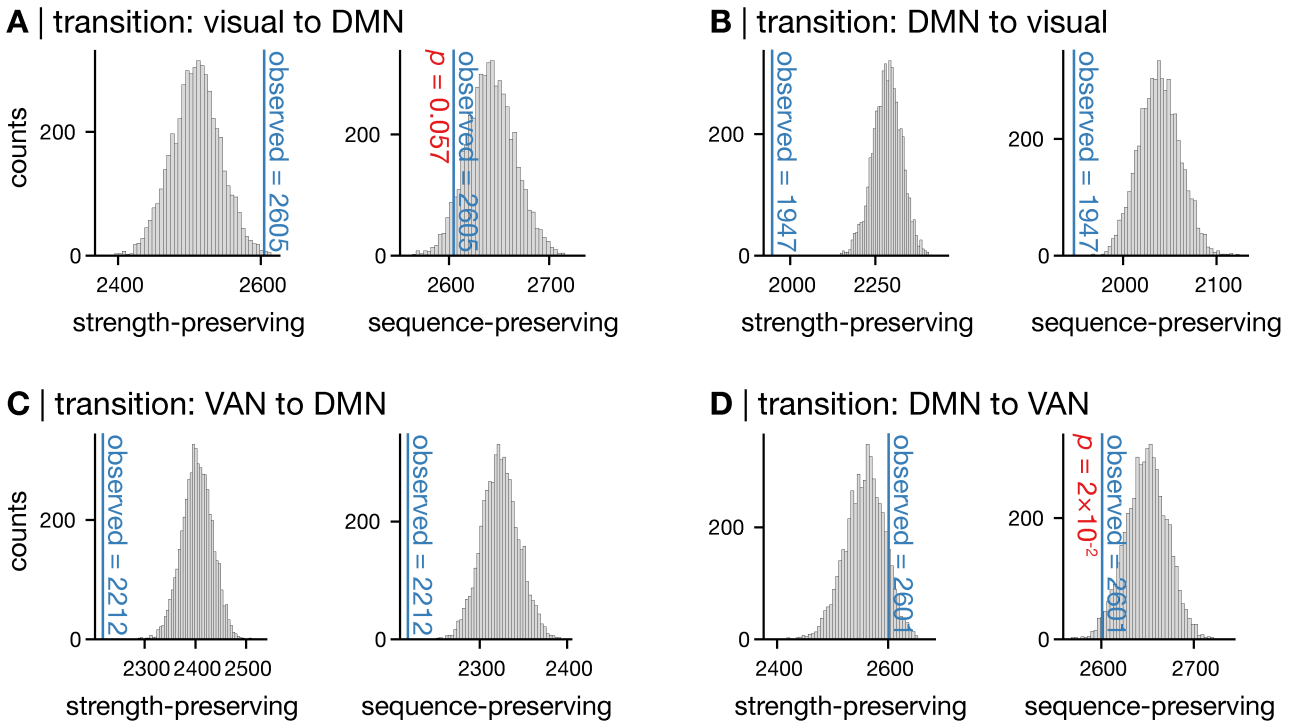

FIG. 9

Null network models uncover the topological properties that are important for control energy. For a given state transition, we recompute *control energy* 5,000 times. Each time we randomly rewire the edges of the adjacency matrix subject to certain constraints. This process generates an empirical null distribution for *control energy*. Here, we generate two null distributions per state transition; one that preserves the spatial embedding of system nodes as well as the strength distribution (strength-preserving), and another that preserves spatial embedding and the strength sequence of the nodes (sequence-preserving). We computed these nulls for the visual-to-DMN transition (**A**), the DMN-to-visual transition (**B**), the VAN-to-DMN transition (**C**), and the DMN-to-VAN transition (**D**). Collectively, these results demonstrate that *control energy* for some transitions is likely driven by strength (e.g., visual-to-DMN and DMN-to-VAN) while others may be driven by higher-order topology (e.g., DMN-to-visual and VAN-to-DMN). See main text for extended discussion.

is consistent with what we observe in the mouse connectome (Figure 7).

Second, when we preserve the strength distribution in our null model, we observe larger-than-expected energy when transitioning from the visual cortex to the DMN (Figure 9A, left). This result appears counter intuitive until we consider the differences in strength between these brain states. In the structural connectome used here, the mean strength of the nodes within the visual state is 67,773, while the mean strength of the nodes in the DMN is 59,455, and the mean strength of the remaining nodes is 53,761. Thus, the connectivity strength between the visual state and the rest of the brain is higher than the connectivity strength between the DMN and the rest of the brain. When we preserve only the strength distribution in the null, this difference in strength is not maintained, which results in relatively high-strength nodes being redistributed throughout the brain. In turn, this redistribution results in reduced *control energy* in the null distribution for the visual-to-DMN transition. This finding suggests that the high strength nodes of the visual system broadcast activity in a way that necessitates high amounts of *control energy* to guide those dynamics toward the DMN. By contrast, when we preserve the strength sequence (Figure 9A, right) we also preserve the between-state difference in strength, which yields a null that is much closer to the observed energy. Together, the results in Figure 9A demonstrate that differences in connectivity strength between brain states drives the energy observed for the visual-to-DMN transition.

Third, the above line of reasoning does not hold when we consider the transition from DMN back to visual cortex (Figure 9B). Here, we observe lower-than-expected energy under both the strength-preserving and the sequence-preserving nulls. This result demonstrates that the aforementioned difference in strength between the visual state and the DMN is not what drives observed energy. In turn, this result suggests that higher-order topological properties of the connectome may support the efficient completion of the DMN-to-visual transition. Finally, Figure 9C and D show that all of the above results and interpretations vary as a function of states.

The above findings—that the spatial embedding of node strength drives energy for one transition direction but not the other—underscores the utility of probing NCT outputs using null network models. That is, through comparing a pair of null network models, we obtained evidence for how certain aspects of network topology (i.e., strength) contribute to

different state transitions.

Null network models can also be applied to *average controllability*:

```
# run permutation
n_perms = 5000  # number of permutations

# container for null distribution
ave_ctrb_null = np.zeros((n_perms, n_nodes))

for perm in tqdm(np.arange(n_perms)):
    # rewire adjacency matrix using geomsurr
    _, _, Wssp = geomsurr(W=adjacency, D=distance_matrix, seed=perm)

    # compute average controllability
    Wssp = matrix_normalization(Wssp, system)
    ave_ctrb_null[perm, :] = ave_control(Wssp, system)
```

The above code will generate an empirical null distribution for *average controllability* at each system node. In turn, this procedure will yield N null distributions for a given null network model, the visualization of which is impractical. As such, researchers may instead assign p-values to the observed *average controllability* values using `get_null_p`:

```
# calculate p-values
p_vals_ssp = np.zeros(n_nodes)

for node in tqdm(np.arange(n_nodes)):
    # version='standard' will calculate the number of times the null is larger than the observed
    value
    # version='reverse' will calculate the number of times the null is smaller than the observed
    value
    p_vals_ssp[node] = get_null_p(
        average_controllability[node], ave_ctrb_null[:, node], version="standard"
    )

p_vals_ssp = get_fdr_p(p_vals_ssp)  # correct p values for multiple comparisons
```

The above code will yield FDR-corrected p-values denoting the proportion of times that a nodes' *average controllability* was larger than expected under the null. As touched upon above, this result only tells half the story, and there may be reasons for *average controllability* to be smaller than expected under the null. As such, we recommend running both `version='standard'` and `version='reverse'` to test both tails of the null distribution.

XI. TIMING

As noted throughout the protocol, the timing of each step is relatively short, often not exceeding 1 second per step. We note two clarifications. First, these time estimates are only for a single execution of each step as shown in the protocol. In reality, these steps will likely need to be executed many times over to achieve researchers' goals. For example, a given study may need to compute control energy for multiple control tasks across multiple subjects, which will increase run time. This time will increase further if null network models are used, wherein each step may be run thousands of times for a single control task. However, in these instances, protocol steps can be trivially parallelized using High Performance Computing (HPC), which will reduce run time. Second, timing will vary as a function of researchers' data processing. For example, in this protocol, we performed analysis on a structural connectome comprising 200 nodes. Increasing parcellation resolution will increase run time.

Acknowledgments:

Funding
National Institute of Mental Health grant K99MH127296 (LP). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.
NARSAD Young Investigator Grant 28995 from the Brain & Behavior Research Foundation (LP)
National Institute of Mental Health grant R21MH106799 (DSB and TDS)
National Institute of Mental Health grant R01MH113550 (DSB and TDS)
National Institute of Mental Health grant RF1MH116920 (DSB and TDS)
Swartz Foundation (DSB)
John D. and Catherine T. MacArthur Foundation (DSB)
National Institute of Mental Health grant R01MH120482 (TDS)
National Institute of Mental Health grant R01MH107703 (TDS)
National Institute of Mental Health grant R01MH112847 (TDS and RTS)
National Institute of Mental Health grant R37MH125829 (TDS)
National Institute of Mental Health grant R01EB022573 (TDS)
National Institute of Mental Health grant R01MH107235 (RCG)
National Institute of Mental Health grant R01MH119219 (RCG and REG)
Penn-CHOP Lifespan Brain Institute
National Science Foundation grant DGE-1321851 (JZK)
National Institute of Mental Health grant RC2MH089983 (Philadelphia Neurodevelopmental Cohort)
National Institute of Mental Health grant RC2MH089924 (Philadelphia Neurodevelopmental Cohort)

Author contributions
Conceptualization: L.P., J.Z.K., T.D.S., and D.S.B.
Methodology: L.P., J.Z.K., J.S., and D.S.B.
Software: L.P., J.Z.K., and J.S.
Formal analysis: L.P., and J.Z.K.
Visualization: L.P., and J.Z.K.
Data curation: J.K.B., M.C., S.C., R.E.G., R.C.G., R.T.S., D.Z., and T.D.S.
Writing—original draft: L.P., and J.Z.K.
Writing—reviewing and editing: L.P., J.Z.K, J.S., J.K.B., M.C., S.C., R.E.G., R.C.G., F.P., R.T.S., D.Z., T.D.S, and D.S.B.

Competing interests
R.T.S. receives consulting compensation from Octave Bioscience and compensation for reviewership duties from the American Medical Association. All other authors declare no competing interests.

Data availability
The PNC data are publicly available in the Database of Genotypes and Phenotypes: accession number: phs00607.v3.p2; https://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/study.cgi?study_id=phs000607.v3.p2

Code availability
All analysis code is freely available at https://github.com/BassettLab/nctpy/

REFERENCES

1. Bassett, D. S. & Sporns, O. Network neuroscience. en. *Nature Neuroscience* **20,** 353–364. ISSN: 1097-6256, 1546-1726. http://www.nature.com/articles/nn.4502 (2022-04-12) (Mar. 2017).
2. Bassett, D. S., Zurn, P. & Gold, J. I. On the nature and use of models in network neuroscience. en. *Nature Reviews Neuroscience* **19,** 566–578. ISSN: 1471-003X, 1471-0048. http://www.nature.com/articles/s41583-018-0038-8 (Sept. 2018).
3. Betzel, R. F. & Bassett, D. S. Multi-scale brain networks. en. *NeuroImage* **160,** 73–83. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811916306152 (Oct. 2017).
4. Fornito, A., Zalesky, A. & Bullmore, E. T. *Fundamentals of Brain Network Analysis* ISBN: 9780124079083 (Academic Press Elsevier, 2016).
5. Menara, T., Katewa, V., Bassett, D. S. & Pasqualetti, F. *The Structured Controllability Radius of Symmetric (Brain) Networks* in *2018 Annual American Control Conference (ACC)* 2018 Annual American Control Conference (ACC) (IEEE, Milwaukee, WI, 2018), 2802–2807. ISBN: 978-1-5386-5428-6. https://ieeexplore.ieee.org/document/8431724/.
6. Pasqualetti, F., Zampieri, S. & Bullo, F. Controllability Metrics, Limitations and Algorithms for Complex Networks. *IEEE Transactions on Control of Network Systems* **1,** 40–52. ISSN: 2325-5870. http://ieeexplore.ieee.org/document/6762966/ (2014).
7. Kim, J. Z. & Bassett, D. S. in *Neural Engineering* (ed He, B.) 497–518 (Springer International Publishing, Cham, 2020). ISBN: 978-3-030-43395-6. https://doi.org/10.1007/978-3-030-43395-6_17.
8. Karrer, T. M. *et al.* A practical guide to methodological considerations in the controllability of structural brain networks. *Journal of Neural Engineering* **17,** 026031. ISSN: 1741-2552. https://iopscience.iop.org/article/10.1088/1741-2552/ab6e8b (2020).
9. Seguin, C., Sporns, O. & Zalesky, A. Brain network communication: concepts, models and applications. en. *Nature Reviews Neuroscience*. ISSN: 1471-003X, 1471-0048. https://www.nature.com/articles/s41583-023-00718-5 (July 2023).
10. Gu, S. *et al.* Controllability of structural brain networks. *Nature Communications* **6,** 8414 (2015).
11. Gu, S. *et al.* Optimal trajectories of brain state transitions. *NeuroImage* **148,** 305–317. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811917300058 (2017).
12. Tang, E. *et al.* Developmental increases in white matter network controllability support a growing diversity of brain dynamics. *Nature Communications* **8,** 1252. ISSN: 2041-1723. http://www.nature.com/articles/s41467-017-01254-4 (2017).
13. Tang, E. *et al.* Control of brain network dynamics across diverse scales of space and time. en. *Physical Review E* **101,** 062301. ISSN: 2470-0045, 2470-0053. https://link.aps.org/doi/10.1103/PhysRevE.101.062301 (June 2020).
14. Stiso, J. *et al.* White Matter Network Architecture Guides Direct Electrical Stimulation through Optimal State Transitions. *Cell Reports* **28,** 2554–2566.e7. ISSN: 22111247. https://linkinghub.elsevier.com/retrieve/pii/S2211124719310411 (2019).
15. Scheid, B. H. *et al.* Time-evolving controllability of effective connectivity networks during seizure progression. *Proceedings of the National Academy of Sciences* **118,** e2006436118. ISSN: 0027-8424, 1091-6490. https://pnas.org/doi/full/10.1073/pnas.2006436118 (2021).
16. Medaglia, J. D. *et al.* Network Controllability in the Inferior Frontal Gyrus Relates to Controlled Language Variability and Susceptibility to TMS. en. *The Journal of Neuroscience* **38,** 6399–6410. ISSN: 0270-6474, 1529-2401. https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.0092-17.2018 (July 2018).
17. Medaglia, J. D. *et al.* Language Tasks and the Network Control Role of the Left Inferior Frontal Gyrus. en. *eneuro* **8,** ENEURO.0382–20.2021. ISSN: 2373-2822. https://www.eneuro.org/lookup/doi/10.1523/ENEURO.0382-20.2021 (Sept. 2021).
18. Muldoon, S. F. *et al.* Stimulation-Based Control of Dynamic Brain Networks. en. *PLOS Computational Biology* **12** (ed Hilgetag, C. C.) e1005076. ISSN: 1553-7358. https://dx.plos.org/10.1371/journal.pcbi.1005076 (Sept. 2016).
19. Cornblath, E. J. *et al.* Sex differences in network controllability as a predictor of executive function in youth. *NeuroImage* **188,** 122–134. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811918321293 (2019).
20. Cornblath, E. J. *et al.* Temporal sequences of brain activity at rest are constrained by white matter structure and modulated by cognitive demands. *Communications Biology* **3,** 261. ISSN: 2399-3642. http://www.nature.com/articles/s42003-020-0961-x (2020).
21. Satterthwaite, T. D. *et al.* Neuroimaging of the Philadelphia Neurodevelopmental Cohort. en. *NeuroImage* **86,** 544–553. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811913008331 (Feb. 2014).

22. Satterthwaite, T. D. *et al.* The Philadelphia Neurodevelopmental Cohort: A publicly available resource for the study of normal and abnormal brain development in youth. en. *NeuroImage* **124,** 1115–1119. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811915002529 (Jan. 2016).
23. Oh, S. W. *et al.* A mesoscale connectome of the mouse brain. *Nature* **508,** 207–214 (2014).
24. Hierarchical organization of cortical and thalamic connectivity. en. *Nature* **575,** 195–202. ISSN: 0028-0836, 1476-4687. https://www.nature.com/articles/s41586-019-1716-z (Nov. 2019).
25. Knox, J. E. *et al.* High-resolution data-driven model of the mouse connectome. en. *Network Neuroscience* **3,** 217–236. ISSN: 2472-1751. https://direct.mit.edu/netn/article/3/1/217-236/2194 (Jan. 2019).
26. Chiêm, B., Crevecoeur, F. & Delvenne, J.-C. Structure-informed functional connectivity driven by identifiable and state-specific control regions. en. *Network Neuroscience* **5,** 591–613. ISSN: 2472-1751. https://direct.mit.edu/netn/article/5/2/591/98351/Structure-informed-functional-connectivity-driven (June 2021).
27. Jeganathan, J. *et al.* Fronto-limbic dysconnectivity leads to impaired brain network controllability in young people with bipolar disorder and those at high genetic risk. en. *NeuroImage: Clinical* **19,** 71–81. ISSN: 22131582. https://linkinghub.elsevier.com/retrieve/pii/S2213158218301025 (2018).
28. Kenett, Y. N. *et al.* Driving the brain towards creativity and intelligence: A network control theory analysis. en. *Neuropsychologia* **118,** 79–90. ISSN: 00283932. https://linkinghub.elsevier.com/retrieve/pii/S0028393218300010 (Sept. 2018).
29. Yuan, J., Ji, S., Luo, L., Lv, J. & Liu, T. Control energy assessment of spatial interactions among ¡span style="font-variant:small-caps;"¿macro-scale¡/span¿ brain networks. en. *Human Brain Mapping* **43,** 2181–2203. ISSN: 1065-9471, 1097-0193. https://onlinelibrary.wiley.com/doi/10.1002/hbm.25780 (May 2022).
30. Singleton, S. P. *et al.* Receptor-informed network control theory links LSD and psilocybin to a flattening of the brain's control energy landscape. en. *Nature Communications* **13,** 5812. ISSN: 2041-1723. https://www.nature.com/articles/s41467-022-33578-1 (Oct. 2022).
31. Singleton, S. P. *et al. Time-resolved network control analysis links reduced control energy under DMT with the serotonin 2a receptor, signal diversity, and subjective experience* en. preprint (Neuroscience, May 2023). http://biorxiv.org/lookup/doi/10.1101/2023.05.11.540409.
32. Luppi, A. I. *et al. Transitions between cognitive topographies: contributions of network structure, neuromodulation, and disease* en. preprint (Neuroscience, Mar. 2023). http://biorxiv.org/lookup/doi/10.1101/2023.03.16.532981.
33. Maxwell, J. C. I. On governors. *Proceedings of the Royal Society of London,* 270–283 (1868).
34. Grasser, F., D'arrigo, A., Colombi, S. & Rufer, A. C. JOE: a mobile, inverted pendulum. *IEEE Transactions on industrial electronics* **49,** 107–114 (2002).
35. Hodgkin, A. L. & Huxley, A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology* **117,** 500 (1952).
36. Papadopoulos, L., Kim, J. Z., Kurths, J. & Bassett, D. S. Development of structural correlations and synchronization from adaptive rewiring in networks of Kuramoto oscillators. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **27,** 073115 (2017).
37. Wilson, H. R. & Cowan, J. D. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical journal* **12,** 1–24 (1972).
38. Schiff, S. J. *et al.* Controlling chaos in the brain. *Nature* **370,** 615–620 (1994).
39. Suárez, L. E., Markello, R. D., Betzel, R. F. & Misic, B. Linking Structure and Function in Macroscale Brain Networks. en. *Trends in Cognitive Sciences* **24,** 302–315. ISSN: 13646613. https://linkinghub.elsevier.com/retrieve/pii/S1364661320300267 (Apr. 2020).
40. Vázquez-Rodríguez, B. *et al.* Gradients of structure–function tethering across neocortex. en. *Proceedings of the National Academy of Sciences* **116,** 21219–21227. ISSN: 0027-8424, 1091-6490. https://pnas.org/doi/full/10.1073/pnas.1903403116 (Oct. 2019).
41. Baum, G. L. *et al.* Development of structure–function coupling in human brain networks during youth. en. *Proceedings of the National Academy of Sciences* **117,** 771–778. ISSN: 0027-8424, 1091-6490. http://www.pnas.org/lookup/doi/10.1073/pnas.1912034117 (Jan. 2020).
42. Preti, M. G. & Van De Ville, D. Decoupling of brain function from structure reveals regional behavioral specialization in humans. en. *Nature Communications* **10,** 4747. ISSN: 2041-1723. http://www.nature.com/articles/s41467-019-12765-7 (Dec. 2019).
43. Luo, N. *et al.* Structural Brain Architectures Match Intrinsic Functional Networks and Vary across Domains: A Study from 15 000+ Individuals. en. *Cerebral Cortex* **30,** 5460–5470. ISSN: 1047-3211, 1460-2199. https://academic.oup.com/cercor/article/30/10/5460/5850539 (Sept. 2020).

44. Seguin, C., Tian, Y. & Zalesky, A. Network communication models improve the behavioral and functional predictive utility of the human structural connectome. en. *Network Neuroscience* **4,** 980–1006. ISSN: 2472-1751. https://direct.mit.edu/netn/article/4/4/980-1006/95847 (Jan. 2020).

45. Betzel, R. F., Faskowitz, J., Mi˘sic, B., Sporns, O. & Seguin, C. Multi-policy models of interregional communication in the human connectome. en, 22 (2022).

46. Fox, P. T. & Friston, K. J. Distributed processing; distributed functions? en. *NeuroImage* **61,** 407–426. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811911014534 (June 2012).

47. Hespanha, J. P. *Linear Systems Theory* ISBN13: 9780691179575 (Princeton Press, Princeton, New Jersey, Feb. 2018).

48. Yan, G. *et al.* Network control principles predict neuron function in the Caenorhabditis elegans connectome. *Nature* **550,** 519–523. ISSN: 0028-0836, 1476-4687. http://www.nature.com/articles/nature24056 (2017).

49. Betzel, R. F., Gu, S., Medaglia, J. D., Pasqualetti, F. & Bassett, D. S. Optimally controlling the human connectome: the role of network topology. *Scientific Reports* **6,** 30770. ISSN: 2045-2322. http://www.nature.com/articles/srep30770 (2016).

50. Kim, J. Z. *et al.* Role of graph architecture in controlling dynamical networks with applications to neural systems. *Nature Physics* **14,** 91–98. ISSN: 1745-2473, 1745-2481. http://www.nature.com/articles/nphys4268 (2018).

51. Parkes, L. *et al.* Asymmetric signaling across the hierarchy of cytoarchitecture within the human connectome. en. *Science Advances* **8,** eadd2185. ISSN: 2375-2548. https://www.science.org/doi/10.1126/sciadv.add2185 (Dec. 2022).

52. Sejnowski, T. J., Churchland, P. S. & Movshon, J. A. Putting big data to good use in neuroscience. en. *Nature Neuroscience* **17,** 1440–1441. ISSN: 1097-6256, 1546-1726. http://www.nature.com/articles/nn.3839 (Nov. 2014).

53. Tong, F. Primary visual cortex and visual awareness. en. *Nature Reviews Neuroscience* **4,** 219–229. ISSN: 1471-003X, 1471-0048. http://www.nature.com/articles/nrn1055 (Mar. 2003).

54. Gordon, E. M. *et al.* A somato-cognitive action network alternates with effector regions in motor cortex. en. *Nature* **617,** 351–359. ISSN: 0028-0836, 1476-4687. https://www.nature.com/articles/s41586-023-05964-2 (2023) (May 2023).

55. Bertolero, M. A., Yeo, B. T. T. & D'Esposito, M. The modular and integrative functional architecture of the human brain. en. *Proceedings of the National Academy of Sciences* **112.** ISSN: 0027-8424, 1091-6490. https://pnas.org/doi/full/10.1073/pnas.1510619112 (Dec. 2015).

56. Bertolero, M. A., Yeo, B. T. T., Bassett, D. S. & D'Esposito, M. A mechanistic model of connector hubs, modularity and cognition. en. *Nature Human Behaviour* **2,** 765–777. ISSN: 2397-3374. http://www.nature.com/articles/s41562-018-0420-6 (Oct. 2018).

57. Van den Heuvel, M. P. & Sporns, O. Network hubs in the human brain. en. *Trends in Cognitive Sciences* **17,** 683–696. ISSN: 13646613. https://linkinghub.elsevier.com/retrieve/pii/S1364661313002167 (Dec. 2013).

58. Van den Heuvel, M. P. & Sporns, O. Rich-Club Organization of the Human Connectome. en. *Journal of Neuroscience* **31,** 15775–15786. ISSN: 0270-6474, 1529-2401. https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.3539-11.2011 (Nov. 2011).

59. Fornito, A., Zalesky, A. & Breakspear, M. The connectomics of brain disorders. en. *Nature Reviews Neuroscience* **16,** 159–172. ISSN: 1471-003X, 1471-0048. http://www.nature.com/articles/nrn3901 (Mar. 2015).

60. Crossley, N. A. *et al.* The hubs of the human connectome are generally implicated in the anatomy of brain disorders. en. *Brain* **137,** 2382–2395. ISSN: 1460-2156, 0006-8950. https://academic.oup.com/brain/article-lookup/doi/10.1093/brain/awu132 (Aug. 2014).

61. Van Essen, D. C. *et al.* The WU-Minn Human Connectome Project: An overview. en. *NeuroImage* **80,** 62–79. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811913005351 (Oct. 2013).

62. Casey, B. *et al.* The Adolescent Brain Cognitive Development (ABCD) study: Imaging acquisition across 21 sites. en. *Developmental Cognitive Neuroscience* **32,** 43–54. ISSN: 18789293. https://linkinghub.elsevier.com/retrieve/pii/S1878929317301214 (Aug. 2018).

63. Alexander, L. M. *et al.* An open resource for transdiagnostic research in pediatric mental health and learning disorders. en. *Scientific Data* **4,** 170181. ISSN: 2052-4463. http://www.nature.com/articles/sdata2017181 (Dec. 2017).

64. Amunts, K. *et al.* BigBrain: An Ultrahigh-Resolution 3D Human Brain Model. en. *Science* **340,** 1472–1475. ISSN: 0036-8075, 1095-9203. https://www.science.org/doi/10.1126/science.1235381 (June 2013).

65. Hawrylycz, M. J. *et al.* An anatomically comprehensive atlas of the adult human brain transcriptome. en. *Nature* **489,** 391–399. ISSN: 0028-0836, 1476-4687. http://www.nature.com/articles/nature11405 (Sept. 2012).

66. Thomas Yeo, B. T. *et al.* The organization of the human cerebral cortex estimated by intrinsic functional connectivity. en. *Journal of Neurophysiology* **106,** 1125–1165. ISSN: 0022-3077, 1522-1598. https://www.physiology.org/doi/10.1152/jn.00338.2011 (Sept. 2011).

67. De Reus, M. A. & van den Heuvel, M. P. Simulated rich club lesioning in brain networks: a scaffold for communication and integration? *Frontiers in Human Neuroscience* **8.** ISSN: 1662-5161. http://journal.frontiersin.org/article/10.3389/fnhum.2014.00647/abstract (Aug. 2014).

68. Van den Heuvel, M. P. & Sporns, O. An Anatomical Substrate for Integration among Functional Networks in Human Cortex. en. *Journal of Neuroscience* **33,** 14489–14500. ISSN: 0270-6474, 1529-2401. https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.2128-13.2013 (Sept. 2013).

69. Parkes, L. *et al.* Network Controllability in Transmodal Cortex Predicts Psychosis Spectrum Symptoms. *Biological Psychiatry* **89,** S370–S371 (2021).

70. Margulies, D. S. *et al.* Situating the default-mode network along a principal gradient of macroscale cortical organization. en. *Proceedings of the National Academy of Sciences* **113,** 12574–12579. ISSN: 0027-8424, 1091-6490. http://www.pnas.org/lookup/doi/10.1073/pnas.1608282113 (Nov. 2016).

71. Cui, Z. *et al.* Optimization of energy state transition trajectory supports the development of executive function during youth. *eLife* **9,** e53060. ISSN: 2050-084X. https://elifesciences.org/articles/53060 (2020).

72. Niendam, T. A. *et al.* Meta-analytic evidence for a superordinate cognitive control network subserving diverse executive functions. en. *Cognitive, Affective, & Behavioral Neuroscience* **12,** 241–268. ISSN: 1530-7026, 1531-135X. http://link.springer.com/10.3758/s13415-011-0083-5 (June 2012).

73. Braun, U. *et al.* Brain network dynamics during working memory are modulated by dopamine and diminished in schizophrenia. *Nature Communications* **12,** 3478. ISSN: 2041-1723. http://www.nature.com/articles/s41467-021-23694-9 (2021).

74. Parkes, L., Fulcher, B. D., Yücel, M. & Fornito, A. Transcriptional signatures of connectomic subregions of the human striatum. en. *Genes, Brain and Behavior* **16,** 647–663. ISSN: 16011848. https://onlinelibrary.wiley.com/doi/10.1111/gbb.12386 (Sept. 2017).

75. Fulcher, B. D., Murray, J. D., Zerbi, V. & Wang, X.-J. Multimodal gradients across mouse cortex. en. *Proceedings of the National Academy of Sciences* **116,** 4689–4695. ISSN: 0027-8424, 1091-6490. http://www.pnas.org/lookup/doi/10.1073/pnas.1814144116 (Mar. 2019).

76. Fulcher, B. D. & Fornito, A. A transcriptional signature of hub connectivity in the mouse connectome. *Proceedings of the National Academy of Sciences* **113,** 1435–1440. ISSN: 0027-8424, 1091-6490. https://pnas.org/doi/full/10.1073/pnas.1513302113 (2016).

77. Larivière, S. *et al.* Microstructure-Informed Connectomics: Enriching Large-Scale Descriptions of Healthy and Diseased Brains. en. *Brain Connectivity* **9,** 113–127. ISSN: 2158-0014, 2158-0022. https://www.liebertpub.com/doi/10.1089/brain.2018.0587 (Mar. 2019).

78. Arnatkeviciūtė, A., Fulcher, B. D. & Fornito, A. A practical guide to linking brain-wide gene expression and neuroimaging data. en. *NeuroImage* **189,** 353–367. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811919300114 (Apr. 2019).

79. Arnatkeviciute, A. *et al.* Genetic influences on hub connectivity of the human connectome. en. *Nature Communications* **12,** 4237. ISSN: 2041-1723. http://www.nature.com/articles/s41467-021-24306-2 (Dec. 2021).

80. Arnatkeviciūtė, A., Fulcher, B. D., Pocock, R. & Fornito, A. Hub connectivity, neuronal diversity, and gene expression in the Caenorhabditis elegans connectome. en. *PLOS Computational Biology* **14** (ed Inman, C.) e1005989. ISSN: 1553-7358. https://dx.plos.org/10.1371/journal.pcbi.1005989 (Feb. 2018).

81. Anderson, K. M. *et al.* Gene expression links functional networks across cortex and striatum. en. *Nature Communications* **9,** 1428. ISSN: 2041-1723. http://www.nature.com/articles/s41467-018-03811-x (Dec. 2018).

82. Anderson, K. M. *et al.* Convergent molecular, cellular, and cortical neuroimaging signatures of major depressive disorder. en. *Proceedings of the National Academy of Sciences* **117,** 25138–25149. ISSN: 0027-8424, 1091-6490. https://pnas.org/doi/full/10.1073/pnas.2008004117 (Oct. 2020).

83. Paquola, C. *et al.* Microstructural and functional gradients are increasingly dissociated in transmodal cortices. en. *PLOS Biology* **17** (ed Kennedy, H.) e3000284. ISSN: 1545-7885. https://dx.plos.org/10.1371/journal.pbio.3000284 (May 2019).

84. García-Cabezas, M. Á., Zikopoulos, B. & Barbas, H. The Structural Model: a theory linking connections, plasticity, pathology, development and evolution of the cerebral cortex. *Brain Structure and Function* **224,** 985–1008. ISSN: 1863-2653, 1863-2661. http://link.springer.com/10.1007/s00429-019-01841-9 (2019).

85. Yarkoni, T., Poldrack, R. A., Nichols, T. E., Van Essen, D. C. & Wager, T. D. Large-scale automated synthesis of human functional neuroimaging data. en. *Nature Methods* **8,** 665–670. ISSN: 1548-7091, 1548-7105. https://www.nature.com/articles/nmeth.1635 (Aug. 2011).

86. Propagation of electrical signals along giant nerve fibres. en. *Proceedings of the Royal Society of London. Series B - Biological Sciences* **140,** 177–183. ISSN: 2053-9193. https://royalsocietypublishing.org/doi/10.1098/rspb.1952.0054 (Oct. 1952).

87. Breakspear, M. Dynamic models of large-scale brain activity. en. *Nature Neuroscience* **20,** 340–352. ISSN: 1097-6256, 1546-1726. http://www.nature.com/articles/nn.4497 (Mar. 2017).

88. Shine, J. M. *et al.* Computational models link cellular mechanisms of neuromodulation to large-scale neural dynamics. en. *Nature Neuroscience* **24,** 765–776. ISSN: 1097-6256, 1546-1726. http://www.nature.com/articles/s41593-021-00824-6 (June 2021).

89. Papadopoulos, L., Kim, J. Z., Kurths, J. & Bassett, D. S. Development of structural correlations and synchronization from adaptive rewiring in networks of Kuramoto oscillators. en. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **27,** 073115. ISSN: 1054-1500, 1089-7682. http://aip.scitation.org/doi/10.1063/1.4994819 (July 2017).

90. Lu, Z. & Bassett, D. S. Invertible generalized synchronization: A putative mechanism for implicit learning in neural systems. en. *Chaos: An Interdisciplinary Journal of Nonlinear Science* **30,** 063133. ISSN: 1054-1500, 1089-7682. http://aip.scitation.org/doi/10.1063/5.0004344 (June 2020).

91. Suárez, L. E., Richards, B. A., Lajoie, G. & Misic, B. Learning function from structure in neuromorphic networks. en. *Nature Machine Intelligence.* ISSN: 2522-5839. https://www.nature.com/articles/s42256-021-00376-1 (Aug. 2021).

92. Roberts, J. A. *et al.* Metastable brain waves. en. *Nature Communications* **10,** 1056. ISSN: 2041-1723. http://www.nature.com/articles/s41467-019-08999-0 (Dec. 2019).

93. Demirtaş, M. *et al.* Hierarchical Heterogeneity across Human Cortex Shapes Large-Scale Neural Dynamics. en. *Neuron* **101,** 1181–1194.e13. ISSN: 08966273. https://linkinghub.elsevier.com/retrieve/pii/S0896627319300443 (Mar. 2019).

94. Deco, G. *et al.* Resting-State Functional Connectivity Emerges from Structurally and Dynamically Shaped Slow Linear Fluctuations. en. *Journal of Neuroscience* **33,** 11239–11252. ISSN: 0270-6474, 1529-2401. http://www.jneurosci.org/cgi/doi/10.1523/JNEUROSCI.1091-13.2013 (July 2013).

95. Deco, G. *et al.* Dynamical consequences of regional heterogeneity in the brain's transcriptional landscape. en. *Science Advances* **7,** eabf4752. ISSN: 2375-2548. https://www.science.org/doi/10.1126/sciadv.abf4752 (July 2021).

96. Monasson, R. & Rosay, S. Transitions between spatial attractors in place-cell models. *Physical review letters* **115,** 098101 (2015).

97. Mante, V., Sussillo, D., Shenoy, K. V. & Newsome, W. T. Context-dependent computation by recurrent dynamics in prefrontal cortex. *nature* **503,** 78–84 (2013).

98. Fornito, A., Zalesky, A. & Breakspear, M. Graph analysis of the human connectome: Promise, progress, and pitfalls. en. *NeuroImage* **80,** 426–444. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811913004345 (Oct. 2013).

99. Bertolero, M. A. & Bassett, D. S. On the Nature of Explanations Offered by Network Science: A Perspective From and for Practicing Neuroscientists. en. *Topics in Cognitive Science* **12,** 1272–1293. ISSN: 1756-8757, 1756-8765. https://onlinelibrary.wiley.com/doi/10.1111/tops.12504 (Oct. 2020).

100. Vázquez-Rodríguez, B., Liu, Z.-Q., Hagmann, P. & Misic, B. Signal propagation via cortical hierarchies. en. *Network Neuroscience* **4,** 1072–1090. ISSN: 2472-1751. https://direct.mit.edu/netn/article/4/4/1072-1090/95842 (Jan. 2020).

101. Bazinet, V., Vos de Wael, R., Hagmann, P., Bernhardt, B. C. & Misic, B. Multiscale communication in cortico-cortical networks. en. *NeuroImage* **243,** 118546. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811921008193 (Nov. 2021).

102. Fornito, A., Zalesky, A., Pantelis, C. & Bullmore, E. T. Schizophrenia, neuroimaging and connectomics. en. *NeuroImage* **62,** 2296–2314. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811912002133 (Oct. 2012).

103. Bassett, D. S. *et al.* Hierarchical Organization of Human Cortical Networks in Health and Schizophrenia. en. *Journal of Neuroscience* **28,** 9239–9248. ISSN: 0270-6474, 1529-2401. https://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.1929-08.2008 (Sept. 2008).

104. Bassett, D. S., Nelson, B. G., Mueller, B. A., Camchong, J. & Lim, K. O. Altered resting state complexity in schizophrenia. en. *NeuroImage* **59,** 2196–2207. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811911011633 (Feb. 2012).

105. Seguin, C., Razi, A. & Zalesky, A. Inferring neural signalling directionality from undirected structural connectomes. en. *Nature Communications* **10,** 4289. ISSN: 2041-1723. http://www.nature.com/articles/s41467-019-12201-w (Dec. 2019).

106. Seguin, C., Mansour L, S., Sporns, O., Zalesky, A. & Calamante, F. Network communication models narrow the gap between the modular organization of structural and functional brain networks. en. *NeuroImage* **257,** 119323. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811922004426 (2023) (Aug. 2022).

107. Mišić, B. *et al.* Cooperative and Competitive Spreading Dynamics on the Human Connectome. en. *Neuron* **86,** 1518–1529. ISSN: 08966273. https://linkinghub.elsevier.com/retrieve/pii/S0896627315004742 (June 2015).

108. Oldham, S. *et al.* The efficacy of different preprocessing steps in reducing motion-related confounds in diffusion MRI connectomics. *NeuroImage* **222,** 117252. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811920307382 (2020).

109. De Reus, M. A. & van den Heuvel, M. P. Estimating false positives and negatives in brain networks. *NeuroImage* **70,** 402–409. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811913000050 (2013).

110. Yendiki, A., Koldewyn, K., Kakunoori, S., Kanwisher, N. & Fischl, B. Spurious group differences due to head motion in a diffusion MRI study. en. *NeuroImage* **88,** 79–90. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811913011312 (Mar. 2014).

111. Baum, G. L. *et al.* The impact of in-scanner head motion on structural connectivity derived from diffusion MRI. en. *NeuroImage* **173,** 275–286. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811918301381 (June 2018).

112. Cieslak, M. *et al.* QSIPrep: an integrative platform for preprocessing and reconstructing diffusion MRI data. en. *Nature Methods* **18,** 775–778. ISSN: 1548-7091, 1548-7105. http://www.nature.com/articles/s41592-021-01185-5 (July 2021).

113. Roalf, D. R. *et al.* The impact of quality assurance assessment on diffusion tensor imaging outcomes in a large-scale population-based cohort. en. *NeuroImage* **125,** 903–919. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811915009854 (Jan. 2016).

114. Bargmann, C. I. & Marder, E. From the connectome to brain function. en. *Nature Methods* **10,** 483–490. ISSN: 1548-7091, 1548-7105. http://www.nature.com/articles/nmeth.2451 (June 2013).

115. Schaefer, A. *et al.* Local-Global Parcellation of the Human Cerebral Cortex from Intrinsic Functional Connectivity MRI. en. *Cerebral Cortex* **28,** 3095–3114. ISSN: 1047-3211, 1460-2199. https://academic.oup.com/cercor/article/28/9/3095/3978804 (Sept. 2018).

116. Sarwar, T., Ramamohanarao, K. & Zalesky, A. Mapping connectomes with diffusion MRI: deterministic or probabilistic tractography? en. *Magnetic Resonance in Medicine* **81,** 1368–1384. ISSN: 0740-3194, 1522-2594. https://onlinelibrary.wiley.com/doi/10.1002/mrm.27471 (Feb. 2019).

117. Robinson, P. *et al.* Eigenmodes of brain activity: Neural field theory predictions and comparison with experiment. en. *NeuroImage* **142,** 79–98. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811916300908 (Nov. 2016).

118. Deco, G., Jirsa, V. K., Robinson, P. A., Breakspear, M. & Friston, K. The Dynamic Brain: From Spiking Neurons to Neural Masses and Cortical Fields. en. *PLoS Computational Biology* **4** (ed Sporns, O.) e1000092. ISSN: 1553-7358. https://dx.plos.org/10.1371/journal.pcbi.1000092 (Aug. 2008).

119. Shenoy, K. V. & Kao, J. C. Measurement, manipulation and modeling of brain-wide neural population dynamics. en. *Nature Communications* **12,** 633. ISSN: 2041-1723. http://www.nature.com/articles/s41467-020-20371-1 (Dec. 2021).

120. Nozari, E. *et al. Is the brain macroscopically linear? A system identification of resting state dynamics* 2020.

121. He, X. *et al.* Uncovering the biological basis of control energy: Structural and metabolic correlates of energy inefficiency in temporal lobe epilepsy. en. *Science Advances* **8,** eabn2293. ISSN: 2375-2548. https://www.science.org/doi/10.1126/sciadv.abn2293 (2023) (Nov. 2022).

122. McCormick, D. A., Shu, Y. & Yu, Y. Hodgkin and Huxley model—still standing? *Nature* **445,** E1–E2 (2007).

123. Durstewitz, D., Seamans, J. K. & Sejnowski, T. J. Neurocomputational models of working memory. *Nature neuroscience* **3,** 1184–1191 (2000).

124. Hu, S. *et al.* Associative memory realized by a reconfigurable memristive Hopfield neural network. *Nature communications* **6,** 1–8 (2015).

125. Vaidyanathan, S., Volos, C., *et al. Advances and applications in nonlinear control systems* (Springer, 2016).

126. Kumar, E. V. & Jerome, J. Robust LQR controller design for stabilizing and trajectory tracking of inverted pendulum. *Procedia Engineering* **64,** 169–178 (2013).

127. Bastos, A. M. & Schoffelen, J.-M. A tutorial review of functional connectivity analysis methods and their interpretational pitfalls. *Frontiers in systems neuroscience* **9,** 175 (2016).

128. Schwemmer, M. A. & Lewis, T. J. in *Phase response curves in neuroscience* 3–31 (Springer, 2012).

129. Park, Y. & Ermentrout, B. Weakly coupled oscillators in a slowly varying world. *Journal of computational neuroscience* **40,** 269–281 (2016).

130. Brunton, S. L., Brunton, B. W., Proctor, J. L. & Kutz, J. N. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PloS one* **11,** e0150171 (2016).

131. Proctor, J. L., Brunton, S. L. & Kutz, J. N. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems* **15,** 142–161 (2016).

132. Zañudo, J. G. T., Yang, G. & Albert, R. Structure-based control of complex networks with nonlinear dynamics. *Proceedings of the National Academy of Sciences* **114,** 7234–7239 (2017).

133. Haynes, G. & Hermes, H. Nonlinear controllability via Lie theory. *SIAM Journal on Control* **8,** 450–460 (1970).

134. Towlson, E. K. *et al. Caenorhabditis elegans* and the network control framework—FAQs. en. *Philosophical Transactions of the Royal Society B: Biological Sciences* **373,** 20170372. ISSN: 0962-8436, 1471-2970. https://royalsocietypublishing.org/doi/10.1098/rstb.2017.0372 (Oct. 2018).

135. Felleman, D. J. & Van Essen, D. C. Distributed Hierarchical Processing in the Primate Cerebral Cortex. *Cerebral Cortex* **1,** 1–47. ISSN: 1047-3211, 1460-2199. https://academic.oup.com/cercor/article-lookup/doi/10.1093/cercor/1.1.1 (1991).

136. Stiso, J. *et al.* Learning in brain-computer interface control evidenced by joint decomposition of brain and behavior. *Journal of Neural Engineering* **17,** 046018. ISSN: 1741-2552. https://iopscience.iop.org/article/10.1088/1741-2552/ab9064 (July 2020).

137. Szymula, K. P., Pasqualetti, F., Graybiel, A. M., Desrochers, T. M. & Bassett, D. S. Habit learning supported by efficiently controlled network dynamics in naive macaque monkeys. *arXiv.* https://arxiv.org/abs/2006.14565 (2020).

138. Rosen, A. F. *et al.* Quantitative assessment of structural image quality. en. *NeuroImage* **169,** 407–418. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811917310832 (Apr. 2018).

139. Ciric, R. *et al.* Mitigating head motion artifact in functional connectivity MRI. en. *Nature Protocols* **13,** 2801–2826. ISSN: 1754-2189, 1750-2799. http://www.nature.com/articles/s41596-018-0065-y (Dec. 2018).

140. Gorgolewski, K. *et al.* Nipype: A Flexible, Lightweight and Extensible Neuroimaging Data Processing Framework in Python. *Frontiers in Neuroinformatics* **5.** ISSN: 1662-5196. http://journal.frontiersin.org/article/10.3389/fninf.2011.00013/abstract (2011).

141. Satterthwaite, T. D. *et al.* An improved framework for confound regression and filtering for control of motion artifact in the preprocessing of resting-state functional connectivity data. en. *NeuroImage* **64,** 240–256. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S1053811912008609 (Jan. 2013).

142. Esteban, O. *et al.* fMRIPrep: a robust preprocessing pipeline for functional MRI. en. *Nature Methods* **16,** 111–116. ISSN: 1548-7091, 1548-7105. http://www.nature.com/articles/s41592-018-0235-4 (Jan. 2019).

143. Harris, J. A. *et al.* Hierarchical organization of cortical and thalamic connectivity. *Nature* **575,** 195–202 (2019).

144. Knox, J. E. *et al.* High-resolution data-driven model of the mouse connectome. *Network Neuroscience* **3,** 217–236 (2018).

145. Mahadevan, A. S. *et al. Alprazolam modulates persistence energy during emotion processing in first-degree relatives of individuals with schizophrenia: a network control study* en. preprint (bioRxiv, Apr. 2021). http://biorxiv.org/lookup/doi/10.1101/2021.04.22.440935.

146. Markello, R. D. *et al.* neuromaps: structural and functional interpretation of brain maps. en. *Nature Methods.* ISSN: 1548-7091, 1548-7105. https://www.nature.com/articles/s41592-022-01625-w (Oct. 2022).

147. Hansen, J. Y. *et al.* Mapping neurotransmitter systems to the structural and functional organization of the human neocortex. en. *Nature Neuroscience.* ISSN: 1097-6256, 1546-1726. https://www.nature.com/articles/s41593-022-01186-3 (Oct. 2022).

148. Regional, Layer, and Cell-Type-Specific Connectivity of the Mouse Default Mode Network. en. *Neuron* **109,** 545–559.e8. ISSN: 08966273. https://linkinghub.elsevier.com/retrieve/pii/S0896627320308898 (Feb. 2021).

149. Váša, F. & Mišić, B. Null models in network neuroscience. en. *Nature Reviews Neuroscience.* ISSN: 1471-003X, 1471-0048. https://www.nature.com/articles/s41583-022-00601-9 (May 2022).

150. Roberts, J. A. *et al.* The contribution of geometry to the human connectome. *NeuroImage* **124,** 379–393. ISSN: 10538119. https://linkinghub.elsevier.com/retrieve/pii/S105381191500806X (2016).

1880

XII. EXTENDED DATA

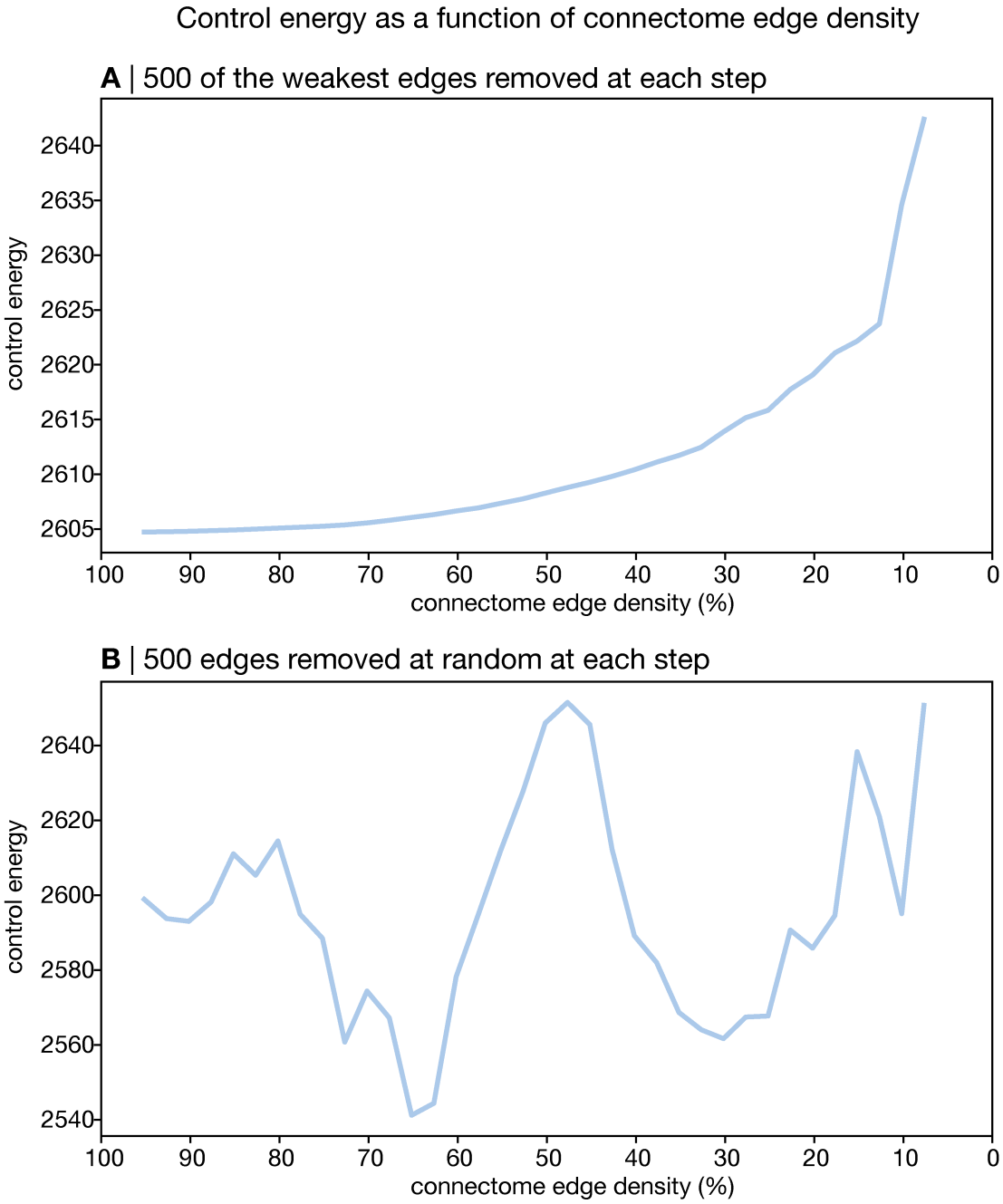

FIG. S1

Control energy as a function of connectome edge density. In the undirected human connectome, we iteratively set 500 of the weakest edges (**A**) or 500 random edges (**B**) to 0, stopping once edge density fell below 10%. *Control energy* was recomputed at each iteration and is shown here. This plot illustrates the fact that energy varies as a function of edge density and is primarily driven by the strongest edges in the network.

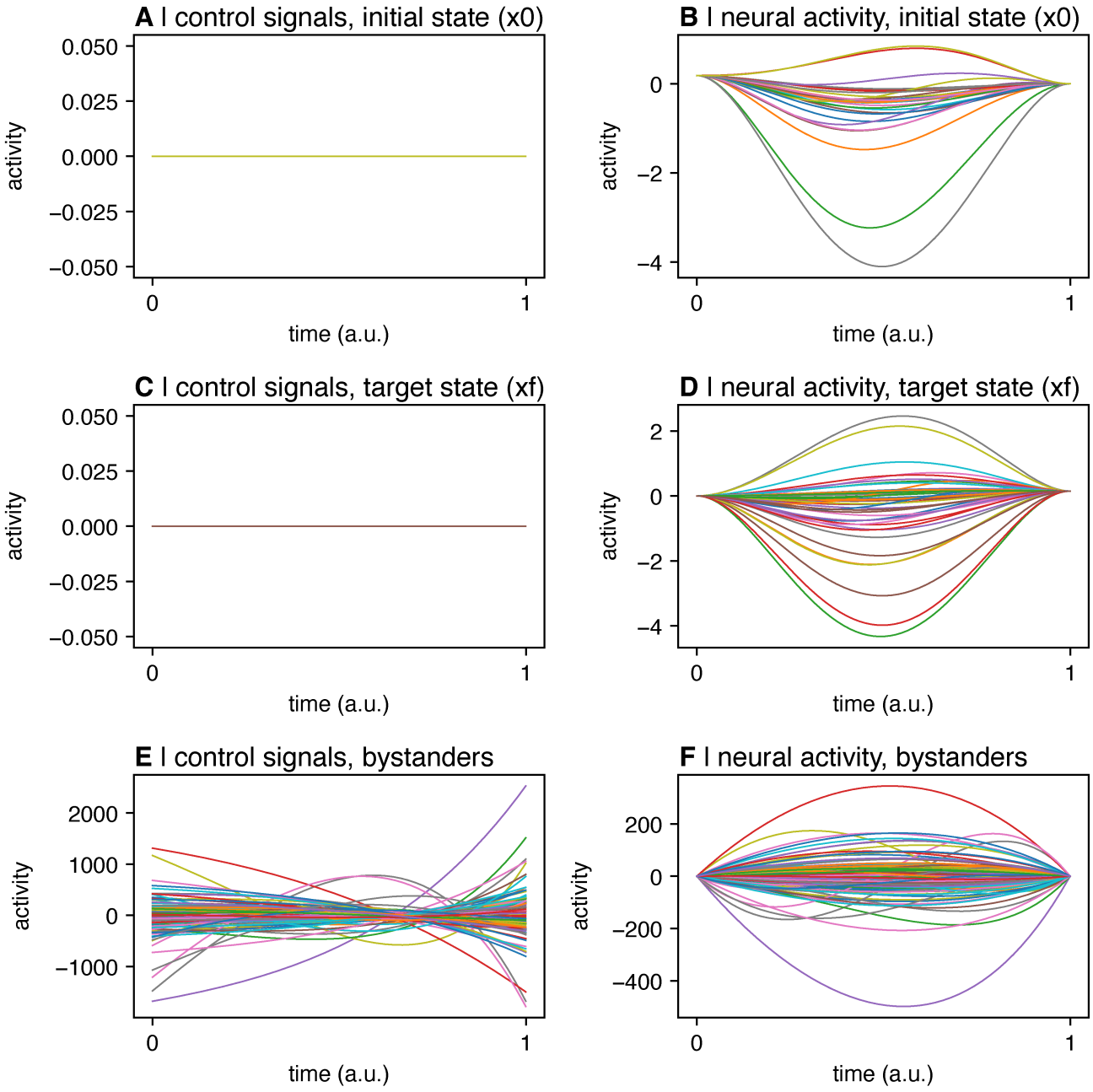

FIG. S2
Control signals and state trajectory. *Only bystanders are set as control nodes*. T=1. Initial state = visual system. Target state = default mode system. Inversion error = 2.09×10^{-10}. Reconstruction error = 1.75×10^{-8}. Energy = 6.64×10^{9}.

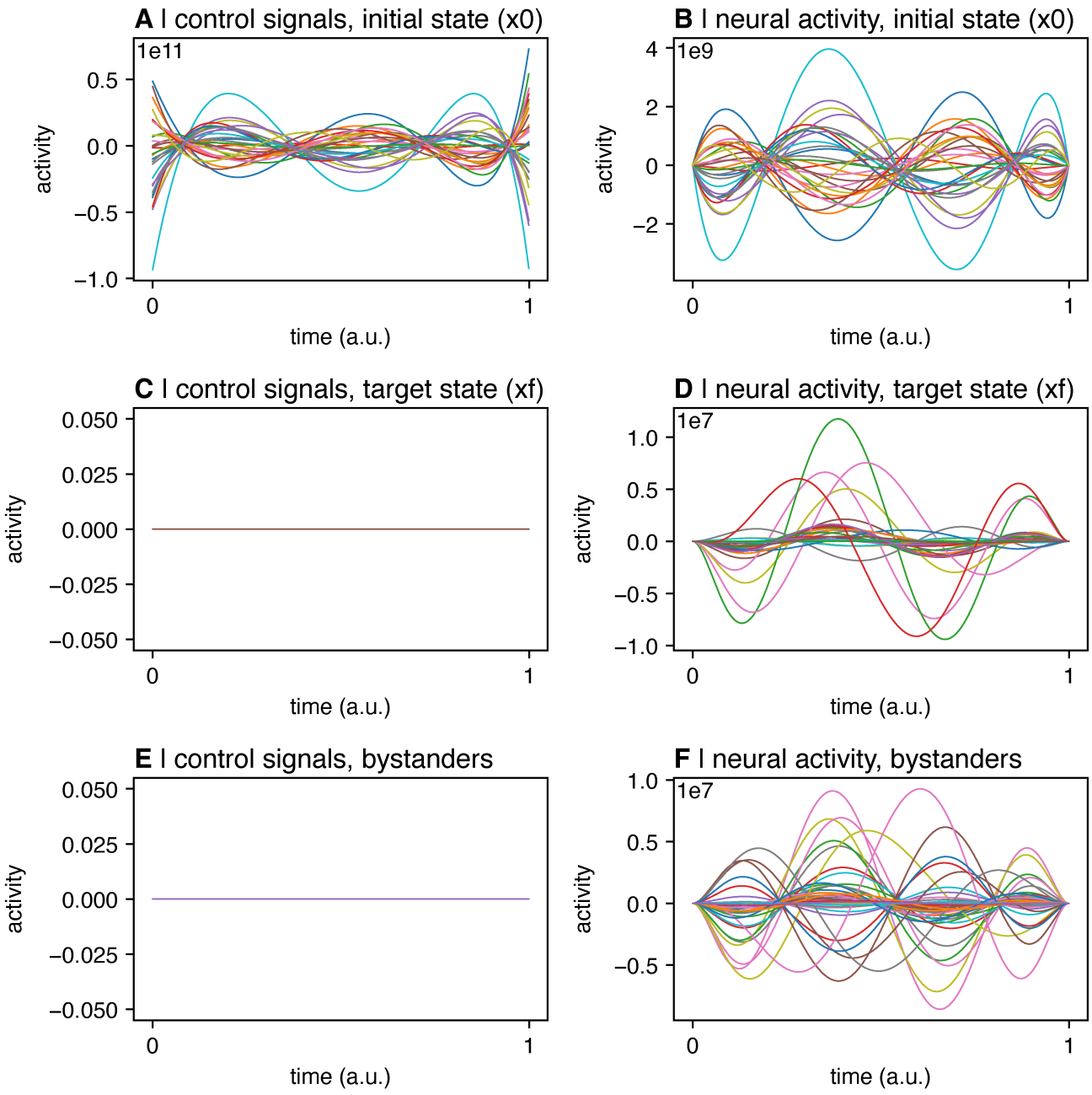

FIG. S3
Control signals and state trajectory. *Only nodes in the initial state are set as control nodes*. T=1. Initial state = visual system. Target state = default mode system. Inversion error = 1.37×10^3. Reconstruction error = 2.04×10^5. Energy = 3.68×10^{24}. Note that this state transition does not complete successfully.

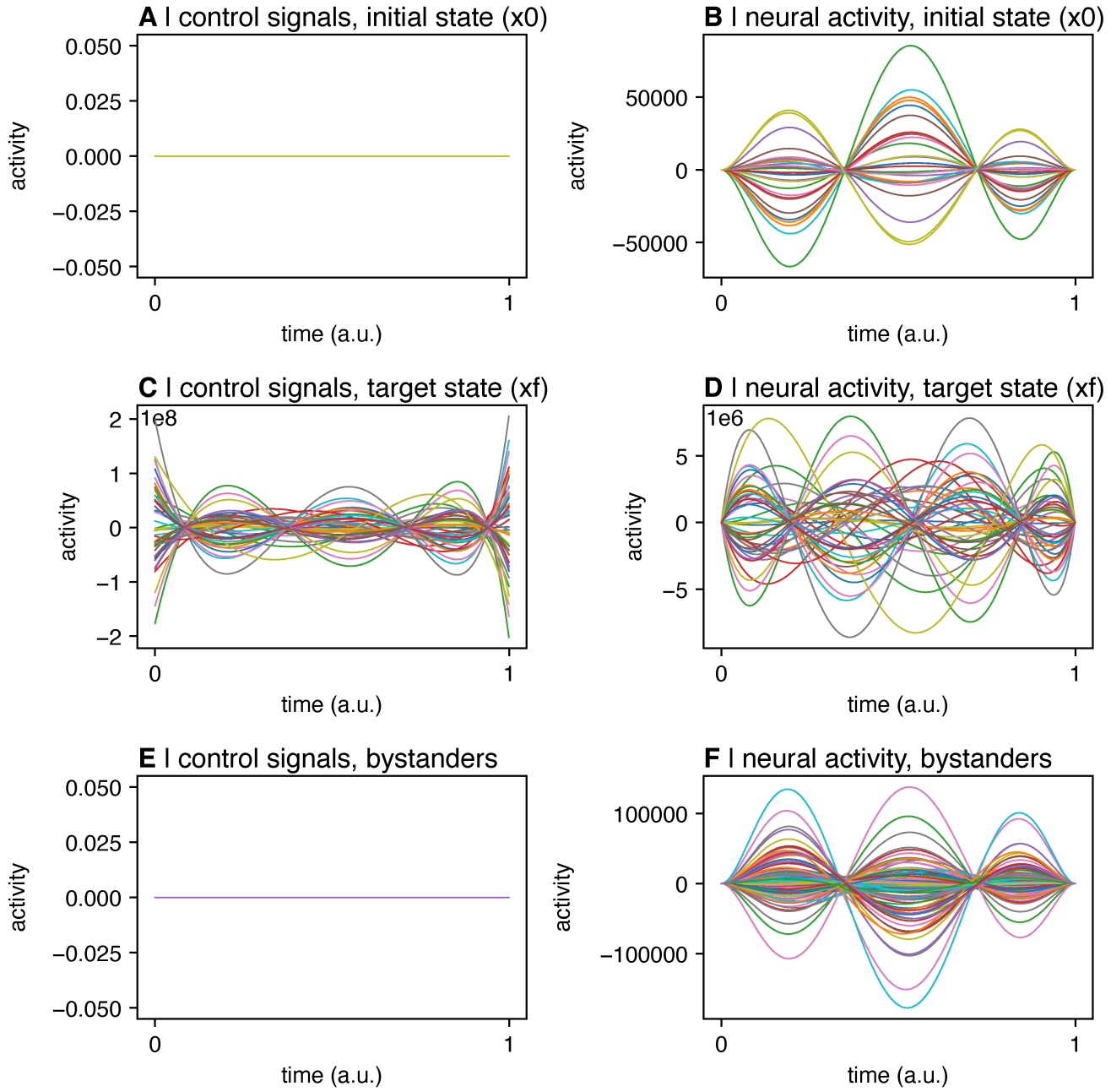

FIG. S4
Control signals and state trajectory. *Only nodes in the target state are set as control nodes*. T=1. Initial state = visual system. Target state = default mode system. Inversion error = 1.50. Reconstruction error = 1.12×10^2. Energy = 3.38×10^{19}. Note that this state transition does not complete successfully.

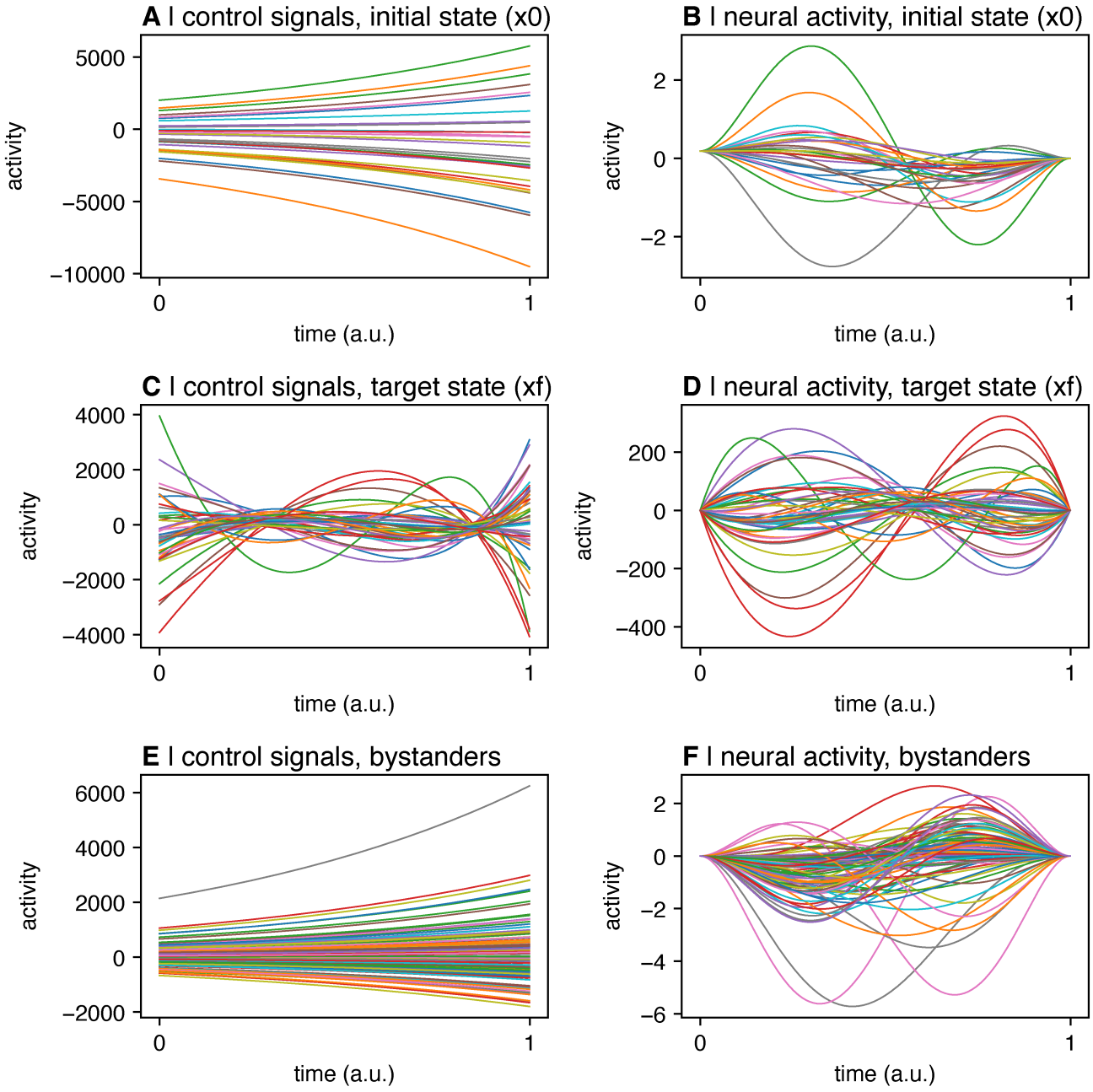

FIG. S5

Control signals and state trajectory. *Nodes in the target state are set as control nodes with control weights of 1, whereas the remaining nodes are given a small amount of control* (1×10^{-5}). T=1. Initial state = visual system. Target state = default mode system. Inversion error = 2.00×10^{-8}. Reconstruction error = 1.16×10^{-6}. Energy = 2.31×10^{11}. Unlike the scenario depicted in Figure S4, this state transition completes successfully.

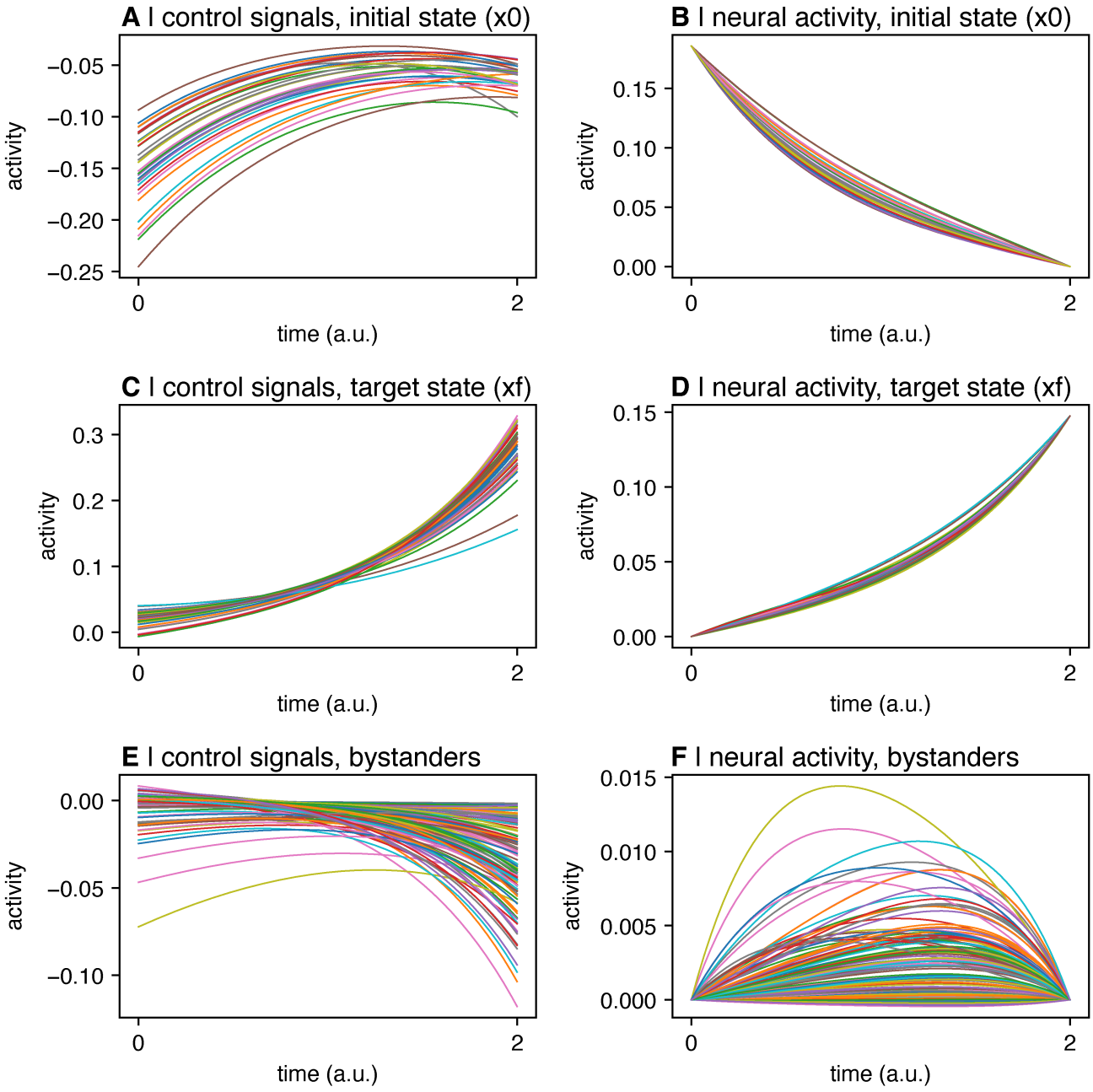

FIG. S6
Control signals and state trajectory. Uniform full control set. T=2. Initial state = visual system. Target state = default mode system. Inversion error = 3.24×10^{-15}. Reconstruction error = 1.55×10^{-13}. Energy = 1904.

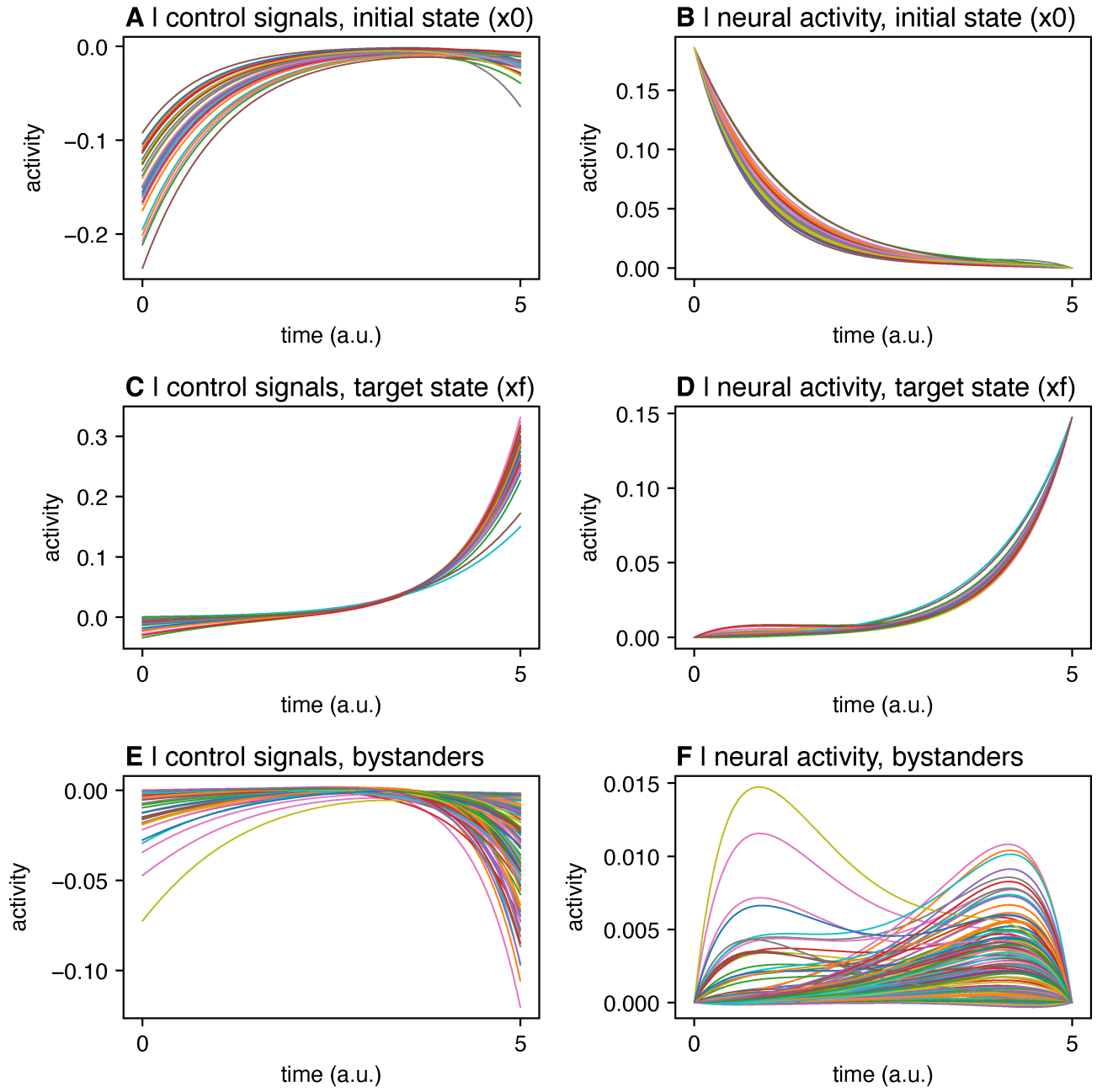

FIG. S7
Control signals and state trajectory. Uniform full control set. T=5. Initial state = visual system. Target state = default mode system. Inversion error = 1.33×10^{-13}. Reconstruction error = 1.03×10^{-11}. Energy = 1797.

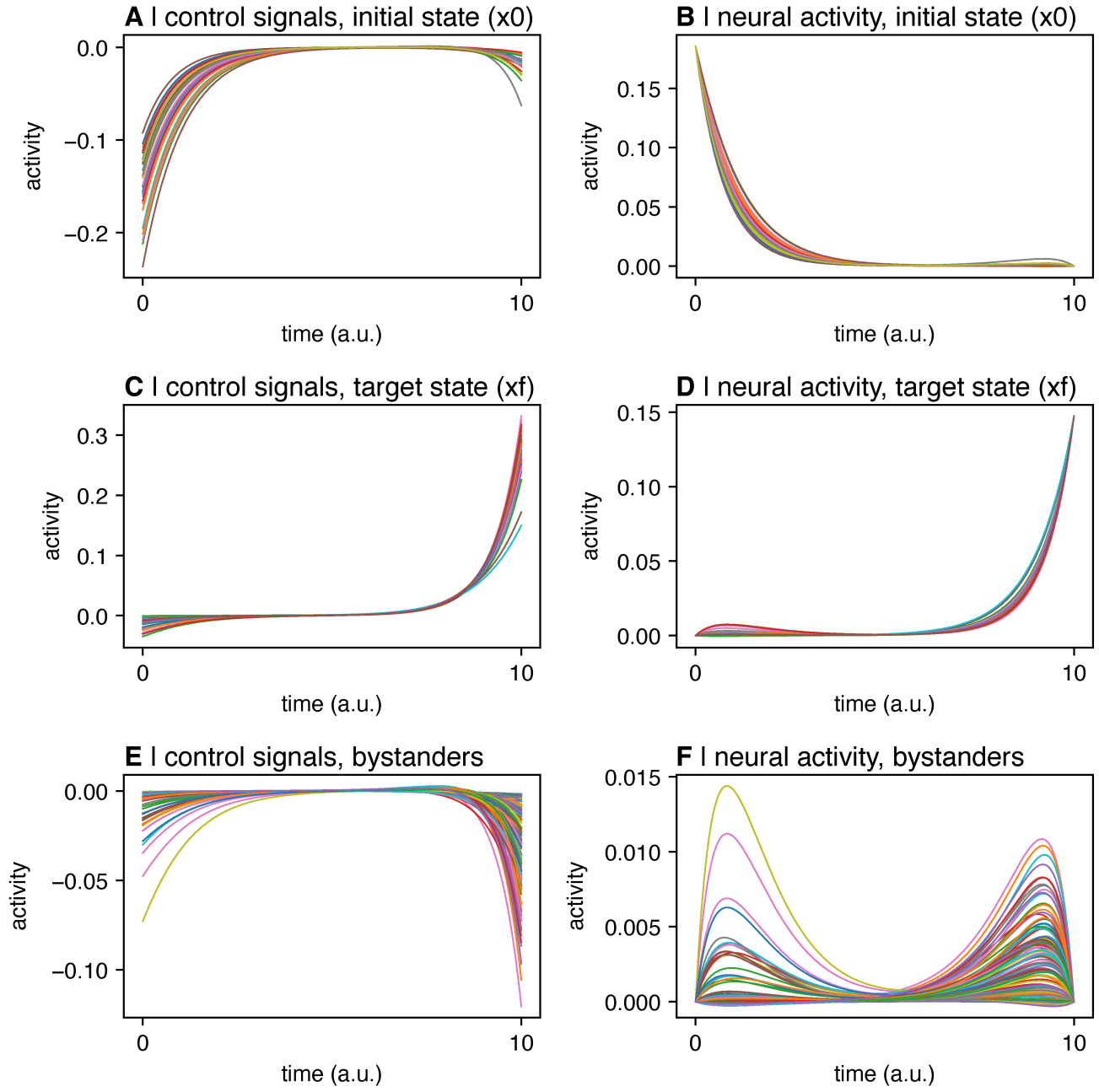

FIG. S8
Control signals and state trajectory. Uniform full control set. *T=10*. Initial state = visual system. Target state = default mode system. Inversion error = 1.48×10^{-10}. Reconstruction error = 1.71×10^{-8}. Energy = 1801.

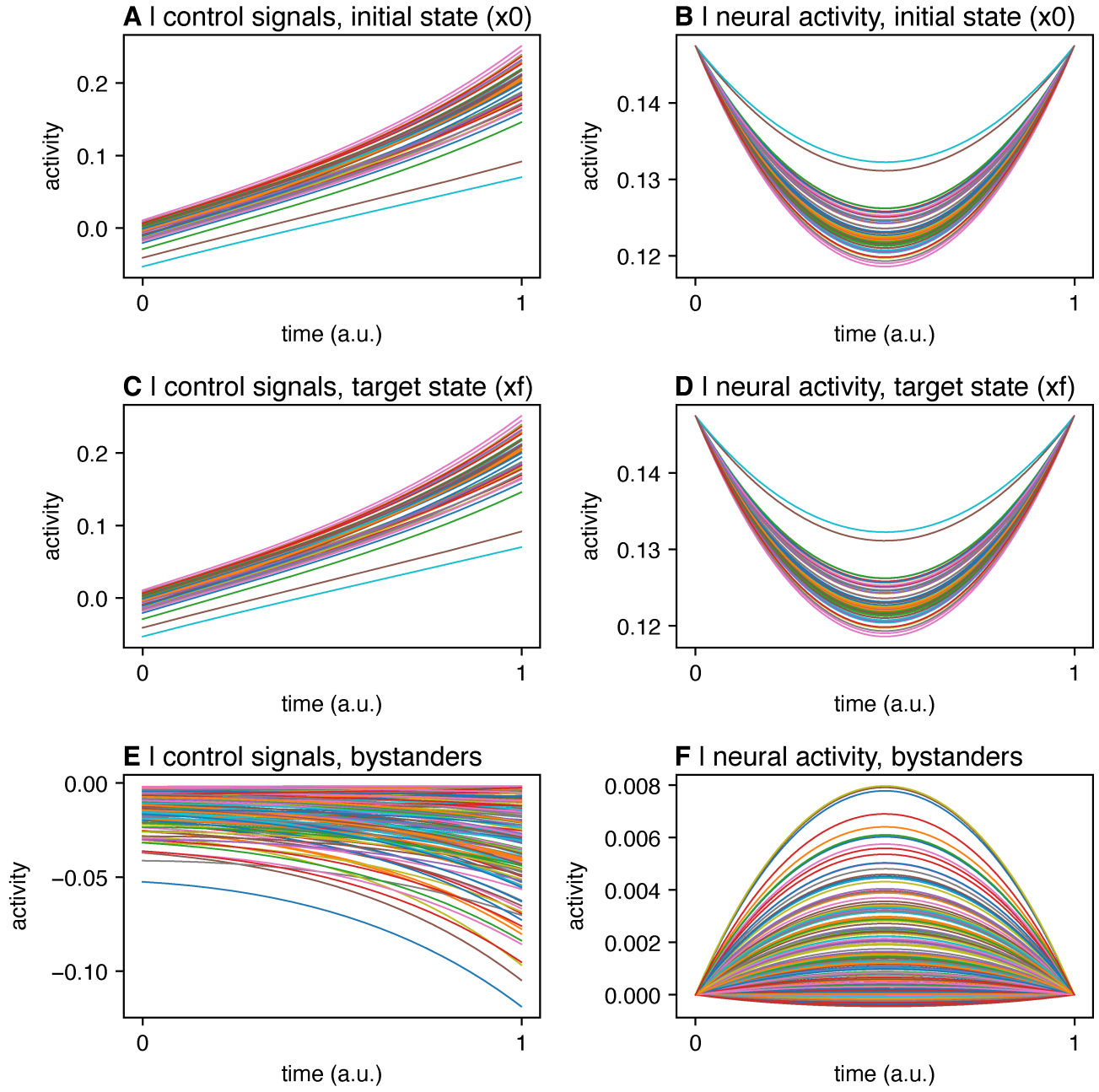

FIG. S9
Control signals and state trajectory. Uniform full control set. T=1. *Initial state = default mode system. Target state = default mode system.* Inversion error = 2.20×10^{-16}. Reconstruction error = 3.95×10^{-14}. Energy = 571.

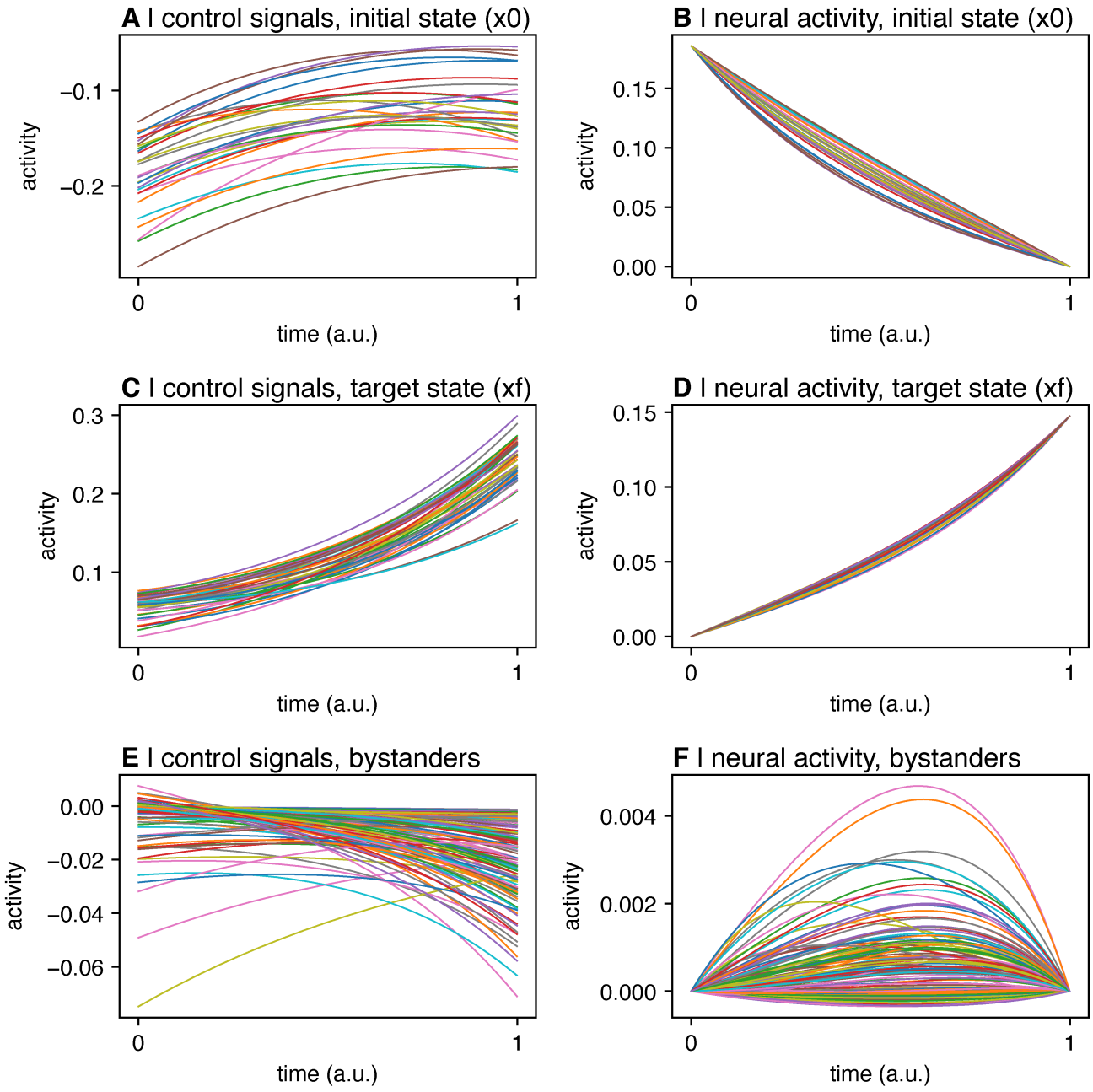

FIG. S10
Control signals and state trajectory. *Annotation map control set*. T=1. Initial state = visual system. Target state = default mode system. Inversion error = 1.87×10^{-15}. Reconstruction error = 6.07×10^{-14}. Energy = 1455.