ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN

DATA SCIENCE AND COMPUTATION

Ciclo XXXVI

Settore Concorsuale: 02/B2 - FISICA TEORICA DELLA MATERIA

Settore Scientifico Disciplinare: FIS/02 - FISICA TEORICA, MODELLI E METODI MATEMATICI – PHYS-04/A – FISICA TEORICA DELLA MATERIA, MODELL METODI MATEMATICI E APPLICAZIONI

MULTIVARIATE INFORMATION THEORETIC METHODS FOR THE ANALYSIS OF NEURAL NETWORK FUNCTION

Presentata da: Gabriel Matías Lorenz

Coordinatore Dottorato

Daniele Bonacorsi

Supervisore

Stefano Vittorio Tiziano Panzeri

Co-supervisore

Andrea Cavalli

Esame finale anno 2025

Multivariate information theoretic methods for the analysis of neural network function

Gabriel Matías Lorenz

Abstract

To understand brain functions, it is necessary to characterize how neural systems encode, process, and transmit information. Information theory provides multivariate analysis tools to address these questions by analyzing activity recordings from real brains or neural network models. These tools are model-independent and can be applied to any recording modality and across different scales. In this thesis, we improved and used information-theoretic tools to study the brain in several ways.

We developed the Multivariate Information in Neuroscience Toolbox (MINT), designed for analyzing neural information. MINT includes tools such as Shannon entropy, mutual information, transfer entropy, and Partial Information Decomposition (PID). It enables researchers to quantify how neural populations encode and transmit behaviorally relevant information across brain regions, enhancing investigations into neural computation. By integrating dimensionality reduction techniques and bias-correction methods, MINT allows precise analysis of high-dimensional neural datasets.

A significant limitation in computing PID components from neural data is sampling bias, particularly in synergy, which increases quadratically with the number of possible neural responses, leading to overestimations. To address this, we developed bias-correction methods that enhance PID estimation accuracy. We applied these methods to data from the auditory cortex, posterior parietal cortex, and hippocampus of mice engaged in cognitive tasks, deriving accurate estimates of how synergy and redundancy vary across regions.

Additionally, we used MINT to analyze simulated spiking neural network models to explore contributions of different cortical interneurons to information encoding. Previous models with a single interneuron type revealed redundant encoding in the gamma frequency range. In contrast, our extended models showed distinct gamma frequencies carry synergistic information about sensory inputs, suggesting interneuron diversity enhances information encoding. Together, our methodological work and network model findings highlight the potential of information theory for advancing understanding of neural encoding and information transmission.

Acknowledgments

First and foremost, I extend my heartfelt gratitude to my family. Their unwavering support, encouragement, and belief in my potential have been my anchor throughout this journey, even from afar. Without their guidance and patience, this accomplishment would not have been possible.

My sincerest thanks go to my supervisor, Stefano Panzeri, for his invaluable mentorship and expertise. His guidance has guided me through many challenges, and his advice has profoundly shaped the direction of my work.

I am also deeply grateful to Nicola Engel, whose insights, constructive feedback, and enthusiasm pushed me to to refine and improve my research. Her dedication and support have been instrumental in the completion of this thesis.

A special thank you to my lab colleagues, who have made the research environment collaborative, inspiring, and enjoyable. Their assistance, and shared dedication made the lab a place where challenges were met with enthusiasm and teamwork.

To my friends, thank you for your kindness, encouragement, and constant reminders to take breaks and stay balanced. Your support and laughter provided much-needed moments of joy and relief during intense times.

To each of you, thank you for your support, patience, and belief in me throughout this journey. This work would not have been possible without you.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 The brain as an information processing center

The brain is a complex organ responsible for an extraordinary array of tasks essential to survival, cognition, and adaptation. It is made by a vast network of neural cells that enables not only the regulation of basic bodily functions, such as breathing and heartbeat, but also the execution of complex motor actions and the processing of sensory information. Its incredible processing power allows the brain to react swiftly to external stimuli, aiding in the avoidance of dangers and in the pursuit of basic needs, like food and shelter. Beyond reacting to its environment, the brain can also learn from experiences, adapt based on prior outcomes, and store information, forming memories that contribute to decision-making and behavior regulation [1].

Sensory information from the surrounding environment reaches the brain through a process of detection, encoding, and transmission. This process begins at specialized sensory receptors located in the periphery—such as photoreceptors in the eyes and mechanoreceptors in the skin—which convert physical stimuli into electrochemical signals. Each receptor encodes specific stimulus features through unique patterns of action potentials [1]. These signals travel through the nervous system, reaching the brain stem and moving up to the thalamus. The thalamus functions as a critical relay station, distributing this sensory data to relevant sensory processing areas in the cerebral cortex, the outermost layer of the brain and a primary site for advanced cognitive functions cortex, where sensory information undergoes detailed processing to create coherent and actionable internal representations of the external world. Through a coordinated interplay between sensory and integrative cortical areas, the brain can synthesize input from multiple senses, allowing it to interpret, remember, and learn from the incoming data. The formation of these representations facilitates the brain's ability to predict and respond appropriately to environmental stimuli.

When sensory information necessitates an immediate behavioral response, such as fleeing from a threat or reaching for an object, the brain translates perception into action. Decision-making centers within the cortex communicate the need for movement to motor areas. This information is then relayed down through the brain stem and out to the spinal

cord, where it ultimately reaches the muscles, activating precise contractions that lead to coordinated movement. From perception to behavior, the brain operates as a sophisticated information-processing center that seamlessly integrates external stimuli, internal representations, and motor responses.

1.2 Multiscale processes in the brain

The myriad of tasks managed by the brain would be impossible without the synchronized actions of numerous processes occurring simultaneously across different levels of organization. These processes span from the activity of individual neurons to computations carried out by large brain regions and occur on timescales ranging from milliseconds to decades, reflecting both rapid responses to immediate stimuli and the gradual adaptations seen in long-term learning and memory formation [2].

At the most fundamental level, neurons serve as the primary information-processing units of the brain. Each neuron is capable of encoding, transmitting, and even performing basic computations on incoming information. For example, neurons in the retina can capture elementary visual features such as contrast, motion, and edges within specific regions of the visual field, known as receptive fields [3]. These neurons then relay this information in the form of action potentials, enabling the brain to construct visual representations from basic sensory input. Single neurons in the human medial temporal lobe encode numerosity and abstract numerals [4] and neurons in the superior temporal gyrus encode speech sound cues such as relative vocal pitch and onsets [5].

While single neurons are capable by themselves to encode and process information, this does not give us an accurate description of how neuronal populations compute. Single cell analysis is often done through trial averaging as a way to compensate neural variability but the brain needs to process information from single events, which means interpreting single cell variability in the context of the general population activity [6, 7]. When neurons work as a population, they form interconnected circuits that can detect complex patterns, support the integration of sensory information across modalities, and guide behavior [8]. The precise temporal correlation between neurons in the lateral geniculate nucleus improve considerably the information that can be extracted about visual stimuli [9]. The recording on the average electrical activity that is produced extracellularly, shows the presence of electrical waves that participate in the encoding of information [10–12].

These neural populations often operate in parallel, performing distinct computations within specialized areas, yet they are interconnected through networks that enable integrated processing [2, 13]. Populations communicate through white matter allowing the brain to merge separate streams of sensory information, integrate memory, and adjust motor responses accordingly. This network connectivity is critical for creating a cohesive experience and adapting behavior based on past knowledge, ongoing sensory input, and future predictions.

1.3 Recording techniques in the brain

As we have seen, the brain hosts a multitude of processes that operate across a wide range of temporal and spatial scales. To better understand these intricate activities, researchers have developed a variety of tools and techniques that enable us to inspect brain function at different levels. These methods include both invasive techniques, which require the implantation of instruments within the brain, and non-invasive approaches that allow for observation from outside the body.

One of the most powerful techniques for examining localized neuronal activity is electrophysiological recording. This method involves inserting microelectrodes into specific areas of the brain, where they can detect extracellular currents and identify individual neuronal spikes by filtering out lower-frequency signals [14]. Recent advancements in technology have made it possible to utilize multiple electrodes simultaneously, facilitating the recording of a larger number of neurons at once. This capability not only increases the volume of data collected but also allows for the estimation of the spatial location of neurons through triangulation methods, enhancing our understanding of their spatial organization.

In addition to capturing action potentials, electrophysiological recordings can also measure extracellular activity across lower frequency ranges, which is crucial for understanding broader brain dynamics. The Local Field Potential (LFP) represents the lower frequency component of the electrical signals recorded by the electrodes [10]. It provides an indirect measure of the collective activity patterns of neuronal populations, reflecting the overall state of brain activity and synchronization among different brain regions. Its analysis allows more information to be extracted from sensory populations [15, 16]. Despite their advantages, electrophysiological methods come with significant drawbacks. The insertion of electrodes into the brain inevitably causes tissue damage, which can affect the health and functionality of the surrounding neural tissue. Furthermore, the number of neurons that can be recorded is inherently limited by the number of electrodes used. Increasing the number of electrodes to capture more neurons further compounds the risk of additional damage, posing challenges for long-term studies of brain function.

Calcium imaging is a powerful non-invasive technique for recording the activity of neuronal ensembles without directly disrupting brain tissue [17]. This method operates by tracking changes in intracellular calcium concentration, utilizing fluorescent calcium indicators that respond to fluctuations in calcium levels associated with neuronal activity. Calcium plays a vital role in numerous cellular processes, and elevated levels in the soma can serve as indicators of action potential firing. When combined with genetic tools, fluorescent indicators can be targeted to specific neuronal populations, allowing for detailed investigations of their activity. However, while calcium imaging is beneficial for observing large groups of neurons, it lacks the temporal precision of electrophysiological recordings, meaning it cannot reliably capture every action potential within a population.

Even though calcium imaging minimizes direct interference with brain tissue, it still requires the creation of an opening in the skull to allow the optical probe to access the brain. Fortunately, there are also non-invasive methods that rely on the electromagnetic

activity produced by the brain and can be measured externally. Techniques such as electroencephalography (EEG) and magnetoencephalography (MEG) record brain activity by detecting the electrical and magnetic fields generated by the neuronal currents [2]. These methods provide valuable insights into the brain's overall activity patterns without the need for surgical intervention, making them particularly useful for studies involving human subjects or long-term monitoring of brain function [18–20].

Together, these diverse recording techniques offer complementary insights into the brain's complex information processing and neural dynamics, advancing our understanding of how the brain operates across different scales and contexts.

1.4 Information theory in Neuroscience

As we have seen, the concept of information is integral to understand the the functional workings in the brain, regardless of which scale or section we are studying. Also, the diverse range of recording types allows us to observe brain activity at different scopes. Information theory offers a solid mathematical foundation to work quantitatively with the processes of encoding, processing and transmission of information [21]. It can be applied to any data type or combination of them, it can detect non-linear relationships between the variables without the need to first build a model that relates the studied variables [22]. Below we will describe briefly the key concepts in the field that will be discussed through the whole work.

To understand what information means mathematically, we first need to establish a precise definition. We can define the level of uncertainty regarding an outcome through the probabilities of each possible outcome, leading us to the definition of entropy H:

$$H(\mathbf{R}) = -\sum_{\mathbf{r}\in\mathbf{R}} p(\mathbf{r}) \log_2(p(\mathbf{r})). \tag{1.1}$$

Here, $H(\mathbf{R})$ quantifies the uncertainty associated with a random variable $\mathbf{R}$ [23]. The core idea is that the greater the uncertainty, the higher the entropy.

In relation to information, we can express the concept of mutual information between two variables. The information gained about one variable after knowing the state of another variable can be mathematically represented as:

$$I(\mathbf{R};\mathbf{S}) = H(\mathbf{R}) - H(\mathbf{R}|\mathbf{S}), \tag{1.2}$$

where $H(\mathbf{R}|\mathbf{S})$ is the conditional entropy, representing the remaining uncertainty about the variable $\mathbf{R}$ given the knowledge of $\mathbf{S}$. This formulation allows us to quantify how much uncertainty is reduced in $\mathbf{R}$ when we have additional information from $\mathbf{S}$.

In the context of brain dynamics, we can further extend these concepts to explore the interactions between different regions. Transfer entropy (TE) is a measure that quantifies the amount of information that a 'sender' variable provides about a 'receiver' variable that cannot be explained solely by the past activity of the receiver [24]. This measure allows us to capture the directional influence one neural population may exert over another, providing insights into the flow of information within neural circuits.

These two quantities—mutual information and transfer entropy—serve as powerful tools to enhance our understanding of brain function. For instance, we can apply these concepts to study the encoding performance of neural populations[6, 25–29].

How can we measure the gain in encoding efficiency that groups of neurons exhibit compared to their individual contributions? Additionally, what role do interconnections among neurons play in either enhancing or inhibiting this encoding? We may also explore the extent to which various stimulus features are represented uniformly across different neurons versus those that are uniquely encoded by specific cells.

One framework that addresses these questions is Partial Information Decomposition (PID) [30, 31]. PID allows us to break down the total information of a collection of sources into more elemental components, with each component representing the information contributed by a specific combination of shared and unique information among the sources. This method offers insights into the functional structure of a population's encoding.

However, while PID is valuable for analyzing two sources, its interpretation can become complex when dealing with multiple sources, often lacking direct mechanistic explanations. Another useful framework for evaluating the encoding of neuron populations is the Information Breakdown [32–34]. This framework expresses the total information a population encodes about a stimulus as a sum of positive and negative components, representing encoding enhancements and inhibitions that arise from measured correlations in neuronal activity.

It is possible that only a small portion, or even none, of the information encoded is actually used to guide behavior. Intersection Information (II) measures how much of the sensory information encoded in neural population activity is read out to inform behavior, and is computed with PID as the component of neural information that is both about stimulus and choice [35–37].

Measures like Feature-specific Information Transfer (FIT) provide a precise analysis of information transmission between neural populations [38]. FIT quantifies the fraction of overall information transfer—essentially transfer entropy—attributable to the information encoded by the sender population regarding a specific stimulus S. Since there can be multiple pathways for information transmission to a receiver population, a conditional variant of this measure, known as conditional FIT (cFIT) [38], can help clarify whether the information about the stimulus is genuinely being transmitted from the sender population or is instead coming from a third, confounding population.

By employing these advanced measures and frameworks, researchers can gain deeper insights into the encoding mechanisms of neural populations and the dynamic interactions that characterize information processing in the brain. This multifaceted approach enhances our understanding of how neural circuits operate, revealing the complex interplay between individual neurons, population dynamics, and information transmission across the brain.

In chapter 2, we provide a toolbox with computational resources for information analysis. The toolbox contains functions to implement all the measures described above plus preprocessing pipelines with binning and correction algorithms.

1.5 Challenges of information analysis in neuroscience

Information theory serves as a powerful tool for neuroscientists seeking to understand the functions of neural networks. However, while this analytical framework offers valuable insights, it also comes with certain caveats that researchers must navigate. One significant challenge is that these measures rely heavily on the accurate estimation of the joint probability distribution of neural responses and the associated stimuli. Consequently, researchers must provide datasets that truly reflect these distributions, a task that becomes increasingly daunting as the size of the neural population under investigation grows. This is primarily due to the exponential increase in the number of potential response combinations that must be sampled to achieve an accurate representation.

This challenge has a direct impact on the estimation of information. When the distribution is undersampled, the resulting entropy estimation is overestimated [39, 40]. Specifically, because the conditional entropy is typically the least sampled term in the mutual information equation, underestimating this component leads to a positive bias in the estimation of mutual information itself. As illustrated in equation 1.2, this bias can skew our understanding of the true informational dynamics at play within neural populations.

The issue of undersampling is not new; it has been extensively studied within the field. Various bias correction methods have been developed to mitigate the effects of undersampling when it is impractical to increase the number of trials in a dataset [39–43]. Many of these techniques have been specifically designed to address the estimation of mutual information, and have undergone testing to validate their effectiveness.

Another possible method is to reduce the impact of high-dimensionality is to apply a dimensionality reduction technique on the recording of population [44] and later use the reduced activity in the information analysis.

In addition to mutual information, efforts have also been made to apply bias correction to the components of information breakdown. However, despite these advancements, there has yet to be a systematic exploration of the biases affecting Partial Information Decomposition (PID) measures. Current research in this area has primarily focused on Gaussian variables [45], leaving a significant gap in our understanding of how PID might be influenced by undersampling in more complex or non-Gaussian distributions.

This highlights the need for further investigation into the systemic biases inherent in PID measures and the development of appropriate correction methods. By addressing these limitations, researchers can enhance the reliability of information-theoretic analyses in neuroscience, leading to a more accurate understanding of neural encoding and information processing in the brain. Ultimately, a robust framework for estimating information content is essential for advancing our comprehension of neural network dynamics and their implications for behavior and cognition.

In chapter 2, we provide tools of bias correction of all the information measures described in the section above. We also show how the comparison of different dimensionality reduction techniques leads to conclusions on the nature of population coding. In chapter 3, we perform a study on the bias produced by undersampling on the components of a two-source PID analysis.

1.6 Information theory in network modeling

So far, the methods discussed have primarily focused on analyzing recorded neural data through the framework of information theory. However, while one of the main strengths of information theory is the lack of necessity to assume a model, it also means that its results do not give us an idea of the mechanisms behind the information encoding of a system. Nevertheless, information-theoretic approaches can also be effectively combined with modeling techniques. By representing the dynamics of observed measurements through selected models that simplify the conditions of the studied system, we can leverage the information values derived from these models to gain insights into the encoding performance of the phenomena being modeled. This approach allows us to minimize the interference of confounding processes that might occur in real neural systems.

For instance, if we consider that the activity of a neural population exhibits minimal sensitivity to trial-to-trial fluctuations in the specific spike patterns generated by individual neurons, we can construct a model where each neuron's response is described by its average firing rate [3]. This firing rate model simplifies our understanding of how populations of neurons encode information by focusing on the mean activity rather than the details of spike timing.

However, while firing rate models offer a straightforward framework for analysis, they may not sufficiently capture the dynamics of a real neural population under certain conditions. For example, in networks that exhibit synchronous firing, the precise timing of spikes becomes crucial for understanding the underlying neural computations. In such cases, relying solely on average firing rates can obscure important information about the temporal structure of neuronal activity.

To address these limitations, researchers often turn to spiking network models, which are designed to capture the dynamics between individual neurons more accurately. These models account for the detailed interactions among neurons and can represent the precise timing of spikes, allowing for a more comprehensive understanding of network behavior [3, 46]. The complexity of a spiking network model will depend on the sophistication of the individual neuron model it employs.

For example, simpler representations can treat neurons as point objects, where the model focuses exclusively on an average representation of the entire cell's dynamics. This approach is often exemplified by models such as the integrate-and-fire model, which simplifies the neuron's behavior to key dynamics without delving into the biophysical details [46]. Conversely, more complex models, like the Hodgkin-Huxley model, incorporate detailed biophysical properties, including ion channel dynamics and membrane potential changes, to simulate the realistic behavior of neurons [46].

By employing these spiking network models, researchers can better capture phenomena such as synchronous firing and oscillatory patterns, which are critical for understanding various cognitive processes and behaviors. These models allow for an investigation of how the timing of spikes contributes to information processing and encoding within neural circuits. Ultimately, the integration of information theory with sophisticated modeling techniques enriches our understanding of neural dynamics and enhances our ability

to analyze the encoding performance of neuronal populations in a more nuanced manner.

As an example, to study the generation and role of gamma oscillation oscillations (40 – 100 Hz) in the cortical network measured through LFP [15, 16]. Previous theoretical and computational studies of gamma oscillations typically included excitatory neurons and a single type of inhibitory neurons [11, 47–51]. The model of this inhibitory neuron did not specify or differentiate the type of inhibitory being modeled, but that was loosely matched to the properties of fast-spiking parvalbumin neurons, the most common inhibitory neuron in many cortical regions. These models could explain the generation of gamma oscillations exhibiting realistic spectral features, and could also explain the privileged encoding of information from the sensory periphery by the power of activity in the gamma band.

In chapters 2 and 3, we use rate models to provide ground truth dynamics that we use to test the reliability of the information measures and the bias corrections. In chapter 4, we model the local field potential activity of a cortical network by using a recurrent spiking network that outputs an aggregate measure of electrical activity. The activity simulated from this network is later analyzed with information theoretic tools to analyze the presence of synergy across frequency bands.

1.7 Overview of the chapters

In this work, we will work on the development and application of information-theoretic tools designed to improve the research on multivariate neural activity.

In chapter 2, we present a computational toolbox built with many long-standing and state-of-the-art information measures. We also provide preprocessing pipelines that are suited to any data type, and tools for afterward processing and analysis. We showcase the toolbox capabilities by applying not only to simulated scenarios to verify the analyses coincide with the designed ground-truth, but we also apply to real datasets from different experimental setups to showcase its usefulness and versatility.

In chapter 3, we focus on the problem of the limited sampling bias in partial information decomposition. We conduct an explorative analysis to characterize the bias influence in different scenarios. We apply on these terms correction procedures that were designed originally for mutual information and we test their efficacy. We then apply these validated procedures onto the analysis of real neuron pairs from different published experiments and we showcase the impact of these corrections on the correct interpretation of the PID results.

In chapter 4, we apply some of the tools developed on the toolbox for an analysis through simulated models. We implement a model of a recurrent cortical population that accounts for the cellular diversity in inhibitory neurons and we analyze how these results compare with previous models with no inhibitory neuron diversity.

Chapter 2

MINT: a toolbox for multivariate neural data analysis

The content of this chapter was submitted for publication, and is currently under submission and being revised [52]. The analyses presented here correspond to those presented in the first submitted version of the paper prior to being revised.

2.1 Introduction

Brain functions are based on the ability of groups of neurons or brain areas to encode, process and transmit information [53, 54]. Consequently, information theory [21], the mathematical theory of communication, has deeply influenced the conceptualization of brain operations. It has become a method of choice to analyze neural activity because of its many advantages [6, 25–28]. It provides single-trial measures of how neural activity encodes variables important for cognitive functions such as sensory stimuli, and it is thus more relevant for single-trial behavior than trial-averaged measures. It captures contributions of both linear and non-linear interactions between variables at all orders, and thus allows hypotheses-free measures of information encoding that place upper bounds to the performance of any decoder. Because of its generality, it can be applied to any type of brain activity recordings. Also, it facilitates direct comparisons between the predictions of normative neural theories and real neural data [26, 29].

Earlier work using information theory to analyze empirical neural data has focused on low-dimensional measures of neural activity such as single neurons, small neural populations or aggregate measures (LFPs, M/EEG, fMRI). These studies have considered only how information is encoded in neural activity, regardless of how it may be used downstream. Such seminal studies have demonstrated e.g. how the temporal structure of neural activity (from single-neuron spike timing to network oscillations [55–61]) contributes to sensory encoding, or how neural mechanisms such as adaptation contribute to brain information processing [28, 62].

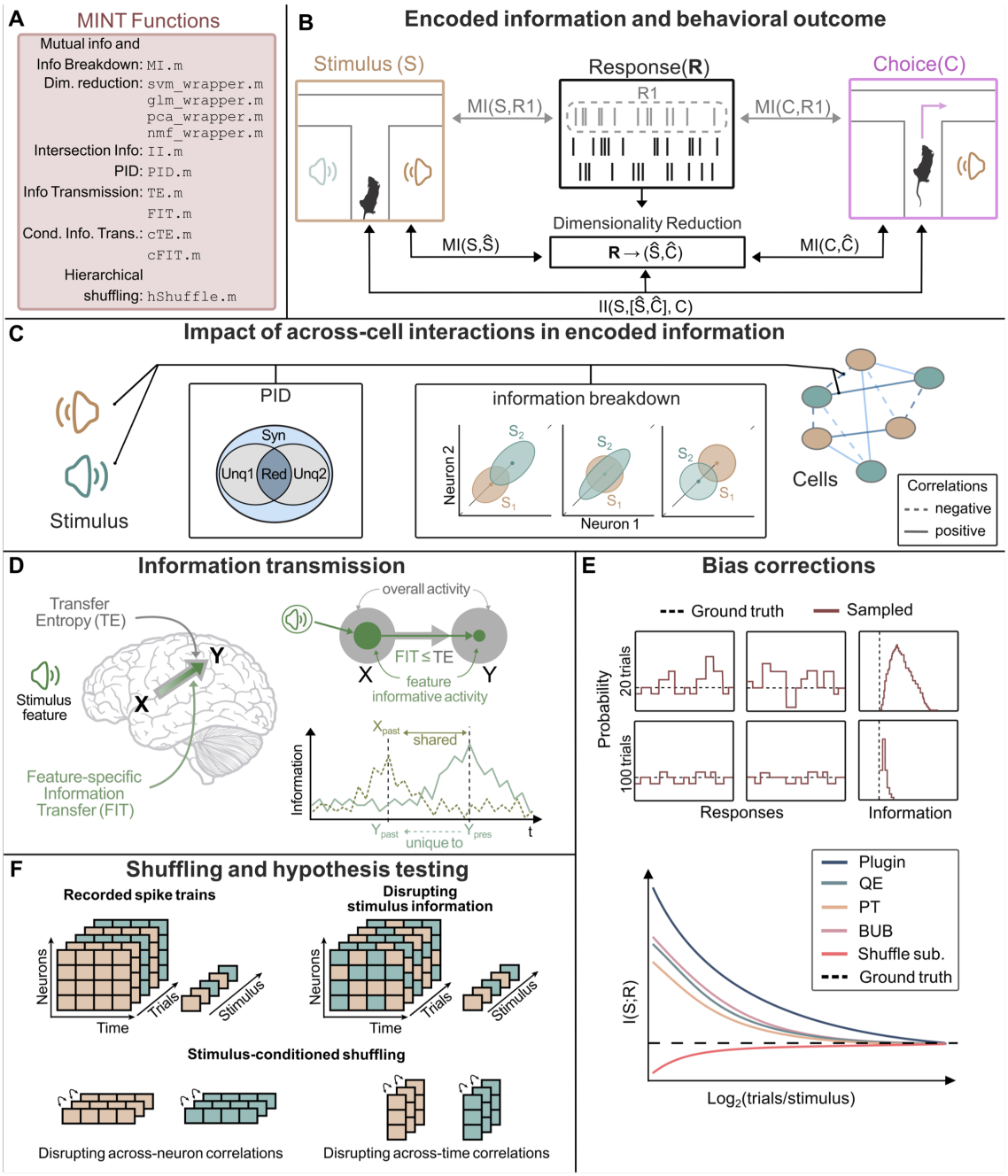
A
MINT Functions
Mutual info and Info Breakdown: MI.m
Dim. reduction: svm_wrapper.m glm_wrapper.m pca_wrapper.m nmf_wrapper.m
Intersection Info: II.m
PID: PID.m
Info Transmission: TE.m FIT.m
Cond. Info. Trans.: cTE.m cFIT.m
Hierarchical shuffling: hShuffle.m
B
Encoded information and behavioral outcome
Stimulus (S)
Response(R)
Choice(C)
R1
MI(S,R1)
MI(C,R1)
Dimensionality Reduction
MI(S,Ŝ)
R → (Ŝ,Ĉ)
MI(C,Ĉ)
II(S,[Ŝ,Ĉ], C)
C
Impact of across-cell interactions in encoded information
Stimulus
PID
Syn
Unq1
Red
Unq2
information breakdown
Neuron 2
Neuron 1
Cells
Correlations
negative
positive
D
Information transmission
Transfer Entropy (TE)
Stimulus feature
Feature-specific Information Transfer (FIT)
overall activity
FIT ≤ TE
feature informative activity
shared
unique to
Information
t
E
Bias corrections
Ground truth
Sampled
Probability
20 trials
100 trials
Responses
Information
Plugin
QE
PT
BUB
Shuffle sub.
Ground truth
I(S;R)
Log2(trials/stimulus)
F
Shuffling and hypothesis testing
Recorded spike trains
Disrupting stimulus information
Neurons
Time
Trials
Stimulus
Stimulus-conditioned shuffling
Disrupting across-neuron correlations
Disrupting across-time correlations

Figure 2.1: Overview of MINT. **A: A:** List of main MINT functions. **B:** MINT provides multivariate information theoretic functions to quantify the amount of information that single neurons or neural populations carry about task-relevant variables (e.g. sensory stimuli or behavioral choices). These methods are based on either direct-method calculation of information from the discretized probabilities (ideal for small neural populations but not scalable with population size), estimation through other techniques such as Kernel-based methods that can operate on real-valued data, or by using supervised or unsupervised dimensionality-reductions techniques to approximate high-dimensional neural population response probabilities with probabilities in lower-dimensional spaces (scalable with population size). It also provides tools to quantify how much of the information encoded by neural activity is used to inform behavior. **C:** MINT has multiple functions to compute, in small or large populations, how interactions between the activity of different neurons shape information encoding and create synergy or redundancy. **D:** MINT has tools to compute transfer of information from one neural population or brain region to another. It can compute both the total or stimulus-specific information transmitted between two nodes, with the option of conditioning over the activity of other nodes. **E:** MINT has tools to correct for the limited-sampling bias, an essential tool for analysis of empirical neuroscience data. **F:** MINT has a set of hierarchical permutation algorithms that provide null hypothesis testing for significance of information encoding and information transmission and for the impact of correlations across neurons or time. Mouse sketch is modified from `doi.org/10.5281/zenodo.3925917` and brain sketch is modified from `doi.org/10.5281/zenodo.3925989`.

Over the last decade, neuroscience has seen major progress in the ability to record simultaneously the activity of many neurons and/or brain areas. These advances have driven the development of novel information theoretic analytical tools to investigate how information processing emerges from the interaction and communication among neurons or areas. Studies have provided multivariate information tools to individuate when synergy and redundancy arise in small populations, or to understand the mechanisms for generating redundancy and synergy, for example to characterize how correlations between the activity of different neurons shape information processing[32–34, 53, 63, 64]. Recent work has also coupled information theory with dimensionality-reduction techniques to study how information is encoded in populations of tens to hundreds of cells[35, 65–72]. Other studies have developed multivariate information theory to quantify transmission, rather than encoding, of information across neurons or areas [24, 38, 73–81]. These methods measure the overall or stimulus-specific information exchanged between simultaneously recorded neurons and areas and determine whether transmission relies on synergistic integration of information across nodes. Another major direction of progress has been in recording neural activity during behavior [82]. To support the growing interest on how neural computations shape behavior, information theory has produced tools to characterize the multivariate simultaneous relationship between sensory stimuli, neural activity and behavioral output to enable quantifying the impact on behavior of the information encoded in a certain area or population [35–37, 83].

While the use and dissemination of information theoretic algorithms has been aided by software toolboxes [22, 84–101], no toolbox yet provides a comprehensive implementation of tools to compute both information encoding and transmission, to break down information into components reflecting the effect of interactions and to quantify behavioral or downstream relevance of the encoded information (see Table A.1). To fill

these gaps and address the need to collect organically these tools in a format that allows immediately multiple analyses, here we introduce a new Multivariate Information in Neuroscience Toolbox (MINT). MINT provides a comprehensive set of information theoretic functions (including Shannon Entropy and Mutual Information, directed information transmission measures, information decompositions) and estimators (binned probability estimators, limited-sampling bias corrections). The implemented information-theoretic functions are detailed in A. What they compute, and how they can be used in neuroscience is summarized in Table 1. The accuracy and applicability of these algorithms has been validated and demonstrated extensively with both discrete neural variables, such as spikes in electrophysiological recordings [28, 58, 59, 102, 103], and continuous neural variables, e.g. LFP, M/EEG, fMRI and calcium traces [35, 72, 104–106].

Importantly, as we demonstrate with examples, combining these multivariate tools enables addressing questions that cannot be addressed with a single tool. For example, combining tools to identify the specific contribution of correlations to population encoding or the amount of encoded information that informs behavior with dimensionality-reduction techniques allows understanding how large neural populations influence behaviors. Combining information encoding tools with content-specific information transmission tools can reverse engineer information flow in neural networks with unprecedented understanding. We thus anticipate that MINT will lead to uncover numerous new insights into neural information processing.

2.2 Design and Implementation

MINT is written in MATLAB (version 2018b or newer) and depends on the Statistics and Machine Learning, Optimization, Parallel Computing and Signal Processing Toolboxes. MINT takes as input neural data (array of neural activity recorded in each trial) and task variables (sensory stimuli or behavioral responses presented or produced in each trial). It outputs information values and their null-hypothesis values for computing statistical significance. A.1 illustrates MINT functions, options, and core routines.

MINT computes Entropy (`H.m`), which measures neural variability; Mutual Information (`MI.m`), which measures information encoding (Fig. 2.1B). It computes the Information Breakdown of Shannon Information into contributions due to correlations between neurons [32, 33, 63, 107, 108]. It also computes Partial Information Decompositions (PID) [30, 109] of the information about a target variable carried by two or more source variables into unique, synergistic and redundant information (Fig. 2.1C). Computation of PID requires specifying a redundancy measure, which can be selected by the user among options [30, 31, 91, 110] with complementary advantages [110, 111]. (Redundancy of [91] requires either a MATLAB-compatible C compiler or pre-compiled files made available by us for Windows 11, macOS, and Linux Debian). MINT computes additional functions of neuroscientific value: Intersection Information (II, function II.m; Fig. 2.1B see [37]), the amount of stimulus information in neural activity that is used to inform behavior; Transfer Entropy (TE, see [112]) and Feature-Specific Information Transfer (FIT,

Function	What is Computed	What it is Used for
Entropy (H)	Variability of random variable X	Assess variability of neural activity
Mutual Information (MI)	How well an ideal observer can predict X from single-trial observations of Y	How well activity of population of neurons encodes info about task variables (e.g. stimuli)
Information Breakdown	MI about X carried by a multivariate Y, broken down into contributions arising from correlations between different dimensions of Y	How correlations between multivariate neural activity (e.g. activity of populations of neurons) contribute to population-level encoding of task variables (e.g. sensory stimuli)
Partial Information Decomposition (PID)	Decomposes MI carried by a multivariate Y about X into unique info about X carried by each element of Y, synergistic info found only in the interactions between elements of Y, and redundant info shared among elements of Y	Whether groups of neurons or brain areas carry synergistic or redundant info about task variables (e.g. stimuli) or about activity of other neurons or brain areas
Redundancy-Synergy Index (RSI)	Quantifies whether the effect on encoding info about X of interactions between different variables within a multidim. Y is predominantly redundant or synergistic	Whether groups of neurons or brain areas carry predominantly synergistic or redundant info
Intersection Information (II)	Info about X carried by Y which is used to inform Z about X	How much of the info about a task variable (e.g. sensory stimulus) encoded in neural activity is used to inform behavioral reports (e.g. choices)
Transfer Entropy (TE)	MI about the past activity of sender X found in present activity of receiver Y, conditioned on past activity of Y	Measures overall transmission of info between nodes of a neural network
Feature Specific Information Transfer (FIT)	MI about feature S presently encoded by Y redundant with MI about S previously encoded by X and unique with respect to MI about S previously encoded by Y	Measures transmission between nodes of a neural network of info about a specific feature of task variables (e.g. sensory stimuli)
Conditional TE (cTE) and Conditional FIT (cFIT)	Versions of TE and FIT with info flow conditioned or unique with respect to another variable	Measure overall or feature specific transmission of info between network nodes discounting info possibly passing through other nodes
Supervised dimensionality reduction (decoding)	Project data onto lower-dimensional space using labeled data to optimize decoding	Intermediate step for info calculations with large neural populations
Unsupervised dimensionality reduction	Project data onto lower-dimensional space using unlabeled data to optimize data explainability	Intermediate step for info calculations with large neural populations
Limited-sampling bias correction	Produces unbiased info estimates unaffected by the limited-sampling bias	Needed to obtain more accurate info estimates in all practical situations
Hierarchical Data shuffling	Random permutations of data used to create null distribution for statistical testing for assessing the role of certain data features in info encoding	Needed for statistical testing in all applications. Useful to assess the role of spike timing or correlation between neurons by comparing info values obtained with these features preserved or shuffled

Table 2.1: Glossary of Main Information Theoretic Functions. This table reports a short explanation of what the implemented information theoretic quantities compute and for what type of applications they may be used. X, Y, Z denote random variables.

see [38]), which measure overall and stimulus-feature-specific information transmission between nodes of neural networks (TE.m, FIT.m and cFIT.m; Fig. 2.1D).

The information quantities depend on the probabilities of task variables (e.g. presented sensory stimuli) and neural responses. MINT implements the direct method [41, 59] estimator based on discretizing neural responses and task variable values and computing the empirical occurrences across experimental trials of the discrete or binned responses. These estimators have been widely used in neuroscience information theoretic studies, because neural spiking activity is intrinsically discrete and is usually quantified as the number of spikes emitted in one- or multiple-time windows of interest [41, 59]. The direct method captures the information carried by spike counts very precisely (Fig. A.3 in A). Because they are simple and do not make assumptions about the probability distributions, discretized estimators have been used to compute information also from continuous-valued aggregate measures of neural activity such as LFP, M/EEG, fMRI [56, 104, 105, 113] or continuous-valued behavioral variables [114]. If the scientific question at hand needs PID in addition to Shannon information and the data are not Gaussian, then discrete or discretized approaches are advised (as non-discrete non-parametric estimators are available only for Shannon information and entropy). MINT provides binning functions to discretize analogue data (equi-spaced or equi-populated binning, binning with user-defined bin edges, and possibly automated determination of bin numbers [115, 116]).

Any real experiment only yields a finite number of trials from which probabilities must be estimated. Finite sampling when using direct methods leads to a systematic error (bias) in information estimates (Fig. 2.1E), which can be as big as the true information values. Thus, bias corrections methods are essential for practical neuroscience applications, and six such well-established methods are included in MINT [39, 40, 42, 43, 117, 118]. These methods, along with binning, parallelization options and other features are user-specified in an input structure (opts). Information (function `MI.m`) is computed by default with the direct method, as it preserves all information available in the discretized neural activity. We recommend its use for small-dimensional (up to $N = 3$ or 3) neural response (e.g. responses of populations of up to 2-3 neurons) as its estimates from datasets of realistic sizes can be still effectively corrected for the limited-sampling bias (Fig. A.3 in A).

Alternatively, probability estimators suited for real-valued data [119–121], such as nearest-neighbors or kernel methods, can be used to estimate information and are available in MINT by specification in the input structure (`opts`). These methods also work well for low-dimensional data.

Neither these estimators nor the direct method, however, work on their own when considering high-dimensional neuronal responses (such as the activity of populations of many neurons), as the curse of dimensionality prevents the direct sampling of the joint response probabilities from high dimensional data (Fig. A.3 in A). We thus provide additional pipelines, recommended for high-dimensional neural responses such as the activity of large neural populations, that compute information from the empirical neural response probabilities but after reducing the dimensionality of neural population activity [35]. These dimensionality-reduction pipelines include supervised methods (Support Vector Machines, SVMs and Generalized Linear Models, GLMs) which reduce the di-

mensionality by providing decoding or posterior probabilities of the task variables given the single-trial neural population activity (Fig. 2.1A) and allow reliable estimations with small datasets (Fig. A.3 in A). We also provide unsupervised methods (Non-negative Matrix Factorization, NMF [122]; Principal Component Analysis, PCA) which reduce dimensionality individuating small numbers of dimensions with the highest explanation power of neural activity. Supervised dimensionality-reduction algorithms that individuate the directions in neural activity space with most discriminability of the task variable (e.g. SVM) may be in general better suited than unsupervised algorithms individuating dimensions that target best reconstruction of the spike trains (e.g. PCA, NMF) when the most information is not encoded in the direction with most variations in neural activity space (Fig. A.3 in A).

MINT provides all these dimensionality-reduction techniques with native MATLAB functions, but it also allows easy interfacing with external libraries (e.g. `libsvm` [123] and `glmnet` [124]) (Fig. A.2 in A). Importantly, these dimensionality-reduction tools can be coupled with MINT's Hierarchical Shuffling tool (`hShuffle.m`) which can disrupt, by trial shuffling, specific features of population activity (such as response timing or correlations between neurons) to probe their contribution to information processing [35, 125].

When deciding which estimator to apply to a given dataset, we recommend users to test different algorithms on synthetic data that match essential features of the experiments (e.g. discrete spike counts or continuous signals, number of trials and data dimensionality, information levels) and chose what suits best. MINT provides a simulator of neural population spike train activity (Poisson/non-Poisson both correlated or uncorrelated) that can be used for this purpose.

2.3 Results

We illustrate how to use MINT to address highly topical neuroscientific questions, emphasizing the utility of using synergistically multiple algorithms, allowed by MINT. In all examples, we use the limited-sampling bias corrections and hierarchical data shuffles of MINT, as they are essential for empirical data analyses.

2.3.1 Computing the role of interactions between neurons in information encoding

An important question in neuroscience is whether and how the functional interactions (measured as activity correlations) between neurons enhance or limit information encoding in neural populations [8, 53]. Several information theoretic methods have been developed to address complementary aspects of this question [30, 32–34, 63, 64, 107, 125, 126]. Here we illustrate what we gain from their combined use enabled by MINT.

We consider how a population of N neurons encodes information about a stimulus variable S. For neuron pairs ($N = 2$), we computed the population information (Mutual Information between stimulus and the joint neural population response) with the direct

method that estimates information directly form the empirical discretized response probabilities (see Design and implementation). The overall effect of interactions between neurons is expressed by the Redundancy-Synergy Index (RSI), the difference between the population information and the sum of single-neuron stimulus information [64, 127]. Positive (negative) RSI indicates predominantly synergistic (redundant) interactions. Contributions of synergy and redundancy can be separated using PID [109, 128]. The Information Breakdown [32–34] shows how RSI arises from interactions between neurons, by breaking down RSI into components $I_{\text{sig-sim}}$ (contribution of the similarity across neurons of trial-averaged responses to different stimuli, see also [64]), $I_{\text{cor-ind}}$ (contribution of the interplay between the signs of signal similarity and of noise correlations, defined as correlations between neurons in trials to the same stimulus), and of $I_{\text{cor-dep}}$ (quantifying information added by the stimulus-modulation of noise correlations, or, equivalently, bounding the information lost when using decoders trained without considering correlations [63, 108]).

These small-population direct information calculations have the advantage of not making assumptions about decoding mechanisms, but do not scale up to large populations because of the curse of dimensionality [40]. Population information can be obtained by estimating probabilities in the reduced space of the stimuli decoded from single-trial neural activity. These estimates scale well with population size and can be computed robustly with small datasets (Fig. A.3 in A). However, specific decoders may severely underestimate total information in neural activity (see Fig. A.3 in A), especially when the decoder does not operate on the features of neural activity that carry most information. We illustrate below how MINT allows determining the role of correlations in population coding by comparing decoders that do or do not use information in correlated activity and by leaving intact or removing information in correlated activity using hierarchical shuffling tools [35, 53, 125].

We illustrate these methods first by simulating the activity of $N = 20$ neurons responding to two stimuli. In the first simulated scenario (Fig. 2.2A), only correlations between activity of different neurons, but not the single-neuron activities, are stimulus-modulated and thus encode stimulus information. The single cell information is zero, but the pairwise population information is not. Positive RSI arises because of large synergy with negligible redundancy. The Information Breakdown reveals that all the synergistic information is due to stimulus-dependent correlations. Population decoding with SVM of the$N = 20$ neurons reveals that large-population information can be accessed exclusively with a non-linear decoder, and that shuffling correlations destroys all information, confirming it is exclusively encoded by correlations.

In the second simulated scenario (Fig. 2.2B), information is encoded by single cells, correlations are only weakly stimulus-modulated, all neurons have equal stimulus tuning (responding more strongly to stimulus 2), and noise correlations are positive. In this configuration, redundancy is created (all neurons have the same trial-averaged response profiles to the stimuli) and correlations reduce information (they are elongated along the axis separating the mean firing rates of individual neurons and thus increase the overlap between the stimulus-specific distributions of neural activity) [32]. Negative RSI arises

because of larger redundancy (created by so called signal similarity expressing the similarity of tuning to stimuli of individual neurons) than synergy (created by the small but present stimulus-modulation of correlations). Information Breakdown analysis reveals that indeed information is more redundant because the signal-noise similarity (captured by $I_{\text{sig-sim}}$ and $I_{\text{cor-ind}}$) is larger than the small stimulus-dependent correlations $I_{\text{cor-dep}}$. In the large $N = 20$ population most information can be accessed with a linear SVM, with the non-linear SVM adding relatively little, and noise correlations reduce information (shuffling them away increases information).

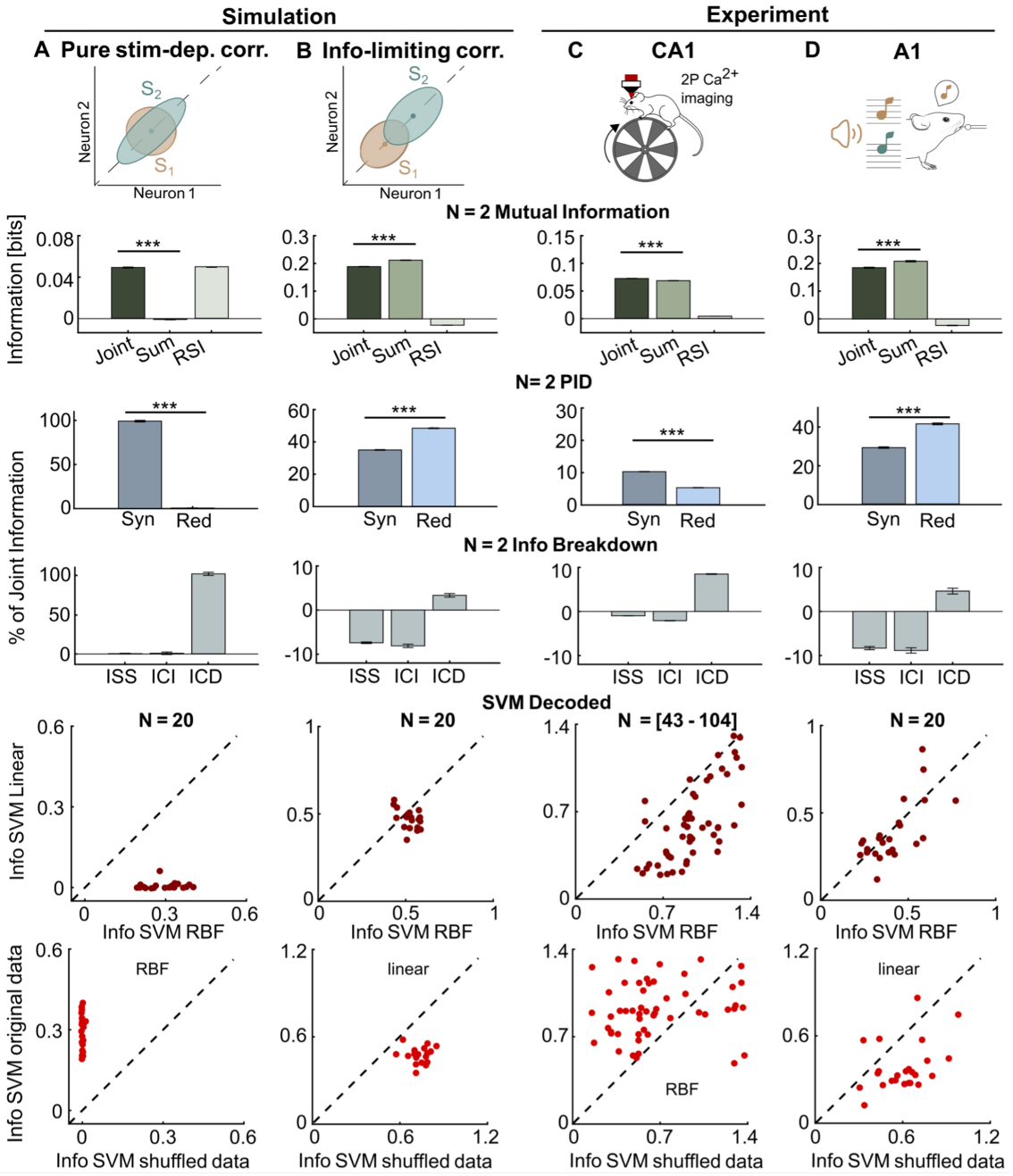

Figure 2.2: Assessing the role of correlations among neurons in neural population encoding. In each column, we consider analysis of a different dataset. **A:** simulated population of $N = 20$ neurons which carry information only by stimulus-dependent correlations, with no stimulus information provided by single-neuron firing rate modulation. **B:** simulated population of $N = 20$ neurons which carry information by single neuron firing modulations and which have information-reducing correlations. **C:** CA1 recordings of $N = 43 - 104$ neurons over $n = 11$ sessions during spatial navigation of a linear track in virtual reality. **D:** A1 recordings of$N = 20$ neurons over $n = 12$ sessions during tone presentation. In each row, we plot from top to bottom: direct calculation of information for neuron pairs and of sum of single neuron information; direct calculation of redundancy-synergy index (RSI), of synergy and redundancy separately and of the Information Breakdown components for neuron pairs; calculation of encoded information of the whole population using the information in the confusion matrix of an SVM decoder (linear or RBF), computed either on the real population responses (which contain correlations between neurons) or pseudo-population "shuffled" response obtained collecting randomly permuted trials to the same stimulus (shuffling removes correlations at fixed stimulus). In columns A-B we compute Shannon Information between neural activity and the identity of the two simulated stimuli. In column C-D we compute Shannon Information between neural activity and the identity of the presented tone ($S = 2$ different tones) or the spatial location of the mouse (binning locations into $S = 12$ equi-distant spatial bins), respectively. In column C, direct measures of pairwise information were obtained with $R = 2$ equi-populated bins (appropriate for this dataset consisting of non-deconvolved calcium fluorescent traces). In column D, direct measures of pairwise information were obtained with $R = 3$ bins, done by capping to 2 spike counts (appropriate for this dataset consisting of calcium signals deconvolved to estimate firing rates and activity counted in short windows). In each panel we plot mean and SEM (for simulated data in panel A-B: over $n = 190$ neural pairs and $n = 10$ simulation repetitions for the direct information calculations; over n=5 different data folds and $n = 10$ simulation repeats for the decoding information values; for CA1 data in Panel C: over $n = 10750$ simultaneously recorded neuron pairs for the direct information calculations, and over $n = 11$ recording sessions and $n = 5$ trial folds for the decoding information values; for A1 data in Panel D: over $n = 2280$ simultaneously recorded neuron pairs for the direct information calculations, and over $n = 12$ recording sessions and $n = 2$ trial folds for the decoding information values). Symbols *, **, *** denote two-tailed $p < 0.05$, $p < 0.01$, $p < 0.001$ respectively, computed with paired t-tests. See A.6.1, A.7.2 and A.7.3 in A for details of simulations and real data analysis. Mouse sketch in Panel D is modified from `doi.org/10.5281/zenodo.3925985`.

We then applied the same analyses to two real neural datasets. We first analyze encoding of the mouse position (within a linear track) by populations of 43-104 simultaneously recorded neurons from the CA1 region of the mouse hippocampus [69] (Fig. 2.2C). With the pairwise analysis, PID shows that both synergy and redundancy are present, but synergy is larger and the Information Breakdown shows that this is due to modulation of the noise correlations strength with the position ($I_{\text{cor-dep}} \approx 10\%$ of the pairwise information). Using a nonlinear decoder of the whole population increases information by $\sim$ 80% over what could be achieved with linear decoders, and shuffling data to destroy correlations decreases the nonlinearly decoded information by $\sim$ 80%, revealing a large effect of hippocampal noise correlations in position encoding by large neural populations, whose size could not be inferred by neuron pairs analysis.

We then analyzed encoding of sound intensity by populations of 20 neurons simultane-

ously recorded from the mouse auditory cortex (A1) during pure-tone sound presentation (Fig. 2.2D). These networks were selected, among all recorded neurons, based on their encoding of task-relevant information in [129]. With the pairwise analysis, PID shows that both synergy and redundancy are present, but redundancy is larger. Information Breakdown analysis shows that this is due to negative $I_{\text{sig-sim}}$ (neuron pairs have similar tuning to the stimuli) and $I_{\text{cor-ind}}$ (most neural pairs have also positive correlations), with $I_{\text{cor-dep}}$ contributing much less. Decoding whole-population activity with a nonlinear SVM did not increase the information decoded with a linear SVM (stimulus-dependent correlations were weak), and shuffling away noise correlations increases information substantially (thus correlations strongly reduced information).

Together, these results illustrate the power of combining MINT tools to understand deeply how interaction between neurons shape neural population coding.

2.3.2 Computing the Impact of Stimulus Information in Neural Activity for Behavioral Discrimination

Traditional approaches to neural information encoding of sensory stimuli have focused solely, as in the above examples, on how neurons or populations encode information about these stimuli. However, it could be that little or none of the information they encoded is actually utilized to inform behavior. It is thus important to have instruments to understand how much information in neural activity contributes to behavior.

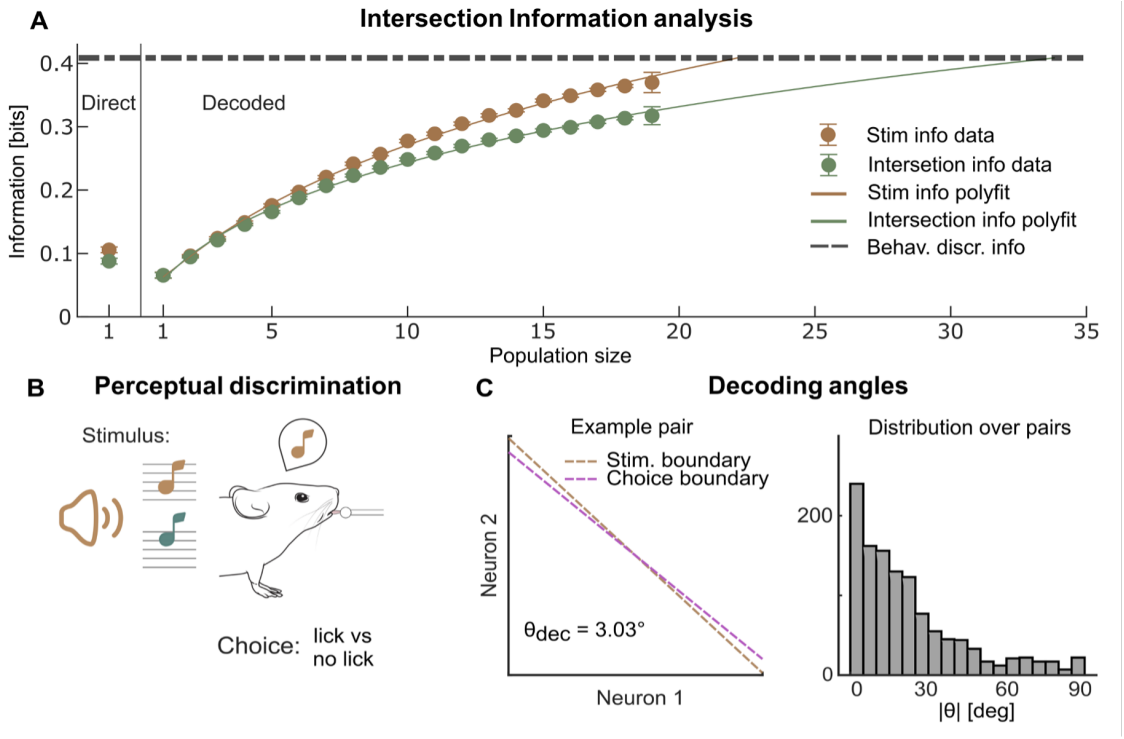

Figure 2.3: Stimulus, choice, and Intersection information coding in populations of cortical auditory neurons during a tone discrimination task. **A:** Stimulus information and Intersection Information encoded in neural activity recorded during a sound tone discrimination task. Left: single cell estimates using the direct method. Right: estimates of the information quantities using a RBF SVM (2-fold cross validation) as function of the population size. We plot the mean and SEM over all $n = 12$ Field of Views and over all folds and over all subpopulations used. For population sizes $N = 1 - 18$, more than 100 independent subpopulations can be obtained, we shortened computation time using only $n = 100$ randomly sampled subpopulations. For population size $N = 1$ and 19, we used all the $n = 20$ different subpopulations available. For the direct information calculation, we used 3 bins for 0 spikes, 1 spike and any value above 1. For all information analyses, we used the shuffle subtraction to correct for the limited-sampling bias. The dashed horizontal line plots the averaged information needed to explain behavioral discrimination accuracy (computed as the information between stimulus and choice). Full lines show log-polynomial fits to the dependence of stimulus and intersection information on population size. The population size with information sufficient to explain behavioral discrimination accuracy is the x-axis intercept of the point at which the fit lines cross. **B:** Schematic of the behavioral task in mice used when recording the data analyzed in this figure. **C:** Stimulus and choice boundary computed with MINT in the space of paired neural activity for one example neural pair in the dataset. The value of the angle between the two axes is reported in the inset. Right: distribution of the absolute value of the angle between the stimulus and choice boundaries for the $n = 2280$ neural pairs in this dataset. See A.6.1 and A.7.3in A for details of simulations and real data analysis. Mouse sketch is modified from `doi.org/10.5281/zenodo.3925985`.

Intersection Information (II) measures how much of the sensory information encoded in neural population activity is read out to inform behavior (Fig. 2.3A), and is computed with PID (using the tri-variate probabilities of stimuli, neural activity and behavioral choices) as the component of neural information that is both about stimulus and choice [35–37]. To demonstrate its use, we applied it to analyze the activity of populations of neurons recorded with 2-photon calcium imaging in mice in auditory cortex during pure-tone perceptual discrimination [129] (Fig. 2.3B).

We first considered information encoded by single neurons, computed with the direct method. If the readout of the stimulus information in neural activity was optimal (respectively, completely suboptimal), II would equal the stimulus information, (respectively, be zero). We found that for single neurons, II was $\sim 90\%$ of the total single-neuron stimulus information, showing that information encoded by these neurons is not read out optimally but still efficiently.

For sampling reasons explained above, the direct calculation of II can be done for small ($N = 1 - 3$), but not for large populations. How can we use II to address how information relevant to behavior scales with population size? Specifically, how large must a population be to account for perceptual discrimination ability? To answer this, in MINT we combined II with dimensionality-reduction techniques. In this application, we used an SVM to compress neural activity (using `svm_wrapper.m` before `II.m`). This compression loses some information (the information values obtained with the direct method are $\sim 20\%$ higher than the single cell values obtained with SVM decoders; Fig. 2.3A). However, II population information computed with SVM decoders are scalable and data-robust (Fig. A.3 in A). Computing how information scales with population size (Fig. 2.3A) shows

that as population size increases, the gap between stimulus information and II widened. This means that behaviorally-relevant information is more redundant across neurons compared to information that is not used to inform behavior, confirming the usefulness of redundancy for behavioral readout [130]. Had we considered only stimulus information, we would have incorrectly concluded that ~ 23 such neurons are sufficient to account for the mouse discrimination performance (Fig. 2.3A). However, taking intersection information into account reveals that ~ 34 such neurons are instead needed to fully account for the perceptual discrimination ability, as not all stimulus information encoded in neural populations is read out (Fig. 2.3A).

We endowed MINT with instruments to characterize neural mechanisms of readout. Suboptimality may arise because of a misalignment between how information is encoded and how the brain reads it out to inform choices [36]. MINT returns the axes in neural activity space trained to discriminate between stimuli and the axes trained to discriminate different choices (using `svm_wrapper.m`, see Fig. A.4 in A for examples on simulated data). Computing decoding angles of pairs of A1 neurons (Fig. 2.3C) shows that most pairs had a small but non-zero mis-alignment between stimulus and choice decoders, which explains the efficient but sub-optimal readout.

In sum, combining Intersection Information and dimensionality reduction can give precise insights about the behavioral relevance of information encoded by neural populations.

2.3.3 Mapping Content-Specific Encoding and Transmission of Information Within a Network

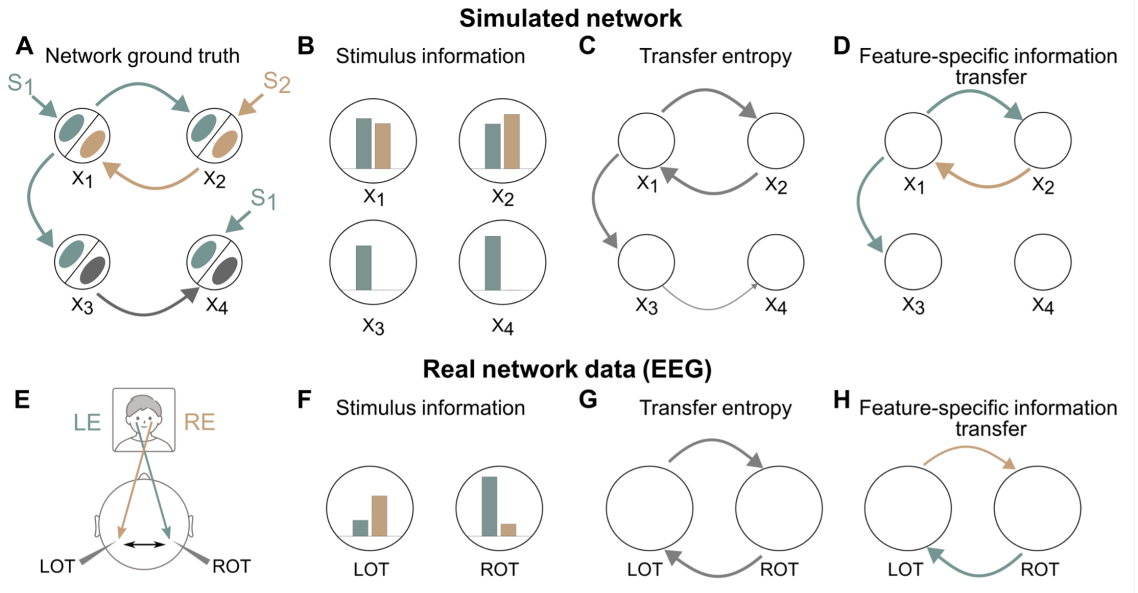

Figure 2.4: Reverse engineering information flow using stimulus-encoding and stimulus-transfer estimation algorithms. Panels A-D test MINT on simulated network data. **A:** Schematic of the simulation. The network comprises four neural nodes (black circles) $X_1, \ldots, X_4$, each containing two subpopulations (ellipses within the circles) encoding two independent binary stimulus features S_1, S_2. The ground-truth stimulus specific information communication is plotted in Panel A, with grey color used to indicate no stimulus selectivity, and green and brown colors used to indicate information selectivity to S_1 and S_2 respectively. **B:** Maximum Mutual Information across time between each neural population X_i and the stimuli S_1 and S_2. **C:** Transfer entropy (TE) between nodes. **D:** FIT about S_1 and S_2 between nodes. In panels C-D, only significant ($p < 0.01$, permutation test) links are plotted, with thickness proportional to the computed value. In each panel we plot the average information values across $n = 10$ simulation repeats. Panels E-F test MINT on real human EEG data. **E:** Schematic of the putative information flow interhemispheric information flow. LOT (ROT) denote Left (Right) occipito-temporal regions. LE (respectively RE) denote the Left (respectively Right) Eye face visibility feature. **F:** Maximum Mutual Information across time about the left or right eye visibility present in left of right OT region. **G:** Significant transfer entropy between LOT and ROT brain regions. **H:** Significant FIT between LOT and ROT brain regions. In panels G-H, only significant ($p < 0.01$, permutation test) links are plotted, with thickness proportional to the computed value. In each panel we plot the average information values across $n = 15$ experimental subjects. See A.6.3 and A.7.1 in A for details of simulations and real data analysis. Human face sketch is modified from `svgrepo.com/svg/493087/men-in-their-20s-and-30s-face` and brain sketch is modified from `svgrepo.com/svg/83465/brain`.

MINT provides both algorithms to study information encoding in individual network nodes and information transmission across nodes. We here illustrate how to combine them for reverse-engineering the information flow within neural networks.

We first simulated a network with four nodes $X_1, \ldots, X_4$ each modeling the aggregate activity of a brain area (as e.g. measured by aggregate neural signals such as LFPs, M/EEG or fMRI, see SM 6.3 in A). This network has a well-defined ground-truth flow of information about two independent stimulus features S_1 and S_2 (Fig. 2.4A). Information about S_1 is received from the outside by nodes X_1 and X_4 in a short time window (3-12 ms from simulation start for X_1 and 15-24 ms for X_4), and is then sent from X_1 to X_2 and X_3 with a 5 ms delay. Information about S_2 is received (in the 3-12 ms window) from the outside by X_2 which then sends it to X_1 with a 5 ms delay. Nodes X_3 and X_4 exchange information (also with a 5 ms delay) which is not about S_1 or S_2. To disentangle the information flow, we computed (using the direct method) information encoded or transmitted at each time (in Fig. 2.4 we plot for each node and link the maximal information values over time, but we show in Fig. A.5 in A that time-resolved analysis reconstructs correctly the ground-truth information encoding windows and communication delays), and we used MINT's non-parametric permutations tests to identify significant encoding or transmission. Using Mutual Information between individual stimulus features and individual node activity reveals correctly that all nodes have information about S_1 and that only X_1 and X_2 have information about S_2 (Fig. 2.4B). To study how this information is exchanged within the network, we first computed overall information transfer with Transfer Entropy, finding correctly significant transfer from X_1 to X_2 and X_3, from X_2 to X_1, and from X_3 to X_4 (Fig. 2.4C). To reveal the information content of this exchange we com-

puted Feature Specific Information Transfer (FIT), revealing correctly that the information transferred from X_1 is about S_1 but not about S_2, and that the information transferred from X_2 is about S_2 (Fig. 2.4D). FIT finds no information transfer from X_3 to X_4 about S_1 or S_2, thus determining correctly that the overall information transfer from X_3 to X_4 detected with TE is not about any of the two stimulus features. Finally, the finding that X_1 and X_4 encode information about S_1 while they do not receive it from other network nodes implies that X_1 and X_4 receive external S_1 information. Similarly, because X_2 encodes information about S_2 while not receiving within-network S_2 information demonstrates that X_2 receive external S_2 information. Thus, combining encoding with transmission analyses could correctly reverse engineer the within-network specific information flow.

We next tested how MINT reverse-engineers information flow in real brain networks by applying it to an existing EEG dataset recorded from human participants detecting the presence of either a face or a random texture from images covered by random bubble masks [131]. Prior work [38, 131, 132] revealed that the visibility of the eye region (proportion of visible pixels in the eye area) is critical for successful face discrimination and that the Occipito-Temporal (OT) EEG electrodes are those encoding most Mutual Information about both left and right eye visibility (Fig. 2.4E,F). To understand if some of this information was exchanged across the OT regions in different hemispheres, we used TE and FIT to analyze transmission of left or right eye visibility information across OTs. TE across hemispheres was found in both directions (right-to-left and left-right), suggesting a bi-directional inter-hemispheric communication (Fig. 2.4G). However, specific information transfer was precisely directional: FIT about the left eye was only from right-to-left and FIT about the right eye was only from left-to-right (Fig. 2.4H). Thus, using MINT allowed establishing encoding and directional transfer of different eye features across hemispheres with high specificity. These analyses could also temporally localize both encoding and inter-hemispheric transfer (Fig. A.6 in A).

Together, these results illustrate the power of combining MINT tools to reverse-engineer encoding and flow of specific information across brain networks.

2.4 Availability and Future Directions

MINT is downloadable in source code (`github.com/panzerilab/MINT` with DOI `10.5281/zenodo.13998526`), including a Dockerfile, and is licensed under GNU GPLv3. It contains documentation on using it and on building and installing it from source, unit tests, use examples, and replication of paper figures (`github.com/panzerilab/MINT_figures`).

The modularity of MINT allows it to be used alongside any other MATLAB function or toolbox. As exemplified above, we already provide pipelines for interfacing with decoding toolboxes. We plan to add plugins to generate neural and behavioral data from data acquisition and preprocessing toolboxes (e.g. [133]) with MINT's input-data format requirements, and to generate MINT's outputs suitable to be fed directly into toolboxes for further advanced analyses, e.g. for network analysis of information-transfer outputs [134].

We plan to further extend the range of information-theoretic methodology implemented in MINT. MINT's current version emphasizes discretized maximum likelihood estimators. However, we provide only a handful of data-discretization techniques that go with it. We plan to endow them with optimal discretization algorithms based on model selection techniques (Akaike and Bayesian information criterions). While MINT already implements a number of probability estimators for real-valued data we plan to extended them to include other binless and kernel-based estimators [135, 136], and parametric probability models (Gaussian, Poisson) proposed in the neuroscience literature. Although we provide several tools for assessing the role of correlated activity, we plan to implement currently missing Maximum Entropy estimators [103]. Finally, the derivation of new neuroscience-related information quantities with PID is highly active [109, 137] and the open source and modularity of MINT will allow rapid integration of new developments.

A limitation that may restrict MINT's usage is that it is developed only in MATLAB at this stage. We are thus developing a translated python version of MINT to widen usage. However, we verified that MINT is usable from Python using the MATLAB Engine API for Python and we provide instructions in A.2 in A.

Chapter 3

Sampling bias corrections on redundant, unique, and synergistic information

The content of this chapter was submitted for publication, and is currently under submission and being revised [138]. The analyses presented here correspond to those presented in the first submitted version of the paper prior to being revised.

3.1 Introduction

It is widely believed that the brain is a complex system and that behavior emerges from the organized pattern of the interactions between the brain's computing elements - the neurons [139]. Because of this, the study of how interactions between neurons shape information processing has fascinated computational and empirical neuroscientists for decades [8, 53, 64, 107, 140]. Information theory has been a prominent tool in this research as it is uniquely positioned and is a natural choice for investigating neural information processing [6, 25, 26]. Shannon information captures all ways in which systems carry information, it is highly general and applicable regardless of the type of noise and statistics, and is thus equally applicable across species and recording modalities and to real data and in silico models. While earlier work has concentrated on understanding whether correlations between the activity of different neurons increase or decrease information [9, 33, 64, 107], recent advances in information theory based on Partial Information Decomposition (PID) [30] have enabled neuroscientists to formulate more precise questions [137] about the unique information carried by each individual neuron, the synergistic information generated by neural interactions (that is, new information that emerges specifically from and is found only in the interactions between neurons) or redundant information present when different neurons share the same information.

Earlier applications of Information Theory to neuroscience have recognized that Shannon Information measures from neural activity suffer from a prominent limited sampling bias [42, 43, 121]. This bias does not only affect the precision of the estimate, but can

also skew comparisons between the information carried by simpler (less biased) and more complex (more biased) neural representations. The information theoretic neuroscience literature has provided effective tools for correcting for the Shannon information sampling bias [40]. However, to date the problem of whether synergy, redundancy and unique information components of PID are biased has not been studied systematically. To our knowledge, only one study investigated PID sampling bias [45]. This method was valid only for Gaussian distributions, which do not apply easily to the discrete spiking activity of neurons. Here, we study the sampling bias of discrete PID estimators suitable to study the spiking activity of neurons. We found that the PID components are unevenly biased, with synergy far more biased than any other component. We study and provide an understanding of the origin and properties of the bias, and based on this we provide methods to reduce this problem effectively that provide more accurate measures than the state of the art. Finally, we test and apply our method to 53117 pairs of neurons simultaneously recorded from the mouse cortex and the hippocampus.

3.2 Background: short introduction to PID

For completeness, we first summarize the concepts of PID that we use. PID decomposes the information jointly carried by a set of source variables (for us, a set of simultaneously recorded neurons) about a target S (for us, a sensory stimulus) into non-negative components that capture information about the target that is either redundantly encoded across sources, uniquely encoded by a single source or synergistically encoded by the combination of sources. In this paper, we will focus on the case of two source variables, which has received the most attention in the PID literature and in real data applications, and which has the most established theoretical foundations [31, 45, 78, 110, 129, 141, 142]. We focus our presentation on the information about sensory stimuli carried by neurons, but this framework straightforwardly extends to information carried by neurons about other quantities, such as cognitive or motor variables or the activity of other neurons. We consider the Shannon information $I(S;R_1,R_2)$ about an external stimulus S carried jointly by the neural spiking activity R_1 and R_2 of two simultaneously recorded neurons and the Shannon information $I(S;R_i)$ that each of the two neurons ($i = 1,2$) carries about S. $I(S;R_1,R_2)$ and $I(S;R_i)$ are computed from the probability distributions $p(S,R_1,R_2)$ and $p(S,R_i)$ ($i = 1,2$) [21, 143], as follows:

$$I(S;R_i) = \sum_{s\in S, r_i \in R_i} p(s,r_i) \log \frac{p(s,r_i)}{p(s)p(r_i)} \tag{3.1}$$

$$I(S;R_1,R_2) = \sum_{s\in S, r_1 \in R_1, r_2 \in R_2} p(s,r_1,r_2) \log \frac{p(s,r_1,r_2)}{p(s)p(r_1,r_2)} \tag{3.2}$$

PID decomposes the joint and single-neuron information into four non-negative components that satisfy the following linear relationships [30]:

$$I(S;R_1,R_2) = RI(S:R_1;R_2) + UI(S:R_1 \setminus R_2) + UI(S:R_2 \setminus R_1) + SI(S:R_1;R_2) \tag{3.3}$$

$$I(S;R_1) = RI(S:R_1;R_2) + UI(S:R_1 \setminus R_2) \tag{3.4}$$

$$I(S;R_2) = RI(S:R_1;R_2) + UI(S:R_2 \setminus R_1) \tag{3.5}$$

where in the above equations $RI(S:R_1;R_2)$ is the redundant (or shared) information that both R_1 and R_2 encode about S; $UI(S:R_1 \setminus R_2)$ and $UI(S:R_2 \setminus R_1)$ are the unique information about S provided by one neuron but not by the other; and $SI(S:R_1;R_2)$ is the synergistic information about S encoded by the combination of R_1 and R_2. (We will sometimes shorthand these components as RI, UI_{R_1}, UI_{R_2}, and SI.) Eq. (3.3-3.5) mean that all 4 components contribute to the joint information, whereas only the information that a source carries uniquely and the information that is redundantly carried by both contribute to single-neuron information. Because the 4 PID components satisfy 3 linear constraints (Eqs. 3.3-3.5), determining one component is sufficient to compute the other three. We provide explicit equations of all components as function of synergy in Eqs (B.5).

Several definitions of PID components have been proposed [144, 145], satisfying desired properties including non-negativity of each component and symmetry of RI and SI under permutation of R_1, R_2. We will mostly use the BROJA definition [31], as it satisfies many desirable properties including additivity of RI, SI, and UI for independent systems of sources and targets [111, 145] and has been extensively applied to neural data [77, 129, 142, 146]. The BROJA defines the Union information, that is the target information in the joint source space that cannot be possibly attributed to synergistic interactions, or equivalently the total target information that can be extracted from a single source, as:

$$Union(S:R_1;R_2) = \min_{q \in \Delta_P} I_q(S;R_1,R_2) \tag{3.6}$$

where Δ_P is the set of all joint probability distributions $q(S,R_1,R_2)$ that have the same pairwise marginals $q(S,R_1) = p(S,R_1)$ and $q(S,R_2) = p(S,R_2)$ as the original distribution $p(S,R_1,R_2)$, and $I_q(S;R_1,R_2)$ is the joint information computed for distribution $q(S;R_1,R_2)$. Then, the synergy is defined as the difference between the joint and the union information:

$$SI(S:R_1;R_2) = I(S;R_1,R_2) - Union(S:R_1;R_2) \tag{3.7}$$

Other PID definitions, which were also successfully applied to neuroscience [38, 76, 78] but do not satisfy additivity include the originally proposed I_{min} [30] and the minimum mutual information I_{MMI} [110]. I_{min} quantifies RI as the similarity between R_1 and R_2 in discriminating individual values of S, while I_{MMI} quantifies RI as the minimum between the mutual information individually carried by R_1 and R_2, thus capturing only the amount but not the content of information carried by each neuron [145].

3.3 Background: discrete estimators of information and PID in neuroscience

Neural spiking activity is an intrinsically discrete variable. Thus, in most information theoretic studies of neural spiking activity, the responses have been treated as discrete

variables. When focusing on spike count codes, the neural response is described by the number of spikes emitted by the neuron in the time window of interest [102, 103, 126, 147–150]. When investigating if the timing of spikes encodes additional information above and beyond that present in the spike counts, then the most established approach [41, 55, 59, 151, 152] is to discretize the neural response time window of interest into a number of small time bins and then turn the spike train into a binary word. Importantly, often neural responses are sparse and information is encoded by relatively low spike numbers [61, 148]. In such cases, as we show in Fig. B.12, the Gaussian approximation to the information is highly inaccurate. Because it is simple and does not require assumptions on the probability distributions, discretization of neural responses for computing information has been used to compute information also from continuous-valued non-spiking aggregate measures of neural activity such as LFP, EEG, or fMRI [56, 57, 104, 105]. In addition, often only a discrete number of different stimulus conditions is presented in an experiment, and thus the stimulus S is typically a discrete variable. Under such conditions, the information theoretic quantities can be computed by simply estimating the probabilities by the empirical occurrences across experimental trials (maximum likelihood estimators) and plugging them into the information equations. This discrete information approach, which we call the plugin approach, has been extensively used in neuroscience for both Shannon information [34, 35, 41, 59, 69, 77, 102, 103, 107, 126, 149–151, 153, 154] and PID [35, 38, 57, 76–78, 129, 146, 155–158]. This discrete PID approach has also been used extensively across fields of biology and applied sciences [159–162].

3.4 Numerical investigation of the bias of individual PID components for discrete estimators

Calculation of information requires accurate estimation of the stimulus-response probabilities. With an infinite amount of data, the true stimulus-response probabilities could be measured exactly. However, any real experiment only yields a finite number of trials from which probabilities must be estimated. The estimated probabilities have finite sampling fluctuations around their true values (Fig. B.2) which lead to both systematic error (bias) and statistical error (variance) in estimates of information (Fig. B.2 and Supplemental Material, SM Section B.3). While variance can be reduced by averaging (e.g. across experimental subjects or groups of neurons), the bias cannot.

The sampling bias properties of discrete estimators have been extensively studied for Shannon information [40, 42, 43], but not to our knowledge for discrete PID estimators. To document them for PID, we simulated spike count responses of a pair of individual neurons in response to a set of $S = 4$ different stimuli. We studied how the estimate of each PID component depends on the number of available trials. We focus the presentation on the BROJA PID decomposition [31], but we confirm in Fig. B.6, B.7, B.10, B.11 that similar results apply to I_{min} and minimum mutual information I_{MMI}.

We developed three different scenarios with varying degrees of synergy and redundancy. In each case, the spike count r_i ($i = 1,2$) of each of the two simulated neurons

for each stimulus was the sum of two Poisson processes. One Poisson process, that was independently drawn for each neuron, expressed the variability of responses "private" to each neuron. Another Poisson process was shared between the two neurons and gave rise to across-neuron correlations. Within each scenario, we also varied the overall level of information, because previous studies showed that the bias of Shannon information depends on it [40]. In our simulations, the parameter B expressed the baseline level of activity; parameter α regulated the strength of stimulus tuning of each neuron (increasing α increased single-neuron information); parameter β regulated the dissimilarity of tuning between neurons (smaller β meaning more independent tuning); a parameter γ regulated the strength of the shared process. The overall information level was increased by increasing α or γ or reducing B. Redundancy was increased by increasing β. Synergy was increased by increasing γ.

In Fig. 3.1 we plot the values of the plugin estimates of joint information and of the PID terms as function of the simulated number of trials per stimulus, averaged over all $n = 96$ repetitions of the simulations with the considered number of trials. In these simulations we discretized the spike counts of each neuron into 4 equipopulated bins (leading to 16 possible discrete joint responses).

The limited sampling bias in the information estimates can be visualized by comparing the average value of information obtained with a given number of trials with the asymptotic value obtained with the largest number of trials. In all scenarios, the plugin estimate of the joint Shannon information was biased upward and the bias decreased with the number of trials, as reported previously [40, 41]. Here, we focus on the bias of the estimates of the PID quantities. We found several highly consistent and important results. First, as for Shannon joint information, PID quantities were biased upward, with the bias decreasing smoothly with the number of trials. Second, for the tens of trials usually available in real experiments, the bias could be as large, or even larger, than the target information quantities, meaning that the bias must be corrected for real data analyses. Third, and unlike what was assumed in previous studies [45], the bias is highly uneven across PID components. The synergy was by far the PID component with the largest upward bias. Its bias was lower than but comparable to that of the joint information. Unique information was also biased upward, albeit much less so than the synergy. Redundancy was almost unbiased. This is important because it shows that conclusions taken from limited empirical data without considering the bias will produce estimates artificially biased toward synergy. For example, in simulations with ground-truth values of redundancy larger than synergy, we would have incorrectly estimated synergy larger than redundancy for lower number of trials due to the sampling bias. Fourth, the bias was larger, both proportionally and in absolute terms, for lower information levels.

In SM Section B.8 we report analytical approximations to the sampling bias that allow an intuitive understanding and support the generality of these findings. The main take-home message from these calculations is as follows. For large enough numbers of trials, the bias can be expanded in inverse powers of the number of trials N, accounting for the smoothly decreasing bias with the number of simulated trials. The spurious levels (upward bias) of information is due to the fact that random fluctuations in probabilities make

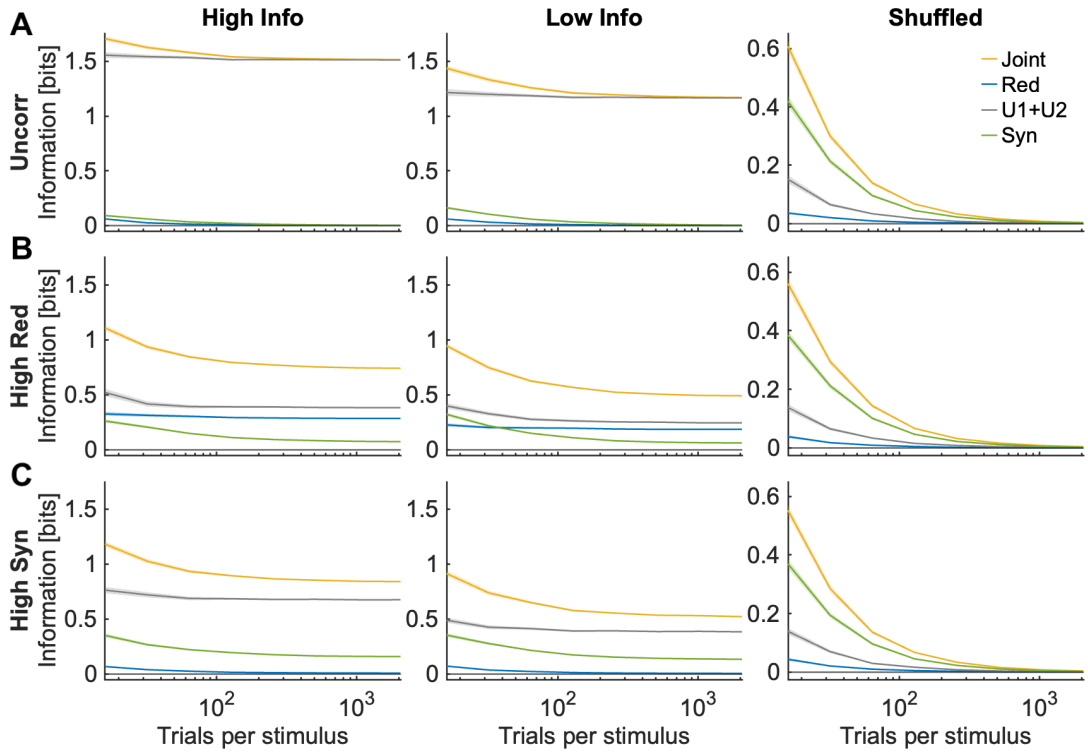

Figure 3.1: Joint information and PID quantities as a function of the number of simulated trials used to compute them. Top, central and bottom rows plot the simulated scenarios with no interaction, high redundancy and high synergy, respectively (see SM Section B.5). Left, center and right columns represent simulations with higher information ($\alpha = 10$), lower information ($\alpha = 7$) and with shuffled low-information data. "Syn": synergy. "Red": redundancy. "U1+U2": sum of the two unique information of each neuron. Here we used the plugin method without bias corrections. We used $R = 4$ discretization bins for each neuron (Table B.1). Each panel plots mean ± 2 SEM over $n = 96$ simulations.

the probabilities more different across stimuli than they actually are. At fixed numbers of trials, the bias induced by these fluctuations depends only on the number of possible responses and the bias is larger for larger number of discretized possible responses. The main term for the synergy bias is due, like for the joint information, to fluctuations in the joint probability $P(r_1, r_2, s)$, which is more undersampled than the marginal probabilities. The main bias term for the unique information originates from and is explained by fluctuations in the marginal probabilities, which are smaller because the marginal probabilities are defined in a smaller single-neuron space and are easier to sample. As a result, the bias of the joint information and of the synergy increases quadratically with the number of single-neuron discrete responses, whereas the bias of the unique and redundant information increase linearly or sublinearly (as found numerically, compare Fig. 3.1, B.4, B.5). Low information levels typically correspond to stimulus-specific distributions with a larger number of possible responses.

3.5 Correcting for the PID limited-sampling bias

Having documented and understood the properties of the bias of the PID, we now use this knowledge to propose and evaluate methods for bias correction.

Because the PID bias decreases smoothly with the number of trials, and because analytical calculations show that when the number of trials is sufficiently large, the bias depends polynomially on the inverse of the number of trials $1/N$, we extend the Quadratic Extrapolation (QE) procedure originally proposed in Ref [41] for Shannon information to correct for the bias of each PID term. We recomputed plugin PID terms using half and a quarter of the available data, we fitted these information values to a polynomial in $1/N$ and we used the best-fit coefficient to estimate and remove the bias. The QE bias correction substantially improved the estimates of all PID quantities. While plugin bias-uncorrected estimates of synergy needed large numbers of trials per stimulus ($N_s \approx 512 - 1024$ trials for 16 joint response bins (Fig. 3.1), corresponding to 32-64 trials per stimulus and joint response bin when R is varied, (Fig. B.4, B.5)), the QE reached accurate estimates with small residual bias with almost an order of magnitude less trials ($N_s \approx 64 - 128$ trials for 16 joint response bins, corresponding to 4-8 trials per stimulus and joint response bin (Fig. 3.2, 3.3, B.8, B.9)). The QE-corrected PID values are relatively accurate but not conservative because they have an upward residual bias (because information bias terms of higher order in $1/N$, not fitted in the QE, are all positive [163]).

We then introduced a second PID bias correction ("shuffle-subtraction" bias correction) that subtracts from the plugin value of each term the value of the same term obtained after randomly shuffling the stimulus-response association. In the shuffled data, all information about the target (the stimulus) is destroyed. Thus, the plugin shuffled PID values can be taken as a bias estimate because they should be zero for infinite trials. Using this correction also increased the precision of the estimates compared to the plugin values (Fig. 3.2, B.8). Importantly, and as supported by the analytical bias expansion (B.8), because lower information levels have larger upward bias, we observed a higher bias in the shuffled data than in the corresponding unshuffled data (Fig. 3.1). Thus, the shuffle-subtraction correction provides conservative estimates and can be used to lower-bound the PID estimates.

Given that the QE and shuffle-subtraction provide positive and negative residual PID bias respectively, we introduced a third bias correction procedure ("QE with shuffle-subtraction") which combines the two operations. This procedure gave highly accurate results over the entire range of trials tested. Because the shuffle-subtraction provides conservative estimates, this third procedure was more conservative than the pure QE and it gave most often (though not always) negative residual bias.

Importantly, when there were enough trials for the bias corrections to work well ($N_s \approx 64 - 128$ trials for a joint distribution for 16 response bins, corresponding to 4-8 trials per stimulus and joint response bin when R is varied, see Fig. 3.2, 3.3, B.8, B.9), the QE and the QE with shuffle-subtraction gave nearly identical estimates and the pure shuffle-subtraction gave only slightly conservative and accurate estimates, suggesting that comparing on real data several bias correction methods on the same data may be beneficial

for gaining confidence on the accuracy of the estimates.

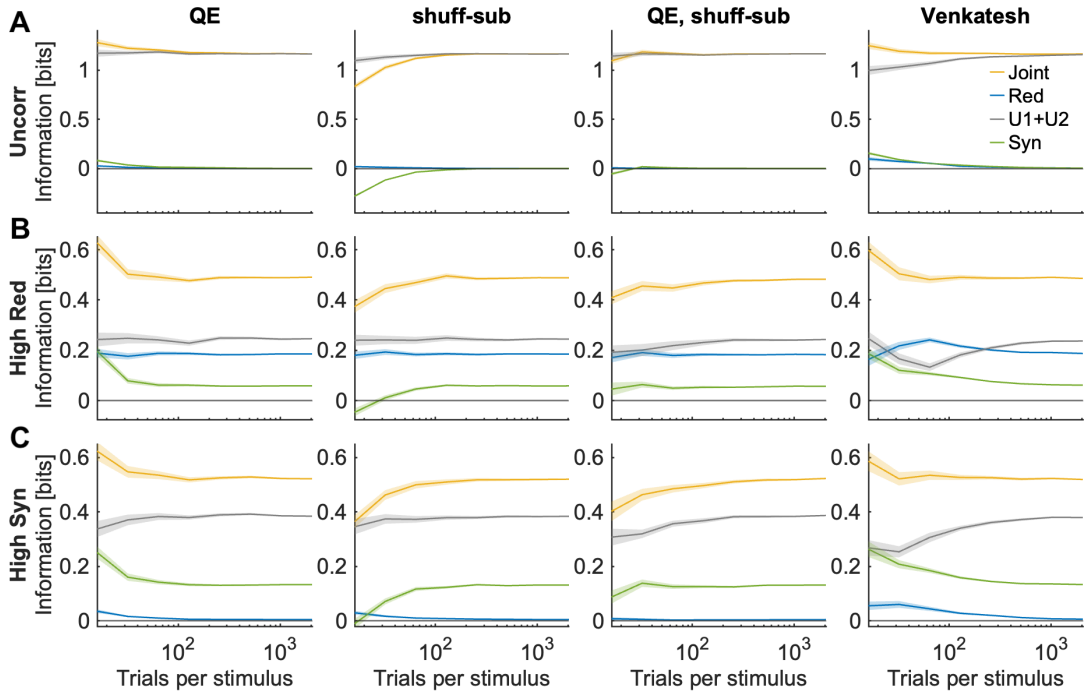

Figure 3.2: Performance of bias corrections with 4 discretization bins for each neuron. Joint information and PID quantities as a function of the number of simulated trials used to compute them. Top, central and bottom rows plot the simulated scenario with no interaction, high redundancy and high synergy, respectively (see SM Section B.5). Left to right columns report results of the QE, shuffle-subtraction, QE with shuffle-subtraction, and Venkatesh procedures, respectively. In each panel we plot the mean ± 2 SEM over $n = 96$ simulations.

The only bias correction which was proposed so far for PID was the one we term Venkatesh correction proposed in Ref [45]. It assumes that the Union information has the same bias as the joint information. Then it rescales all PID accordingly and implements post-hoc rectifications to make sure the PID terms are non-negative and still respect the PID properties of Eq. (3.3-3.5). Because, as we demonstrated above, the union information is much less biased upward than the joint information, this procedure leads to a major underestimation of union information, which then leads to major overestimations of synergy and redundancy that are present also for relatively large numbers of trials. We thus do not consider this correction further.

There are other bias corrections in the neural literature of discrete estimation of Shannon information [42, 43, 121, 164]. We did not consider them here because their derivation has not been extended to PID and because they performed worse than or equal to the QE with discrete estimators of Shannon information when tested on realistic simulations of neural population activity [40].

Although in the above we simulated S as a sensory stimulus, in many neuroscience applications S is the activity of other neurons. Our asymptotic expansions and considerations are valid as long as S is a discrete variable. In SM Section B.11 and Fig. B.13

we show that the bias exists also when the target variable S is the discretized activity of another neuron and we show that the bias corrections are effective also in that case.

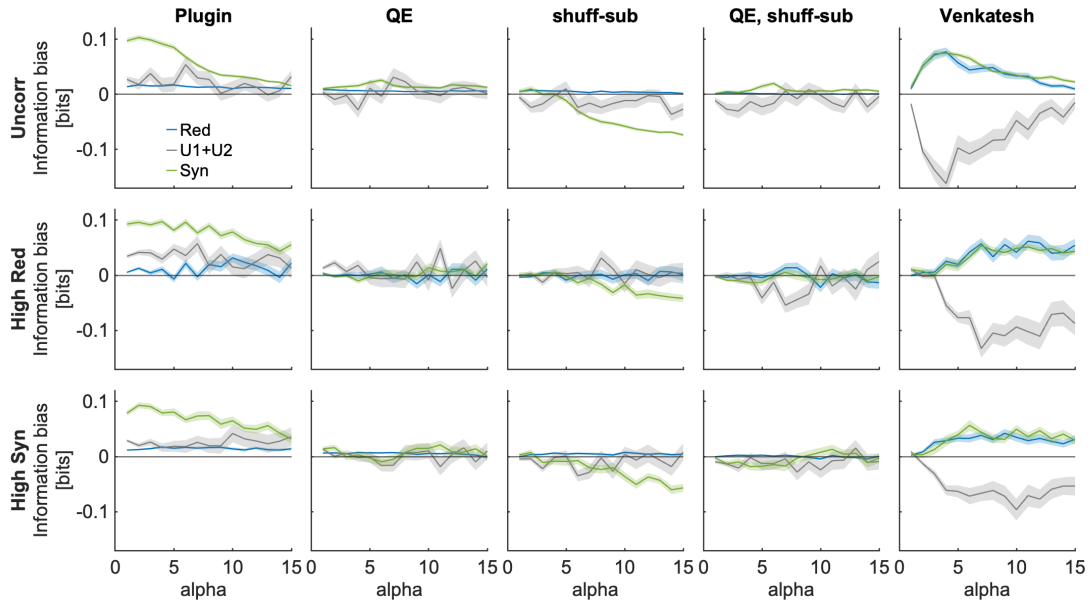

Figure 3.3: Performance of bias corrections using $N_s = 64$ trials per stimulus with 4 discretization bins for each neuron. In each panel we plot (rather than the information component value itself) the information component bias (computed as the information component estimated with the considered number of simulated trials minus the asymptotic information component estimated using the largest available number of simulated trials, that is 2048 trials per stimulus) as a function of the parameter α increasing single-neuron information in the simulated data. Top, central and bottom rows plot the simulated scenario with no interaction, high redundancy and high synergy, respectively (see SM Section B.5). Left to right columns report results with plugin estimators and with the QE, shuffle-subtraction, QE with shuffle-subtraction, and Venkatesh procedures, respectively. In each panel we plot mean ± 2 SEM over $n = 96$ simulations.

3.6 Evaluation of bias correction procedures on real neural data

We evaluated the utility of the PID bias correction using 3 datasets from previously published studies recording simultaneously with two-photon calcium imaging the activity of many neurons from the brain of mice performing cognitive tasks.

The first dataset consisted of $n = 6209$ pairs of neurons simultaneously recorded from auditory cortex during a sound intensity discrimination task [129]. We computed the information that the neurons carry about the sound intensity (a binary $S = 2$ stimulus set consisting of high vs low tone intensity). The activity of each neuron was first deconvolved to estimate the time-localized spiking activity from the calcium fluorescence signal imaged from each neuron. To compute the neural response variables r_1 and r_2 that

enter the information calculations, for each neuron we summed the deconvolved activity within a 333-ms time window centered around the time of maximal information, and we discretized this signal into $R = 3$ bins (see SM Section B.10 for full details).

The second dataset [35] consisted of $n = 10750$ pairs of neurons simultaneously recorded from posterior parietal cortex (PPC) during a sound localization task in which mice reported perceptual decisions about the location (left or right of the midline) of an auditory stimulus while navigating through a visual virtual reality T-maze (Fig. 3.4B). Because PPC is an area involved in converting sensory information into perceptual decisions, we computed the information that the neurons carry about whether the sound came from left or right of the midline ($S = 2$ stimuli), corresponding to the sound location categorization that the mouse had to perform to turn toward the reward location. To compute the neural response variables r_1 and r_2 used for information calculations, for each neuron we summed the deconvolved activity within 320 ms time windows centered around the time of maximal information and we discretized this signal into $R = 3$ bins (see SM Section B.10).

The third dataset consisted of $n = 36158$ pairs of neurons simultaneously recorded from the CA1 region of the hippocampus [69] while mice navigated a linear track in virtual reality (Fig. 3.4B). Because the hippocampus encodes position in space, we computed the information that the neurons carry about the spatial location along the linear track (the location was discretized in $S = 12$ spatial bins). The neural response variables r_1 and r_2 used to compute information were the activity of the neurons in one imaging frame (333 ms) when the mouse was in a given position. As the slow kinetics of the calcium indicator and the slower imaging frame rate made it difficult to deconvolve the calcium fluorescence traces to estimate spiking activity, following Ref. [69] we discretized the $\delta F/F$ calcium traces into $R = 2$ equipopulated bins (low and high activity) (see SM Section B.10).

We computed the joint information between the above-defined activity r_1, r_2 of simultaneously recorded pairs of neurons and the above defined stimulus s. We broke up this information into PID components using BROJA PID [31, 91]. We computed all 4 PID terms in Eq. (3.3). However, we focus the analysis and the neuroscientific interpretation on the differences between synergy and redundancy, to understand how these two emergent properties shape neural population coding in different brain regions. To test and exemplify the use of the bias-corrected PID algorithms, we compared (Fig. 3.4) the plugin estimates of PID with those obtained after applying the 3 bias corrections that we developed.

As shown by the reduced values obtained after applying the bias corrections, the plugin joint information and synergy were biased upward (consistent with simulations and theory). Despite the good number of trials available in these experiments, the synergy bias was substantial. Using the uncorrected plugin estimator would have led to a considerable overestimation of the bias and to a qualitative change of results in two datasets. In the auditory dataset (Fig. 3.4A), because of the large synergy bias, the uncorrected plugin estimator could not detect a significant difference between synergy and redundancy, whereas all 3 bias-corrected estimates consistently detected with high significance that redundancy was higher than synergy. In the hippocampal dataset (Fig. 3.4B), both the

plugin and the 3 bias-corrected estimates reported a higher level of synergy than redundancy. However, the use of the plugin bias-uncorrected method would have overestimated by 211% the amount of synergy with respect to what was consistently found with the bias corrections. The PPC data (Fig. 3.4B) gave consistent values of higher redundancy than synergy with all methods.

Another important result is that, reassuringly, we found highly consistent estimates of both redundancy and synergy across bias correction methods. Redundancy was essentially identical across methods for each dataset. Synergy estimates obtained with QE and QE with shuffle subtraction were within 3% of each other in each dataset, suggesting that these estimates on real data are precise and unbiased.

The more conservative shuffle subtraction underestimated (only slightly for the auditory and hippocampal dataset, and a little more for the PPC dataset) the less conservative synergy values obtained with QE or QE with shuffle subtraction. However, and importantly, the fact that the synergy obtained with the conservative shuffle subtraction is positive proves that there is genuine synergy that cannot be due to finite sampling artifacts.

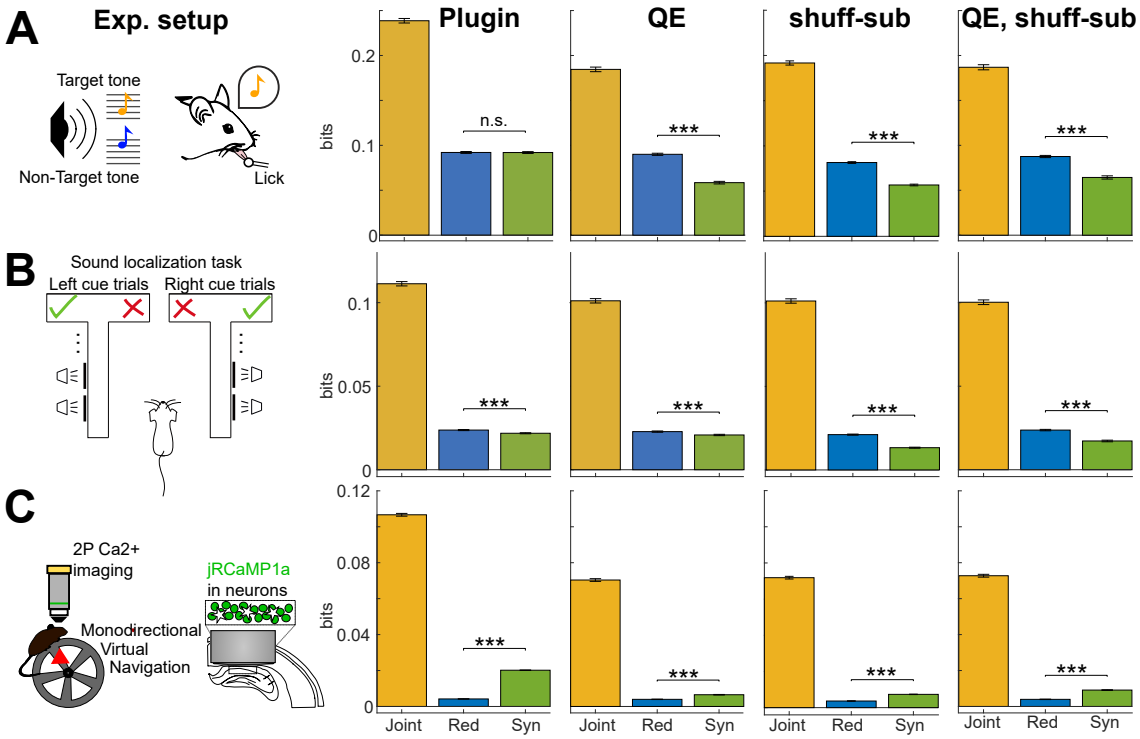

Figure 3.4: PID bias corrections on real neural data. Each panel plots mean ± 2 SEM over all analyzed simultaneously recorded neural pairs ($n = 6209, 10750, 36158$ for auditory cortex (top row), posterior parietal cortex (middle) and hippocampus (bottom) of joint information, synergy and Redundancy . Mean number N_s of available trials per stimulus per dataset was 70, 100, and 72, respectively. Columns from left to right plot: schematic of each task; results with plugin, QE, shuffle-subtraction, QE with shuffle subtraction respectively. Comparisons between synergy and redundancy were performed with a two-tailed paired t-test (*** : $p < 0.001$, n.s.: $p > 0.05$).

3.7 Discussion

We found that PID components suffer from a considerably limited sampling bias. Under simulated conditions relevant to neuroscience experiments, the bias was as large as the information quantities to be estimated, and thus cannot be neglected in neuroscience applications. Importantly, our work highlights and explains the presence of a major and previously neglected difference across PID terms in the sampling bias, which inflates synergy disproportionately. Neural synergy has been widely reported in recent years and has led neuroscientists to rethink how the brain integrates information. Our discovery calls for a careful re-evaluation of these reports with bias-corrected estimator.

We provided an analytical understanding of the properties and origin of the bias in terms of simple properties of experimental design (number of trials) and of analysis setting (number of discrete or discretized responses). Thus, our work helps informing experimental design.

Importantly, we provide generally applicable algorithms that correct for the bias and greatly improve information estimates with respect to state-of-the-art neural measures, which either neglected the bias problem (using uncorrected estimators) or correct for the bias using the incorrect assumptions that the bias is even across components. Importantly, the algorithms we develop not only improve in a major way the estimates, but some of these algorithms present positive and some other present negative residual estimation errors, allowing to empirically bound estimates with reasonable confidence.

From the neuroscientific point of view, applications of our method to simultaneous recordings with cellular resolution, confirmed the usefulness of the method to obtain reliable conclusions, highlight a widespread presence at the cellular level of synergy and redundancy in neuron-to-neuron interactions, and of region-to-region variations of the relationship between synergy and redundancy which were previously reported only at the level of aggregate signals without cellular resolution [76, 137].

We tested the bias of BROJA, I_{min} and I_{MMI} PID definitions. It would be important to test others. The bias correction procedures are heuristic, although we confirm and support them by providing an analytical expansion of the bias derived in the large N limit. However, our derivation is partly heuristic and we did not provide theoretical guarantees of sign of residual errors of different bias correction procedures. We tested information between pairs of neurons and stimuli but we have not tested large populations. The direct calculation of information is very precise for small populations and is largely assumption-free but it does not scale up well with population size unless dimensionality reduction methods are used with it. However, neuroscience literature has consistently shown the power and value of considering pairwise or small-group interactions between larger networks to get insights into whole networks [76, 137]. We support the feasibility of this approach simulating discovery of interactions from pairwise-source to single-neuron targets within a 6-neuron network using bias corrections (Fig. B.13). We show that using out bias corrections allows discovering the true pairs of neurons that transmit synergistically information even with small numbers of trials, whereas PID without bias corrections would find widespread artificial synergy due to bias.

Chapter 4

Contribution of interneuron diversity to recurrent network oscillation generation and information coding

The content of this chapter was published at the 15th International Conference on Brain Informatics, held in Hoboken and New Jersey, USA, in August 1-3 2023 [165].

4.1 Introduction

Oscillations are a ubiquitous feature of neural activity, which are thought to serve several important brain functions [10, 12, 16, 49, 166]. One of the functions that have been imputed to oscillations is the participation in the encoding of information from the sensory environment. Several experimental studies have demonstrated that cortical oscillations, especially in sensory areas, encode sensory information by modulating their power as a function of the sensory stimuli [15, 16, 56, 167]. Especially in visual cortices, the most information is carried by the power of gamma-band (40 – 100 Hz) oscillations. Experimental evidence [168] shows that the generation of these oscillations within recurrent circuits mainly relies on fast-spiking parvalbumin-expressing (PV) inhibitory neurons and their interaction with excitatory pyramidal neurons. Previous theoretical and computational studies of gamma oscillations typically included excitatory neurons and a single type of inhibitory neurons. The model of this inhibitory neuron did not specify or differentiate the type of interneuron being modeled, but that was loosely matched to the properties of fast-spiking PV neurons. These models could explain the generation of gamma oscillations exhibiting realistic spectral features, and could also explain the privileged encoding of information from the sensory periphery by the power of activity in the gamma band [11, 47–51].

Despite its success, recurrent network modeling based on a single undifferentiated interneuron type ignores the contribution of individual interneuron cell types to cortical oscillations and information coding. Besides the fast-spiking PV neurons considered

above, these interneuron types include somatostatin-expressing (SOM) and vasoinstestinal peptide-expressing (VIP) neurons [169].

In this work, we address these questions about the role of different interneuron types in network-level oscillatory information coding by extending previous modeling work on information encoding by recurrent networks [51, 170]. Specifically, we develop a recurrent network model containing excitatory neurons and SOM and VIP, as well as PV interneurons. We then analyze the behavior of this model to understand how the interaction between these types of neurons affects oscillations and their encoding of information. Using a conductance-based spiking neural network model adapted from [171], we computed the local field potential (LFP) and measured the mutual information between the external stimuli and the power generated by the network at each frequency.

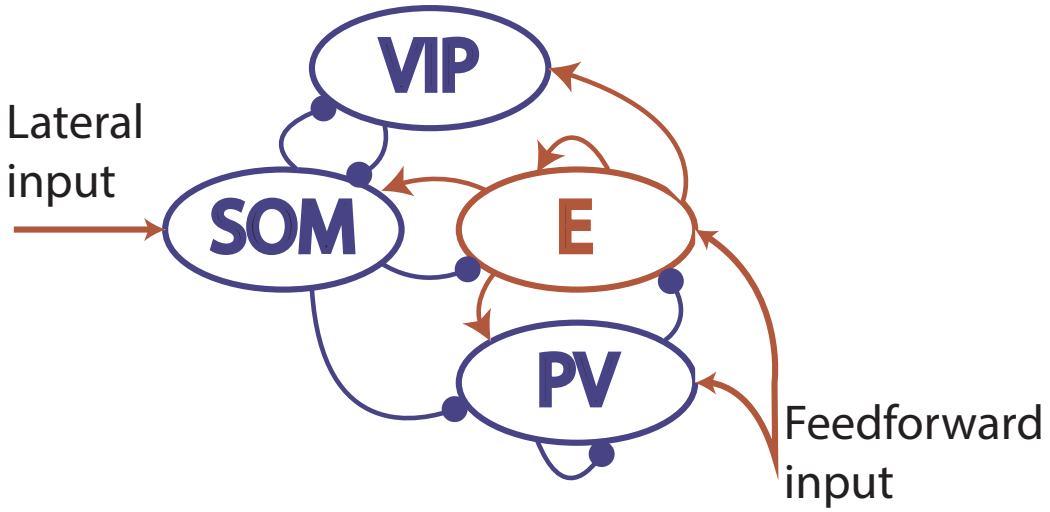

Figure 4.1: The network is composed of 500 neurons (400 excitatory neurons, 50 PV neurons, 25 SOM neurons, and 25 VIP neurons). There are two types of input that we change across simulations: a feed-forward excitatory input to E and PV neurons and a lateral one only targeting SOM neurons. There is also a background input that is kept constant across simulations

4.2 Methods

4.2.1 Network model

We implemented a spiking network of a recurrent cortical circuit that follows the characteristics of mouse visual cortex V1 [171, 172]. Besides the excitatory pyramidal population (E), we included three distinct inhibitory neurons. Parvalbumin-expressing neurons (PV) have strong recurrent connections within themselves and excitatory neurons. Somatostatin-expressing (SOM) neurons inhibit all neuron types except themselves, are preferentially excited by horizontal cortical connections [173] and have a causal role

in feedback- and horizontal-mediated suppression [173–176]. Vasointestinal peptide-expressing (VIP) neurons form the third largest population of inhibitory cell subtypes and preferentially inhibit SOM neurons.

Besides the inputs from recurrent connectivity, neurons in the network receive also an external stimulus signal S mimicking the effect of feedforward sensory inputs from the thalamus that targets E and PV neurons. We also include a lateral input rate that only targets SOM neurons [172, 173]. Both the stimulus and the lateral input were implemented as a Poisson process (240 Poisson sources) with a time-independent rate. The spike rate of the feed-forward input varied from 2 to 8 Hz/cell with steps of 2 Hz/cell and the lateral input was set to either to 2 or 4 Hz/cell.

The neurons were simulated using an adaptive exponential integrate-and-fire (aEIF) neuron model. This model is a relatively simple one, yet it reproduces the experimental qualitative properties of gamma oscillations and the different firing patterns of cortical dynamics [48, 177]. The following equation defines the evolution of the membrane potential V:

$$C\frac{dV}{dt} = I(t) - w(t) - g_L \cdot (V(t) - E_L) + g_L \cdot \Delta_T \cdot \mathrm{e}^{\frac{V(t)-V_{th}}{\Delta_T}}. \tag{4.1}$$

We took the parameter values from [171, 177]. The resting membrane potential E_L was set to -60 mV. The membrane capacitance C was set to 180 pF in excitatory neurons and 80 pF in inhibitory neurons. The leak conductance g_L is set as 6.25 nS in excitatory neurons and 5 nS in inhibitory neurons. The slope parameter Δ_T was set to 0.25 mV for PV neurons and 1 mV otherwise. The threshold potential V_{th} was set to -45 mV in excitatory neurons and -40 mV in all three types of inhibitory neurons.

The adaptation variable $w(t)$ evolved according to the following equation

$$\tau_w \frac{dw}{dt} = a(V(t) - E_L) - w(t), \tag{4.2}$$

where τ_w was set to 150 ms, and a to 4 nS [177]. Whenever the membrane potential reached 20 mV, a spike event was detected, the membrane potential was set to the reset voltage $V_{reset} = -70$ mV and the neuron did not spike for a refractory period of $t_{ref} = 2$ ms. The spike event also increases w by an amount $b = 80$ pA [177]. In the case of PV neurons, the adaptation w is set to zero [178].

The synaptic currents were modeled as $I_{syn} = g_{syn}(t) \cdot (V(t) - E_{syn})$, where $g_{syn}(t)$ is the synaptic conductance and E_{syn} is the reversal potential. The conductance $g_{syn}(t)$ was modeled as a double exponential (beta synapses) [179]:

$$g(t) = \frac{g_0}{\tau_d - \tau_r}\left(\mathrm{e}^{-\frac{t-\tau_l}{\tau_d}} - \mathrm{e}^{-\frac{t-\tau_l}{\tau_r}}\right), \tag{4.3}$$

where τ_l is the latency time, set to 1 ms. The time constants τ_r and τ_d are the rise and decay times of the post-synaptic conductance. For connections from pyramidal and PV neurons, τ_r is set to 0.5 ms, and 1 ms otherwise. The decay time τ_d was set to 2 ms for connections from pyramidal neurons, 3 ms for PV neurons, and 4 ms otherwise. The values of g_0 were chosen to match the post-synaptic current amplitudes in the visual cortex [171, 172] and are reported in Table 4.1.

Table 4.1: Table of the conductance values for each connection between two populations. The values are set in nS and the columns indicate the population of the presynaptic neuron and the rows, the post-synaptic one.

	E	PV	SOM	VIP
E	1.66	136.4	68.2	0
PV	5.0	136.4	45.5	0
SOM	0.83	0	0	136.4
VIP	1.66	27.3	113.6	0

Table 4.2: Table of the connection probability for any pair of neurons between two populations. The columns indicate the population of the presynaptic neuron and the rows, the post-synaptic one.

	E	PV	SOM	VIP
E	0.1	0.6	0.6	0
PV	0.6	0.6	0.6	0
SOM	0.6	0	0	0.4
VIP	0.6	0.1	0.6	0

The network model was adapted from [171], removing for simplicity the orientation tuning and synaptic plasticity. It comprises 500 neurons: 400 pyramidal, 50 PV neurons, 25 SOM neurons, and 25 VIP neurons. The probability of connection between neurons belonging to different population classes is shown in Table 4.2, chosen in accordance with connectivity measurements in visual cortex [171, 172].

The neurons in the network receive a third type of input that we named background input. The background input targets all neurons in the network model and sets the network in a regime of minimal spiking and gamma band oscillations, made by a constant input rate plus noise. The constant input rates are set as $(r_{0E}, r_{0P}, r_{0S}, r_{0V}) = (2.0, 0.33, 0.66, 0.25)\,\mathrm{Hz/cell}$. The noise was set as a slow-varying Ornstein-Uhlenbeck process, whose power spectrum is constant until 10 Hz, after which it decays.

For all three types of inputs, the random Poisson sources generating the stochastic input made excitatory synapses to the network's neurons with the same synaptic parameters that we used for the excitatory recurrent synapses.

For each combination of feed-forward and lateral stimulus values, we generated 50 simulations of 2.5 s with a time step of 0.1 ms using the NEST simulator module in Python [180]. The first half second of the simulations was discarded to ensure stationary dynamics.

Cortical oscillations are usually measured with Local Field Potentials (LFPs) [50]. However, LFPs are generated mostly by dendrosomatic dipoles whose computation cannot be obtained by our point-like neurons because generating these dipoles would require spatially extended neurons. However, we have demonstrated in previous work that the LFP generated by a network can be approximated simply and with high accuracy (ap-

proximately 90% of variance) by the sum of the absolute values of the synaptic current of all types of neurons, both excitatory and inhibitory [51, 166]. This proxy is based on the geometrical arrangement of the pyramidal neurons in the cortex [10, 11, 51, 166, 181]. We thus computed LFPs from our simulated network of point-like neurons using this proxy.

4.2.2 Information-theoretic and spectral analysis of simulated network activity

For each simulation, we computed the power spectral density (PSD) of the LFP signal with the multitaper method using the Chronux package [182].

We used mutual information [21, 23] to measure the information about the stimulus carried by the LFP power at each frequency f. Mutual information between a stimulus feature S and the LFP power at a given frequency R_f is defined as follows:

$$I(S;R_f) = \sum_s \sum_{r_f} P(r_f,s) \log_2 \frac{P(r_f,s)}{P(r_f)P(s)}. \tag{4.4}$$

where $P(r_f,s)$ is the joint probability of presenting stimulus s and observing response r_f, and $P(r_f)$, $P(s)$ are the marginal probabilities. For the numerical evaluation, following our previous work [166], we binned computed probabilities by binning the responses in 3 equi-populated bins [85], and by using a Panzeri-Treves bias correction [40, 42] to remove the sampling bias. To study how different frequency bands complemented each other for information coding, we also computed (in similar ways) the information $I(S;R_{f_1},R_{f_2})$ that was jointly encoded by the observation of the power at two different frequencies f_1 and f_2. To understand if power at different bands carried similar or different information about the stimulus, we also calculated their information synergy, defined as the difference between the information jointly carried by two frequencies and the sum of the information carried by the different frequencies, as follows [32, 33, 64]:

$$Syn(R_{f_1},R_{f_2}) = I(S;R_{f_1},R_{f_2}) - I(S;R_{f_1}) - I(S;R_{f_2}). \tag{4.5}$$

Note that this quantity is also termed Co-Information [30, 31]. Unlike more sophisticated quantities based on the Partial Information Decomposition [30, 31], it computes the total effect or synergy vs redundancy without further tearing apart the two. If $Syn(R_{f_1},R_{f_2})$ is negative, then the responses at frequencies f_1 and f_2 are carrying predominantly redundant information. If instead $Syn(R_{f_1},R_{f_2})$ is positive, the information carried by the two frequencies is predominantly synergistic and it would mean that the frequency pair has a fraction of information on the stimulus that cannot be accessed by each frequency value separately. Both the joint and synergistic information were computed with the shuffling technique from [40] which provides a conservative estimation of synergy.

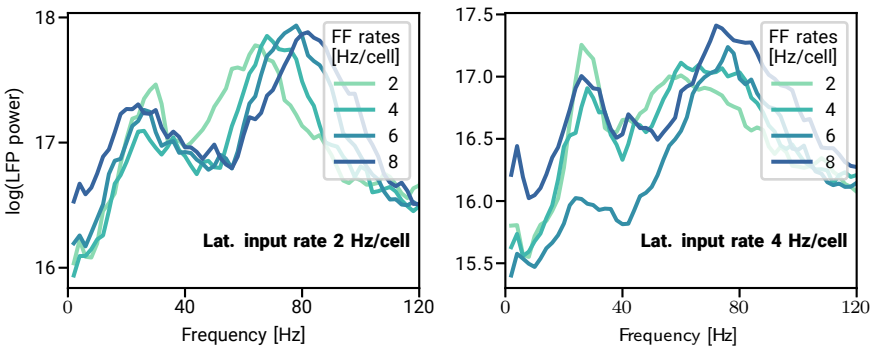

Figure 4.2: LFP spectra for different values of the feed-forward (S) input rates. Left: Spectra computed with a lateral input value of 2 Hz/cell. Right: Spectra computed with a lateral input value of 4 Hz/cell.

4.3 Results

As an intuitive demonstration of how the network responds to the feed-forward stimulus, Figure 4.2 shows the power spectrum of the local field potential (LFP) changes for different values of the feed-forward (S) input rates. All spectra show two local peaks. The first peak is at approximately 30 Hz (in the high-beta/low-gamma frequency band), and the second peak lies between 60 and 80 Hz in the gamma frequency band. The feed-forward input strength modulates mostly the frequencies in the gamma range. The power in the gamma range appeared to be more consistently modulated by the feedforward stimulus than the beta band power, across different values of the lateral input. This suggests that power in the gamma range may carry information about the feed-forward stimulus, as found in real data in visual cortices [15] and in earlier models with just one inhibitory class [51]. Importantly, the power at frequencies below the gamma peak frequency was modulated in different ways than the power of the frequencies above the gamma peak frequency. This suggests that power at different frequencies in the gamma range may carry some complementary information about the feed-forward stimulus.

To quantify this intuition in rigorous terms, we next computed and determined the mutual information carried by the LFP about the feed-forward stimulus. We express this as a function of frequency by calculating the mutual information $I(S;R_f)$ between the feed-forward S stimulus rates and the power R_f at the frequency f of the LFP spectrum. Information was higher in the gamma frequency range. The information had two peaks, one at approximately 55 Hz, just below the peak of the gamma power, and one at approximately 90 Hz, just above the peak of the gamma power. This is different from earlier recurrent network models, which had only one peak of gamma information which was located in approximate correspondence with the peak of the gamma power [51, 170, 181]. The values of the information peaks were slightly modulated by the strength of the lateral input, but similar structures were observed across changes in the lateral input (Figure 4.3).

The presence of two information peaks at different frequencies and the fact that the feed-forward stimulus modulates differently the LFP power above and below the peak

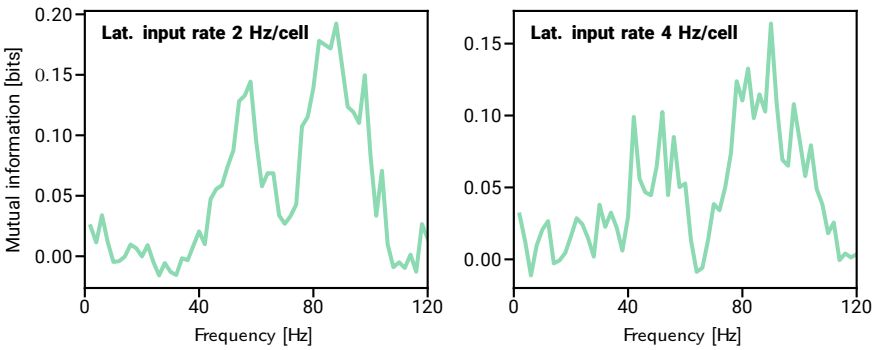

Figure 4.3: The mutual information $I(S;R_f)$ that the power of the LFP carries about the strength S of the feed-forward stimulus. Left: values of $I(S;R_f)$ when using as the value of the lateral input rate 2 Hz/cell. Right: values of $I(S;R_f)$ when the lateral input has a rate of 4 Hz/cell.

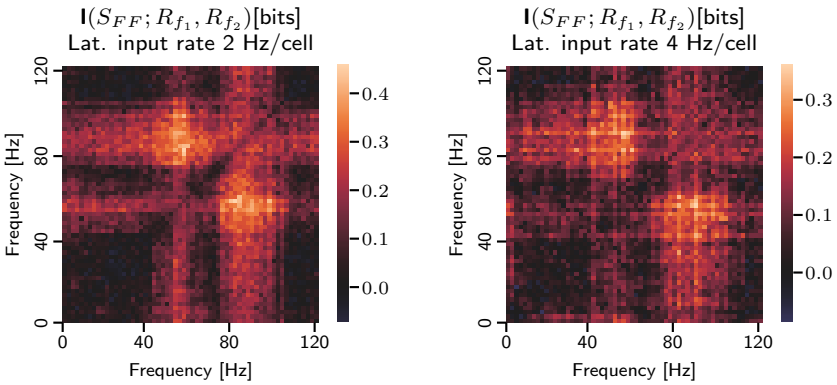

Figure 4.4: The joint information $I(S;R_{f_1},R_{f_2})$ that the LFP power of each pair of frequencies carries about the feed-forward stimulus S. Left: values of $I(S;R_{f_1},R_{f_2})$ when the lateral input rate has strength 2 Hz/cell. Right, the values of $I(S;R_{f_1},R_{f_2})$ when the lateral input rate has a value of 4 Hz/cell.

power frequency prompted us to study how much information could be gained by observing simultaneously in the same trial the power at two different frequencies. To investigate this, in Figure 4.4 we report for each pair of frequencies f_1, f_2 the joint information $I(S;R_{f_1},R_{f_2})$ that they carry about the feed-forward stimulus S. Interestingly, we found that the highest joint information value was reached when considering one frequency around the first gamma-band information peak and a frequency around the second gamma-band information peak. This suggests that the power of frequencies above or below the frequency with the highest gamma power carries complementary information about the stimuli.

To quantify this, in Figure 4.5 we report for each pair of frequencies f_1, f_2 the synergy $Syn(R_{f_1},R_{f_2})$ of the information of the two frequencies about the feed-forward stimulus. We found that, while the pairs of frequencies around the same information peak carried largely redundant information (negative values of synergy), pairs with one frequency around one information peak and one frequency around the other information peak car-

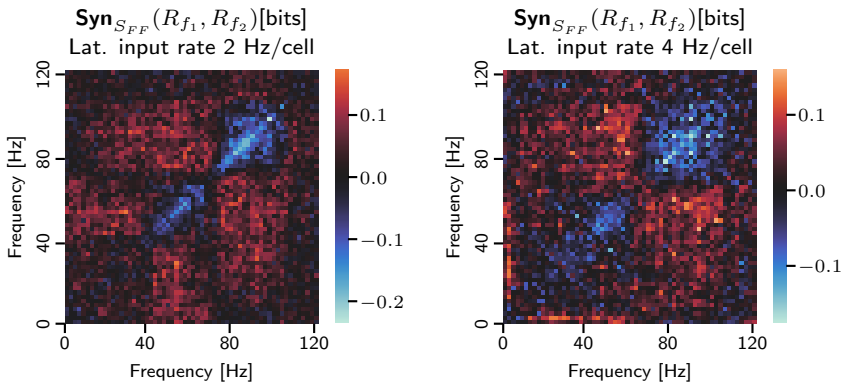

Figure 4.5: The amount of synergistic information $Syn(R_{f_1}, R_{f_2})$ that each pair of LFP powers at different frequencies f_1 and f_2 carries about the feed-forward stimulus S. Left: values of $Syn(R_{f_1}, R_{f_2})$ when using a lateral input rate of 2 Hz/cell. Right: values of synergy when using a lateral input rate of 4 Hz/cell.

ried synergistic information (positive synergy values; joint information larger than the sum of the two information values). The two regions with high redundancy along the diagonal have different sizes as expected from the different widths of the information peaks at the single frequency level, see Figure 4.3. Importantly, these patterns of synergy were not found in earlier modeling work with just one class of interneurons [51], in which we found only redundant information shared across frequencies in the gamma range.

4.4 Discussion

We used computer simulations of the dynamics of recurrent networks of spiking neurons to study if interneuron diversity affects network-level information coding. Several previous studies investigated how different interneuron subpopulations affect network dynamics [171, 175, 176]. However, none of these studies examined the effect of interneuron types on information encoding in single frequencies or in multiple frequencies. While a straightforward prediction would be that adding cell diversity may enrich the information processing capabilities of the network, it would be more difficult to predict from intuition only without the support of systematic simulations exactly how information coding at each frequency is affected, and how the patterns of information synergy and redundancy across bands are affected. Our main result was that a network with diverse types of interneurons has different and richer information encoding dynamics than a network with only one interneuron type, with patterns of synergy of encoding across frequencies that were not observed in less diverse networks.

One main result that we confirmed from previous studies of simpler networks is that we found was that, as reported in some previous experimental studies [16], the frequencies with the highest power were not necessarily those with the highest information or input modulation. Indeed, the frequencies that had the highest information were those above or below the frequency of the peak of the gamma power.

However, the main difference with respect to previous models was that oscillations in different frequency ranges (below or above the frequency with the peak gamma power) within the gamma band were differentially modulated by the strength of the feed-forward input to the network. In the model of [51] that includes only one un-differentiated interneuron type, the spectrum of network oscillations is modulated redundantly by the feedforward input at frequencies above 40 Hz, with no combination of frequencies in the power spectrum contributing synergistically to encode the input firing rate. This comparison leads us to attribute the formation of across-frequency synergistic information patterns to interneuron diversity.

In future work, it will be interesting to analyze datasets with field potential responses to different kinds of stimuli and see the extent to which the synergy of information across frequencies is realized, and which behavioral function it may serve. The differential and synergistic modulation by the feed-forward input of different frequencies within the gamma range suggests that different types of interactions between interneuron types control and modulate each part of the spectrum. Another relevant direction is to further study systematically in our model how each type of interaction between neuron classes regulates each part of the oscillation spectrum. Also, since the co-information expresses a net effect of synergy/redundancy, examining information encoding with partial information decompositions [30] would help us characterize whether overall redundancy or synergy of information encoding across two specific frequencies results from the simultaneous presence of different degrees of redundancy and synergy, or whether it results exclusively from the presence of synergy or redundancy.

Chapter 5

Conclusions

5.1 A new resource for neural information analysis

We showed here the Multivariate Information in Neuroscience Toolbox (MINT), a toolbox designed to provide a unified suite of information-theoretic tools for neural data analysis. Unlike previous toolboxes, which often focused solely on either information encoding or transmission, MINT integrates a comprehensive range of functions, including Shannon entropy, mutual information, directed transmission measures, and information decompositions. MINT also incorporates tools, such as limited sampling bias corrections and dimensionality reduction pipelines, that enhance accuracy for high-dimensional neural data.

MINT's utility is demonstrated on both real and simulated datasets, where it enables complex multivariate analyses that go beyond single-tool applications. Additionally, integrating encoding and transmission tools helps decode neural network information flow, offering a more detailed understanding of neural computation. This capability positions MINT as a resource for advancing discoveries in neuroscience.

5.2 Better estimates of PID

We examined and addressed sampling bias in Partial Information Decomposition (PID) under realistic experimental conditions relevant to neuroscience. We found that this bias can reach magnitudes comparable to the actual information quantities, particularly inflating synergy, a component critical to understanding how the brain integrates information. This inflation suggests that reports of neural synergy may need careful re-evaluation with bias-corrected methods.

To address this, we developed and validated algorithms to correct for the bias, providing significant improvements over current neural measures that either overlook or incorrectly assume uniform bias across PID components. Our algorithms not only yield more accurate estimates but also present bounded errors, offering a more reliable way to interpret information decomposition results. Additionally, our analysis reveals that PID bias is tied to basic properties of experimental design, such as trial numbers and the number

of discrete responses, which can inform more precise experimental planning. Applying these bias-corrected methods to real, high-resolution neuronal recordings, we confirmed the value of these corrections.

While promising, our work has limitations. We focused on specific PID definitions (BROJA, I_{min}, and I_{MMI}) and have yet to test other formulations. Our bias-correction procedures are heuristic and lack full theoretical guarantees, particularly regarding error signs. Although we validated PID on neuron pairs, scaling up to large populations remains challenging without dimensionality reduction, as direct computation becomes inefficient. Nonetheless, evidence from smaller network models supports the utility of pairwise analysis in revealing network-level interactions.

5.3 Interneuron diversity and encoding

We used simulations of recurrent spiking neuron networks to investigate how interneuron diversity influences network-level information coding, focusing on encoding dynamics across different frequency bands. While previous studies examined interneuron diversity's impact on network dynamics, they did not address its role in information encoding at single or multiple frequencies. Our findings reveal that networks with diverse interneuron types show richer information encoding patterns than those with only one interneuron type, displaying unique synergy across frequency bands.

In line with past studies, we observed that the highest-power frequencies do not always carry the most information; in our simulations, frequencies above or below the gamma power peak encoded more input information. However, unlike previous models that assume a single interneuron type, our model shows that oscillations in different gamma sub-bands respond uniquely to feed-forward input strength, suggesting that interneuron diversity enables synergistic encoding patterns across frequencies.

This work suggests that interneuron diversity may create nuanced modulation within the gamma band, with distinct neuron interactions controlling specific spectral parts. Future work could explore real neural data to confirm these synergy patterns across frequencies and assess their behavioral relevance. Additionally, further analysis with partial information decomposition may clarify whether the observed synergy across frequencies results from a balance of redundancy and synergy or from purely synergistic encoding.

Bibliography

1. Kandel, E. *Essentials of neural science and behavior* (Appleton & Lange, 1995).
2. Bassett, D. S. & Sporns, O. Network neuroscience. *Nature Neuroscience* **20,** 353–364 (2017).
3. Dayan, P. & Abbott, L. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems* (Massachusetts Institute of Technology Press, 2001).
4. Kutter, E. F., Bostroem, J., Elger, C. E., Mormann, F. & Nieder, A. Single Neurons in the Human Brain Encode Numbers. *Neuron* **100,** 753–761.e4 (2018).
5. Leonard, M. K. *et al.* Large-scale single-neuron speech sound encoding across the depth of human cortex. *Nature* **626,** 593–602 (2024).
6. Quian Quiroga, R. & Panzeri, S. Extracting information from neuronal populations: information theory and decoding approaches. *Nature Reviews Neuroscience* **10,** 173–185 (2009).
7. Xie, W. *et al.* Neuronal sequences in population bursts encode information in human cortex. *Nature,* 1–8 (2024).
8. Averbeck, B. B., Latham, P. E. & Pouget, A. Neural correlations, population coding and computation. *Nature Reviews Neuroscience* **7,** 358–366 (2006).
9. Dan, Y., Alonso, J.-M., Usrey, W. M. & Reid, R. C. Coding of visual information by precisely correlated spikes in the lateral geniculate nucleus. *Nature Neuroscience* **1,** 501–507 (1998).
10. Buzsáki, G., Anastassiou, C. A. & Koch, C. The origin of extracellular fields and currents — EEG, ECoG, LFP and spikes. *Nature Reviews Neuroscience* **13,** 407–420 (2012).
11. Mazzoni, A. *et al.* Computing the Local Field Potential (LFP) from Integrate-and-Fire Network Models. *PLOS Computational Biology* **11,** e1004584 (2015).
12. Pesaran, B. *et al.* Investigating large-scale brain dynamics using field potential recordings: analysis and interpretation. *Nature Neuroscience* **21,** 903–919 (2018).
13. Bullmore, E. & Sporns, O. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience* **10,** 186–198 (2009).

14. Buzsáki, G. Large-scale recording of neuronal ensembles. *Nature Neuroscience* **7,** 446–451 (2004).

15. Henrie, J. A. & Shapley, R. LFP Power Spectra in V1 Cortex: The Graded Effect of Stimulus Contrast. *Journal of Neurophysiology* **94,** 479–490 (2005).

16. Belitski, A., Panzeri, S., Magri, C., Logothetis, N. K. & Kayser, C. Sensory information in local field potentials and spikes from visual and auditory cortices: time scales and frequency bands. *Journal of Computational Neuroscience* **29,** 533–545 (2010).

17. Grienberger, C., Giovannucci, A., Zeiger, W. & Portera-Cailliau, C. Two-photon calcium imaging of neuronal activity. *Nature Reviews Methods Primers* **2,** 67 (2022).

18. Johnson, E. L. *et al.* A rapid theta network mechanism for flexible information encoding. *Nature Communications* **14,** 2872 (2023).

19. Coffey, E. B. J., Herholz, S. C., Chepesiuk, A. M. P., Baillet, S. & Zatorre, R. J. Cortical contributions to the auditory frequency-following response revealed by MEG. *Nature Communications* **7,** 11070 (2016).

20. Albouy, P., Mattout, J., Sanchez, G., Tillmann, B. & Caclin, A. Altered retrieval of melodic information in congenital amusia: insights from dynamic causal modeling of MEG data. *Frontiers in Human Neuroscience* **9,** 20 (2015).

21. Shannon, C. A mathematical theory of communication. *The Bell System Technical Journal* **27,** 379–423 (1948).

22. Timme, N. M. & Lapish, C. A tutorial for information theory in neuroscience. *eNeuro* **5** (2018).

23. Cover, T. M. & Thomas, J. A. Information theory and statistics. *Elements of information theory* **1,** 279–335 (1991).

24. Wibral, M., Vicente, R. & Lizier, J. T. *Directed information measures in neuroscience* (Springer, 2014).

25. Borst, A. & Theunissen, F. E. Information theory and neural coding. *Nature Neuroscience* **2,** 947–957 (1999).

26. Fairhall, A., Shea-Brown, E. & Barreiro, A. Information theoretic approaches to understanding circuit function. *Current Opinion in Neurobiology* **22,** 653–659 (2012).

27. Azeredo da Silveira, R. & Rieke, F. The geometry of information coding in correlated neural populations. *Annual Review of Neuroscience* **44,** 403–424 (2021).

28. Fairhall, A. L., Lewen, G. D., Bialek, W. & de Ruyter van Steveninck, R. R. Efficiency and ambiguity in an adaptive neural code. *Nature* **412,** 787–792 (2001).

29. Atick, J. J. & Redlich, A. N. Towards a theory of early visual processing. *Neural Computation* **2,** 308–320 (1990).

30. Williams, P. L. & Beer, R. D. Nonnegative decomposition of multivariate information. *arXiv:1004.2515* (2010).

31. Bertschinger, N., Rauh, J., Olbrich, E., Jost, J. & Ay, N. Quantifying unique information. *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies* **16,** 2161–2183 (2014).
32. Panzeri, S., Schultz, S. R., Treves, A. & Rolls, E. T. Correlations and the encoding of information in the nervous system. *Proceedings of the Royal Society of London Series B: Biological Sciences* **266,** 1001–1012 (1999).
33. Pola, G, Thiele, A, Hoffmann, K. & Panzeri, S. An exact method to quantify the information transmitted by different mechanisms of correlational coding. *Network: Computation in Neural Systems* **14,** 35–60 (2003).
34. Nigam, S., Pojoga, S. & Dragoi, V. Synergistic coding of visual information in columnar networks. *Neuron* **104,** 402–411 (2019).
35. Runyan, C. A., Piasini, E., Panzeri, S. & Harvey, C. D. Distinct timescales of population coding across cortex. *Nature* **548,** 92–96 (2017).
36. Panzeri, S., Harvey, C. D., Piasini, E., Latham, P. E. & Fellin, T. Cracking the Neural Code for Sensory Perception by Combining Statistics, Intervention, and Behavior. *Neuron* **93,** 491–507 (2017).
37. Pica, G. *et al. Quantifying how much sensory information in a neural code is relevant for behavior* in *Advances in neural information processing systems 30* (2017), 3686–3696.
38. Celotto, M. *et al. An information-theoretic quantification of the content of communication between brain regions* in *Advances in neural information processing systems* **36** (2023), 64213–64265.
39. Optican, L. M., Gawne, T. J., Richmond, B. J. & Joseph, P. J. Unbiased measures of transmitted information and channel capacity from multivariate neuronal data. *Biological Cybernetics* **65,** 305–310 (1991).
40. Panzeri, S., Senatore, R., Montemurro, M. A. & Petersen, R. S. Correcting for the sampling bias problem in spike train information measures. *Journal of Neurophysiology* **98,** 1064–1072 (2007).
41. Strong, S. P., Koberle, R., De Ruyter Van Steveninck, R. & Bialek, W. Entropy and information in neural spike trains. *Physical Review Letters* **80,** 197 (1998).
42. Panzeri, S. & Treves, A. Analytical estimates of limited sampling biases in different information measures. *Network: Computation in Neural Systems* **7,** 87 (1996).
43. Paninski, L. Estimation of entropy and mutual information. *Neural Computation* **15,** 1191–1253 (2003).
44. Cunningham, J. P. & Yu, B. M. Dimensionality reduction for large-scale neural recordings. *Nature Neuroscience* **17,** 1500–1509 (2014).
45. Venkatesh, P. *et al. Gaussian partial information decomposition: Bias correction and application to high-dimensional data* in *Advances in neural information processing systems* **36** (2023), 74602–74635.

46. Gerstner, W., Kistler, W. M., Naud, R. & Paninski, L. *Neuronal dynamics: From single neurons to networks and models of cognition* (Cambridge University Press, 2014).

47. Barbieri, F., Mazzoni, A., Logothetis, N. K., Panzeri, S. & Brunel, N. Stimulus Dependence of Local Field Potential Spectra: Experiment versus Theory. *Journal of Neuroscience* **34,** 14589–14605 (2014).

48. Brunel, N. & Wang, X.-J. What Determines the Frequency of Fast Network Oscillations With Irregular Neural Discharges? I. Synaptic Dynamics and Excitation-Inhibition Balance. *Journal of Neurophysiology* **90,** 415–430 (2003).

49. Buzsáki, G. & Draguhn, A. Neuronal Oscillations in Cortical Networks. *Science* **304,** 1926–1929 (2004).

50. Einevoll, G. T., Kayser, C., Logothetis, N. K. & Panzeri, S. Modelling and analysis of local field potentials for studying the function of cortical circuits. *Nature Reviews Neuroscience* **14.** Publisher: Nature Publishing Group, 770–785 (2013).

51. Mazzoni, A., Panzeri, S., Logothetis, N. K. & Brunel, N. Encoding of Naturalistic Stimuli by Local Field Potential Spectra in Networks of Excitatory and Inhibitory Neurons. *PLoS Computational Biology* **4** (ed Friston, K. J.) e1000239 (2008).

52. Lorenz, G. M. *et al.* MINT: a toolbox for the analysis of multivariate neural information coding and transmission. *bioRxiv* (2024).

53. Panzeri, S., Moroni, M., Safaai, H. & Harvey, C. D. The structures and functions of correlations in neural population codes. *Nature Reviews Neuroscience* **23,** 551–567 (2022).

54. Urai, A. E., Doiron, B., Leifer, A. M. & Churchland, A. K. Large-scale neural recordings call for new insights to link brain and behavior. *Nature Neuroscience,* 1–9 (2022).

55. Panzeri, S., Brunel, N., Logothetis, N. K. & Kayser, C. Sensory neural codes using multiplexed temporal scales. *Trends in neurosciences* **33,** 111–120 (2010).

56. Belitski, A. *et al.* Low-Frequency Local Field Potentials and Spikes in Primary Visual Cortex Convey Independent Visual Information. *Journal of Neuroscience* **28,** 5696–5709 (2008).

57. Douchamps, V., di Volo, M., Torcini, A., Battaglia, D. & Goutagny, R. Gamma oscillatory complexity conveys behavioral information in hippocampal networks. *Nature Communications* **15,** 1849 (2024).

58. Kayser, C., Montemurro, M. A., Logothetis, N. K. & Panzeri, S. Spike-Phase Coding Boosts and Stabilizes Information Carried by Spatial and Temporal Spike Patterns. *Neuron* **61,** 597–608 (2009).

59. De Ruyter van Steveninck, R. R., Lewen, G. D., Strong, S. P., Koberle, R. & Bialek, W. Reproducibility and variability in neural spike trains. *Science* **275,** 1805–1808 (1997).

60. Victor, J. D. Approaches to information-theoretic analysis of neural activity. *Biological Theory* **1,** 302–316 (2006).

61. Gollisch, T. & Meister, M. Rapid neural coding in the retina with relative spike latencies. *Science* **319,** 1108–1111 (2008).

62. Młynarski, W. F. & Hermundstad, A. M. Efficient and adaptive sensory codes. *Nature Neuroscience* **24,** 998–1009 (2021).

63. Latham, P. E. & Nirenberg, S. Synergy, redundancy, and independence in population codes, revisited. *Journal of Neuroscience* **25,** 5195–5206 (2005).

64. Schneidman, E., Bialek, W. & Berry, M. J. Synergy, redundancy, and independence in population codes. *Journal of Neuroscience* **23,** 11539–11553 (2003).

65. Ince, R. A., Panzeri, S. & Kayser, C. Neural codes formed by small and temporally precise populations in auditory cortex. *Journal of Neuroscience* **33,** 18277–18287 (2013).

66. Froudarakis, E. *et al.* Population code in mouse V1 facilitates readout of natural scenes through increased sparseness. *Nature Neuroscience* **17,** 851–857 (2014).

67. Onken, A. *et al.* Using Matrix and Tensor Factorizations for the Single-Trial Analysis of Population Spike Trains. *PLOS Computational Biology* **12** (ed Bethge, M.) e1005189 (2016).

68. Kühn, N. K. & Gollisch, T. Activity correlations between direction-selective retinal ganglion cells synergistically enhance motion decoding from complex visual scenes. *Neuron* **101,** 963–976.e7 (2019).

69. Curreli, S., Bonato, J., Romanzi, S., Panzeri, S. & Fellin, T. Complementary encoding of spatial information in hippocampal astrocytes. *PLOS Biology* **20,** e3001530 (2022).

70. Sharpee, T. O. & Berkowitz, J. A. Linking neural responses to behavior with information-preserving population vectors. *Current Opinion in Behavioral Science* **29,** 37–44 (2019).

71. Iurilli, G. & Datta, S. R. Population coding in an innately relevant olfactory area. *Neuron* **93,** 1180–1197 (2017).

72. Kira, S., Safaai, H., Morcos, A. S., Panzeri, S. & Harvey, C. D. A distributed and efficient population code of mixed selectivity neurons for flexible navigation decisions. *Nature Communications* **14,** 2121 (2023).

73. Vicente, R., Wibral, M., Lindner, M. & Pipa, G. Transfer entropy—a model-free measure of effective connectivity for the neurosciences. *Journal of Computational Neuroscience* **30,** 45–67 (2011).

74. Colenbier, N. *et al.* Disambiguating the role of blood flow and global signal with partial information decomposition. *NeuroImage* **213,** 116699 (2020).

75. Besserve, M., Lowe, S. C., Logothetis, N. K., Scholkopf, B. & Panzeri, S. Shifts of gamma phase across primary visual cortical sites reflect dynamic stimulus-modulated information transfer. *PLoS Biology* **13,** e1002257 (2015).

76. Luppi, A. I. *et al.* A synergistic core for human brain evolution and cognition. *Nature Neuroscience* **25,** 771–782 (2022).

77. Lemke, S. M., Celotto, M., Maffulli, R., Ganguly, K. & Panzeri, S. Information flow between motor cortex and striatum reverses during skill learning. *Current Biology* **34,** 1831–1843 (2024).

78. Sherrill, S. P., Timme, N. M., Beggs, J. M. & Newman, E. L. Partial information decomposition reveals that synergistic neural integration is greater downstream of recurrent information flow in organotypic cortical cultures. *PLoS Computational Biology* **17,** e1009196 (2021).

79. Reid, A. T. *et al.* Advancing functional connectivity research from association to causation. *Nature Neuroscience* **22,** 1751–1760 (2019).

80. Stramaglia, S., Wu, G.-R., Pellicoro, M. & Marinazzo, D. Expanding the transfer entropy to identify information circuits in complex systems. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* **86,** 066211 (2012).

81. Wibral, M. *et al.* Measuring information-transfer delays. *PLOS ONE* **8,** e55809 (2013).

82. Pereira, T. D., Shaevitz, J. W. & Murthy, M. Quantifying behavior to understand the brain. *Nature Neuroscience* **23,** 1537–1549 (2020).

83. Panniello, M. *et al.* Stimulus information guides the emergence of behavior-related signals in primary somatosensory cortex during learning. *Cell Reports* **43,** 114244 (2024).

84. Ince, R. A. A. Python for information theoretic analysis of neural data. *Frontiers in Neuroinformatics* **3** (2009).

85. Magri, C., Whittingstall, K., Singh, V., Logothetis, N. K. & Panzeri, S. A toolbox for the fast information analysis of multiple-site LFP, EEG and spike train recordings. *BMC Neuroscience* **10,** 81 (2009).

86. Lizier, J. T. JIDT: An information-theoretic toolkit for studying the dynamics of complex systems. *Frontiers in Robotics and AI* **1,** 11 (2014).

87. Combrisson, E. *et al.* Group-level inference of information-based measures for the analyses of cognitive brain networks from neurophysiological data. *NeuroImage* **258,** 119347 (2022).

88. Ince, R. A. *et al.* A statistical framework for neuroimaging data analysis based on mutual information estimated via a gaussian copula. *Human Brain Mapping* **38,** 1541–1573 (2017).

89. Climer, J. R. & Dombeck, D. A. Information theoretic approaches to deciphering the neural code with functional fluorescence imaging. *eNeuro* **8,** ENEURO.0266–21.2021 (2021).

90. M.C. & E.J.I. infotheory: A C++/Python package for multivariate information theoretic analysis. *Journal of Open Source Software* **57,** 1609 (2020).

91. Makkeh, A., Theis, D. O. & Vicente, R. BROJA-2PID: A Robust Estimator for Bivariate Partial Information Decomposition. *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies* **20,** 271 (2018).

92. James, R. G., Ellison, C. J. & Crutchfield, J. P. dit: a Python package for discrete information theory. *Journal of Open Source Software* **3,** 738 (2018).

93. Moore, D. G., Valentini, G., Walker, S. I. & Levin, M. *Inform: A toolkit for information-theoretic analysis of complex systems* in *2017 IEEE symposium series on computational intelligence (SSCI)* (2017).

94. Pastore, V. P., Poli, D., Godjoski, A., Martinoia, S. & Massobrio, P. ToolConnect: a functional connectivity toolbox for in vitro networks. *Frontiers in Neuroinformatics* **10** (2016).

95. Montalto, A., Faes, L. & Marinazzo, D. MuTE: a MATLAB toolbox to compare established and novel estimators of the multivariate transfer entropy. *PLOS ONE* **9,** e109462 (2014).

96. Szabó, Z. Information theoretical estimators toolbox. *The Journal of Machine Learning Research* **15,** 283–287 (2014).

97. Lindner, M., Vicente, R., Priesemann, V. & Wibral, M. TRENTOOL: A Matlab open source toolbox to analyse information flow in time series data with transfer entropy. *BMC Neuroscience* **12,** 119 (2011).

98. Ito, S. *et al.* Extending transfer entropy improves identification of effective connectivity in a spiking cortical network model. *PLOS ONE* **6,** e27431 (2011).

99. Goldberg, D. H., Victor, J. D., Gardner, E. P. & Gardner, D. Spike train analysis toolkit: Enabling wider application of information-theoretic techniques to neurophysiology. *Neuroinformatics* **7,** 165–178 (2009).

100. Candadai, M. & Izquierdo, E. J. infotheory: A C++/Python package for multivariate information theoretic analysis. *Journal of Open Source Software* **5,** 1609 (2020).

101. Neri, M. *et al.* HOI: A Python toolbox for high-performance estimation of Higher-Order Interactions from multivariate data. *Journal of Open Source Software* **9,** 7360 (2024).

102. Emanuel, A. J., Lehnert, B. P., Panzeri, S., Harvey, C. D. & Ginty, D. D. Cortical responses to touch reflect subcortical integration of LTMR signals. *Nature* **600,** 680–685 (2021).

103. Schneidman, E., Berry, M. J., Segev, R. & Bialek, W. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature* **440,** 1007–1012 (2006).

104. Ostwald, D., Porcaro, C. & Bagshaw, A. P. An information theoretic approach to EEG–fMRI integration of visually evoked responses. *Neuroimage* **49,** 498–516 (2010).

105. Pessoa, L. & Padmala, S. Decoding near-threshold perception of fear from distributed single-trial brain activation. *Cerebral Cortex* **17,** 691–701 (2007).

106. Schultz, S. R., Kitamura, K., Post-Uiterweer, A., Krupic, J. & Häusser, M. Spatial pattern coding of sensory information by climbing fiber-evoked calcium signals in networks of neighboring cerebellar Purkinje cells. *Journal of Neuroscience* **29,** 8005–8015 (2009).

107. Nirenberg, S., Carcieri, S. M., Jacobs, A. L. & Latham, P. E. Retinal ganglion cells act largely as independent encoders. *Nature* **411,** 698–701 (2001).

108. Oizumi, M., Ishii, T., Ishibashi, K., Hosoya, T. & Okada, M. Mismatched decoding in the brain. *Journal of Neuroscience* **30,** 4815–4826 (2010).

109. Wibral, M., Priesemann, V., Kay, J. W., Lizier, J. T. & Phillips, W. A. Partial information decomposition as a unified approach to the specification of neural goal functions. *Brain and Cognition* **112,** 25–38 (2017).

110. Barrett, A. B. Exploration of synergistic and redundant information sharing in static and dynamical Gaussian systems. *Physical Review E* **91,** 052802 (2015).

111. Rauh, J., Banerjee, P. K., Olbrich, E., Montúfar, G. & Jost, J. Continuity and additivity properties of information decompositions. *International Journal of Approximate Reasoning* **161,** 108979 (2023).

112. Schreiber, T. Measuring information transfer. *Physical Review Letters* **85,** 461–464 (2000).

113. Gross, J. *et al.* Speech rhythms and multiplexed oscillatory sensory coding in the human brain. *PLOS Biology* **11,** e1001752 (2013).

114. Orlowska-Feuer, P. *et al.* Look-up and look-down neurons in the mouse visual thalamus during freely moving exploration. *Current Biology* **32,** 3987–3999 (2022).

115. Scott, D. W. Scott's rule. *WIREs Computational Statistics* **2,** 497–502 (2010).

116. Freedman, D. & Diaconis, P. On the histogram as a density estimator: L2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* **57,** 453–476 (1981).

117. Montemurro, M. A., Senatore, R. & Panzeri, S. Tight data-robust bounds to mutual information combining shuffling and model selection techniques. *Neural Computation* **19,** 2913–2957 (2007).

118. Ince, R. A. *et al.* Information-theoretic methods for studying population codes. *Neural Networks* **23,** 713–727 (2010).

119. Kraskov, A., Stögbauer, H. & Grassberger, P. Estimating mutual information. *Physical Review E* **69,** 066138 (2004).

120. Holmes, C. M. & Nemenman, I. Estimation of mutual information for real-valued data with error bars and controlled bias. *Physical Review E* **100,** 022404 (2019).

121. Nemenman, I., Bialek, W. & de Ruyter van Steveninck, R. Entropy and information in neural spike trains: Progress on the sampling problem. *Physical Review E* **69,** 056111 (2004).

122. Lee, D. D. & Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature* **401,** 788–791 (1999).

123. Chang, C.-C. & Lin, C.-J. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* **2,** 1–27 (2011).

124. Friedman, J. H., Hastie, T. & Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33,** 1–22 (2010).

125. Graf, A. B. A., Kohn, A., Jazayeri, M. & Movshon, J. A. Decoding the activity of neuronal populations in macaque primary visual cortex. *Nature Neuroscience* **14,** 239–245 (2011).

126. Reich, D. S., Mechler, F. & Victor, J. D. Independent and redundant information in nearby cortical neurons. *Science* **294,** 2566–2568 (2001).

127. Chechik, G. *et al.* Group redundancy measures reveal redundancy reduction in the auditory pathway. *Advances in neural information processing systems* **14** (2001).

128. Beer, R. D. & Williams, P. L. Information processing and dynamics in minimally cognitive agents. *Cognitive Science* **39,** 1–38 (2015).

129. Francis, N. A. *et al.* Sequential transmission of task-relevant information in cortical neuronal networks. *Cell Reports* **39,** 110878 (2022).

130. Valente, M. *et al.* Correlations enhance the behavioral readout of neural population activity in association cortex. *Nature Neuroscience* **24,** 975–986 (2021).

131. Rousselet, G. A., Ince, R. A., van Rijsbergen, N. J. & Schyns, P. G. Eye coding mechanisms in early human face event-related potentials. *Journal of Vision* **14,** 7 (2014).

132. Ince, R. A. A. *et al.* The Deceptively Simple N170 Reflects Network Information Processing Mechanisms Involving Visual Feature Coding and Transfer Across Hemispheres. *Cerebral Cortex* **26,** 4123–4135 (2016).

133. Oostenveld, R., Fries, P., Maris, E. & Schoffelen, J. M. FieldTrip: Open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Computational Intelligence and Neuroscience* **2011,** 156869 (2011).

134. Rubinov, M. & Sporns, O. Complex network measures of brain connectivity: Uses and interpretations. *NeuroImage* **52,** 1059–1069 (2010).

135. Safaai, H., Onken, A., Harvey, C. D. & Panzeri, S. Information estimation using nonparametric copulas. *Physical Review E* **98,** 053302 (2018).

136. Victor, J. D. Binless strategies for estimation of information from neural data. *Physical Review E* **66,** 051903 (2002).

137. Luppi, A. I., Rosas, F. E., Mediano, P. A., Menon, D. K. & Stamatakis, E. A. Information decomposition and the informational architecture of the brain. *Trends in Cognitive Sciences* **24,** 352–358 (2024).

138. Koçillari, L. *et al.* Sampling bias corrections for accurate neural measures of redundant, unique, and synergistic information. *bioRxiv* (2024).

139. Barlow, H. B. Single units and sensation: a neuron doctrine for perceptual psychology? *Perception* **1,** 371–394 (1972).

140. Abbott, L. F. & Dayan, P. The effect of correlated variability on the accuracy of a population code. *Neural Computation* **11,** 91–101 (1999).

141. Harder, M., Salge, C. & Polani, D. Bivariate measure of redundant information. *Physical Review E* **87,** 012130 (2013).

142. Varley, T. F., Sporns, O., Schaffelhofer, S., Scherberger, H. & Dann, B. Information-processing dynamics in neural networks of macaque cerebral cortex reflect cognitive state and behavior. *Proceedings of the National Academy of Sciences* **120,** e2207677120 (2023).

143. Cover, T. M. & Thomas, J. A. *Elements of information theory* (Wiley & Sons, Hoboken, New Jersey, 2006).

144. Lizier, J. T., Bertschinger, N., Jost, J. & Wibral, M. Information decomposition of target effects from multi-source interactions: Perspectives on previous, current and future work. *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies* **20,** 307 (2018).

145. Kolchinsky, A. A novel approach to the partial information decomposition. *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies* **24** (2022).

146. Wollstadt, P. *et al.* Information-theoretic analyses of neural data to minimize the effect of researchers' assumptions in predictive coding studies. *PLoS Computational Biology* **19,** e1011567 (2023).

147. Tolhurst, D. J., Movshon, J. A. & Dean, A. F. The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vision research* **23,** 775–785 (1983).

148. Panzeri, S., Petersen, R. S., Schultz, S. R., Lebedev, M. & Diamond, M. E. The role of spike timing in the coding of stimulus location in rat somatosensory cortex. *Neuron* **29,** 769–777 (2001).

149. Ohiorhenuan, I. E. *et al.* Sparse coding and high-order correlations in fine-scale cortical networks. *Nature* **466,** 617–621 (2010).

150. Kuan, A. T. *et al.* Synaptic wiring motifs in posterior parietal cortex support decision-making. *Nature* **627,** 367–373 (2024).

151. Stevens, C. & Zador, A. *Information through a spiking neuron* in *Advances in neural information processing systems* (eds Touretzky, D., Mozer, M. & Hasselmo, M.) **8** (MIT Press, 1995).

152. Kayser, C., Logothetis, N. K. & Panzeri, S. Millisecond encoding precision of auditory cortex neurons. *Proceedings of the National Academy of Sciences* **107,** 16976–16981 (2010).

153. Schneidman, E., Bialek, W. & Ii, M. An information theoretic approach to the functional classification of neurons. *Advances in neural information processing systems* **15,** 197–204 (2002).

154. Granot-Atedgi, E., Tkačik, G., Segev, R. & Schneidman, E. Stimulus-dependent maximum entropy models of neural population codes. *PLOS Computational Biology* **9,** 1–14 (2013).

155. Timme, N. M. *et al.* High-degree neurons feed cortical computations. *PLOS Computational Biology* **12,** 1–31 (2016).

156. Pica, G., Piasini, E., Chicharro, D. & Panzeri, S. Invariant components of synergy, redundancy, and unique information among three variables. *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies* **19,** 451 (2017).

157. Wibral, M., Finn, C., Wollstadt, P., Lizier, J. T. & Priesemann, V. Quantifying information modification in developing neural networks via partial information decomposition. *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies* **19,** 494 (2017).

158. Koçillari, L. *et al.* Behavioural relevance of redundant and synergistic stimulus information between functionally connected neurons in mouse auditory cortex. *Brain Informatics* **10,** 34 (2023).

159. Chan, T. E., Stumpf, M. P. & Babtie, A. C. Gene regulatory network inference from single-cell data using multivariate information measures. *Cell Systems* **5,** 251–267.e3 (2017).

160. Cang, Z. & Nie, Q. Inferring spatial and signaling relationships between cells from single cell transcriptomic data. *Nature Communications* **11,** 2084 (2020).

161. Kramer, B. A., Sarabia del Castillo, J. & Pelkmans, L. Multimodal perception links cellular state to decision-making in single cells. *Science* **377,** 642–648 (2022).

162. Wollstadt, P., Schmitt, S. & Wibral, M. A rigorous information-theoretic definition of redundancy and relevancy in feature selection based on (partial) information decomposition. *Journal of Machine Learning Research* **24,** 1–44 (2023).

163. Treves, A. & Panzeri, S. The upward bias in measures of information derived from limited data samples. *Neural Computation* **7,** 399–407 (1995).

164. Nemenman, I., Shafee, F. & Bialek, W. *Entropy and inference, revisited* in *Advances in neural information processing systems* (eds Dietterich, T., Becker, S. & Ghahramani, Z.) **14** (MIT Press, 2001).

165. Lorenz, G. M., Martínez-Cañada, P. & Panzeri, S. in *Brain Informatics* (eds Liu, F., Zhang, Y., Kuai, H., Stephen, E. P. & Wang, H.) 33–44 (Springer Nature Switzerland, Cham, 2023).

166. Martínez-Cañada, P., Noei, S. & Panzeri, S. Methods for inferring neural circuit interactions and neuromodulation from local field potential and electroencephalogram measures. *Brain Informatics* **8,** 27 (2021).

167. Kayser, C. & König, P. Stimulus locking and feature selectivity prevail in complementary frequency ranges of V1 local field potentials. *European Journal of Neuroscience* **19,** 485–489 (2004).

168. Cardin, J. A. *et al.* Driving fast-spiking cells induces gamma rhythm and controls sensory responses. *Nature* **459,** 663–667 (2009).

169. DeFelipe, J. *et al.* New insights into the classification and nomenclature of cortical GABAergic interneurons. *Nature Reviews Neuroscience* **14,** 202–216 (2013).

170. Cavallari, S., Panzeri, S. & Mazzoni, A. Comparison of the dynamics of neural interactions between current-based and conductance-based integrate-and-fire recurrent networks. *Frontiers in Neural Circuits* **8** (2014).

171. Litwin-Kumar, A., Rosenbaum, R. & Doiron, B. Inhibitory stabilization and visual coding in cortical circuits with multiple interneuron subtypes. *Journal of Neurophysiology* **115,** 1399–1409 (2016).

172. Pfeffer, C. K., Xue, M., He, M., Huang, Z. J. & Scanziani, M. Inhibition of inhibition in visual cortex: the logic of connections between molecularly distinct interneurons. *Nature Neuroscience* **16,** 1068–1076 (2013).

173. Adesnik, H., Bruns, W., Taniguchi, H., Huang, Z. J. & Scanziani, M. A neural circuit for spatial summation in visual cortex. *Nature* **490,** 226–231 (2012).

174. Angelucci, A. *et al.* Circuits and Mechanisms for Surround Modulation in Visual Cortex. *Annual Review of Neuroscience* **40,** 425–451 (2017).

175. Urban-Ciecko, J. & Barth, A. L. Somatostatin-expressing neurons in cortical networks. *Nature Reviews Neuroscience* **17,** 401–409 (2016).

176. Veit, J., Hakim, R., Jadi, M. P., Sejnowski, T. J. & Adesnik, H. Cortical gamma band synchronization through somatostatin interneurons. *Nature Neuroscience* **20,** 951–959 (2017).

177. Brette, R. & Gerstner, W. Adaptive Exponential Integrate-and-Fire Model as an Effective Description of Neuronal Activity. *Journal of Neurophysiology* **94,** 3637–3642 (2005).

178. Descalzo, V. F., Nowak, L. G., Brumberg, J. C., McCormick, D. A. & Sanchez-Vives, M. V. Slow Adaptation in Fast-Spiking Neurons of Visual Cortex. *Journal of Neurophysiology* **93,** 1111–1118 (2005).

179. Roth, A. & van Rossum, M. C. W. in *Computational Modeling Methods for Neuroscientists* (ed De Schutter, E.) (The MIT Press, 2009).

180. Gewaltig, M.-O. & Diesmann, M. NEST (NEural simulation tool). *Scholarpedia* **2,** 1430 (2007).

181. Martínez-Cañada, P., Ness, T. V., Einevoll, G. T., Fellin, T. & Panzeri, S. Computation of the electroencephalogram (EEG) from network models of point neurons. *PLOS Computational Biology* **17,** e1008893 (2021).

182. Mitra, P. *Observed brain dynamics* (Oxford University Press, 2007).

183. Maris, E. & Oostenveld, R. Nonparametric statistical testing of EEG- and MEG-data. *Journal of Neuroscience Methods* **164,** 177–190 (2007).

184. Delorme, A. & Makeig, S. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods* **134,** 9–21 (2004).

185. Gomes, A. F. C. & Figueiredo, M. A. T. A measure of synergy based on union information. *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies* **26,** 271 (2024).

186. Faes, L., Marinazzo, D. & Stramaglia, S. Multiscale information decomposition: Exact computation for multivariate gaussian processes. *Entropy. An International and Interdisciplinary Journal of Entropy and Information Studies* **19,** 408 (2017).

187. Feller, W. *An introduction to probability theory and its applications* (Wiley & Sons, Hoboken, New Jersey, 1970).

Appendix A

Supplementary information of Chapter 2

The analyses presented here correspond to those presented as supplementary information in the second submitted version of the paper [52] while in a peer-review process.

A.1 Comparison with Other Toolboxes

Table A.2 provides a synthetic comparison of main features of different currently available toolboxes.

A.2 Description of installation and testing of MINT, and of information theoretic tools implemented in MINT

MINT can be downloaded at the public repository `github.com/panzerilab/MINT`.

Documentation on building and installing the software from source is provided as a README file that specifies the installation requirements, as well as a build file (`buildMINT.m`) designed to automate the software's compilation process. Instructions on how a user can test the software on supplied simulated test data are provided in a folder `How_to_use_MINT` in MINT's public repository, containing detailed instructions for testing it on simulated data. We also provide an additional repository `github.com/panzerilab/MINT_figures` containing the code that replicates all analyses in all figures, on both real neural data and simulated data. The dataset with CA1 neural data is provided as an attachment in Supplemental Material, and the dataset with A1 neural data is provided by the public link `doi.org/10.13016/m2yt-mfxk`.

Table A.1: Comparison of MINT with other existing information-theoretic toolboxes. Abbreviations: FIT=Feature-specific information transfer; II= Intersection Information; MI= Mutual Information; PID=Partial Information Decomposition; TE = Transfer Entropy.

Software	**Information Measures**	**Input Data**	**Signif. Testing**	**Estimation Methods**	**Bias corr**	**Dim. red.**	**Platform**
MINT (Lorenz et al., 2024) (this paper)	H, MI, TE, Info breakdown, PID, II, FIT, Cond. variants	Discrete, Continuous	Non-parametric permutation tests	Binning, Gaussian fit	✓	✓	MATLAB
HOI (Neri et al., 2024)	H, MI,Redundancy, Synergy, Netw. behavior and encoding metrics	Discrete, Continuous	X	Binning, KSG, Kernel-based, Gaussian fit	✓	✓	Python
FRITES (Combrisson et al., 2022)	H, MI, TE	Discrete, Continuous	Non- parametric, group stats	Binning, Gaussian copula	✓	X	Python
infoTheory (Climer and Dombeck, 2021)	MI (SMGM estimator)	Discrete, Continuous	X	X	X	X	MATLAB
Infotheory (Candadai, 2019)	H, MI, PID	Discrete, Continuous	X	Binning, Kernel-based	X	X	Python, C++
BROJA-2PID (Makkeh et al.,2018)	PID	Discrete	X	X	X	X	Python
Neuroscience Information Theory Toolbox (Timme and Lapish, 2018)	H, MI, TE, PID, Info transfer, Cond. variants	Discrete, Continuous	X	Binning	X	X	MATLAB
dit (James et al., 2018)	H, MI, PID	Discrete	X	X	X	X	Python
Gaussian Copula Mutual Information (Ince et al., 2017)	H, MI	Discrete, Continuous	X	Gaussian copula	✓	X	MATLAB, Python

Inform (Moore, 2017)	H, MI, TE	Discrete	X	Binning	X	X	C (Python, R, Julia, Mathematica wrappers)
ToolConnect (Pastore et al., 2016)	H, TE	Spike trains	X	X	X	X	C#
JIDT (Lizier, 2014)	H, MI, TE, Cond. variants	Discrete, Continuous	Non-parametric	Binning, Kernel-based Gaussian fit	✓	X	Java (Python, MATLAB wrappers)
MuTE (Montalto et al., 2014)	TE	Continuous	Non-parametric	Binning, Gaussian fit, Kernel-based	✓	X	MATLAB
ITE Toolbox (Szabó, 2014)	H, MI	Discrete, Continuous	X	Kernel-based	X	X	MATLAB, Python
Trentool (Lindner et al., 2011)	TE	Continuous	Non-parametric, group stats	Kernel-based	✓	X	MATLAB
Transfer Entropy Toolbox (Ito et al., 2011)	TE	Spike trains	X	X	X	X	MATLAB
Information Breakdown Toolbox (Magri et al. 2009)	H, MI, TE, Info breakdown	Discrete, Continuous	Non-parametric	Binning, Gaussian fit	✓	✓	MATLAB
PyEntropy (Ince et al., 2009)	H, MI, Max entropy	Discrete, Continuous	X	Binning, Shrink estimator	✓	X	Python
STAToolkit (Goldberg et al., 2009)	H, MI	Spike trains	Non-parametric	Binning	✓	X	MATLAB (with .mex files)

For users who prefer conducting their analysis workflows in Python, it is possible to use MINT by setting up the MATLAB Engine API for Python. This requires installing the MATLAB engine library via pip (a package manager for Python), ensuring that the library version matches the installed version of MATLAB on the system. Detailed installation instructions can be found on the MathWorks website: `mathworks.com/help/matlab/matlab_external/install-the-matlab-engine-for-python.html`. To prepare input data, the user can convert Numpy arrays to MATLAB-compatible format using the `matlab.double` conversion function. Optional arguments can be structured as Python dictionaries and used directly. Additionally, the `How_to_use_MINT` folder includes example Python scripts demonstrating how to initialize the MATLAB engine, format input data and organize the options structure in Python.

MINT provides information theoretic tools to give quantitative answers to questions about information processing when applied to single neurons, population of neurons, or to aggregate neural signals recorded across multiple areas (including LFPs, M/EEG, fMRI). The information processed by the considered neural activity can be about a specific task variable, such as a sensory stimulus, a behavioral output, or about the activity of other neurons or neural populations.

All the information theoretic quantities are functions of the joint probability, sampled across experimental trials, of observing a given value for a set of task variables (e.g. sensory stimuli, movement parameters, behavioral choices) $\mathbf{s} \in \mathbf{S}$ and of neural responses $(\mathbf{r_1}, \ldots, \mathbf{r_N}) \in \mathbf{R}_{\mathbf{1},\ldots,\mathbf{R_N}}$. Each of the variables is indicated with bold font because it can be a multidimensional vector. Importantly, each dimension in the task and neural response variables is assumed to have discrete values, so that the probabilities can be estimated by empirical occurrences. In many cases, neural data will be already discrete in nature (for example, spike counts) and the same applies to some categories of task variables like behavioral choices or identity of the presented stimulus (which experimentally usually fall into a number of discrete categories). In other cases, either task variables or neural responses will be continuous data (e.g. LFPs, etc.). These input data will be automatically discretized by MINT to perform the information calculations by specifying discretized into a finite number of bins by defining the number of bins n_{bins} (by default, 3 bins) and the binning strategy bin_method (including equi-spaced binning, equi-populated binning, and binning with user-defined bin edges; by default, no binning) as fields within the options input structure `opts`. MINT allows the use of these discretization procedures for all its information theoretic measures, and direct probability distribution sampling after discretization was used for all results in the paper. In addition to binning, MINT offers the option to compress the multi-dimensional neural activity space $(\mathbf{r_1}, \ldots, \mathbf{r_N}) \in \mathbf{R}_{\mathbf{1},\ldots,\mathbf{R_N}}$ into a dimensionality reduced representation, obtained with either supervised decoding methods or unsupervised data reduction methods, which can also be discretized and used for information calculations with the direct probability distribution sampling (see Section A.5, Fig. A.2).

The functions to compute the information quantities in MINT follow a consistent structure (Fig. A.1). The first input is the data in the form of a cell array. Optionally, the user can input a cell array of strings specifying the requested (called outputs in our

examples in the tutorials) quantities to compute, as well as a structure `opts` that contains optional arguments for the computation. The inputs are organized with the following format: {A, B, C, ...}. Outputs are specified as functions of these input variables.

The outputs of the functions are also implemented in a consistent structure. The first output variable contains cells with the requested information quantities given in the `outputs` cell array (in the same order as specified). The second output variable contains cells with the plugin information quantities (i.e., no limited-sampling bias correction) and the last output variable contains the null distribution for each specified information quantity, if the `opts` field `computeNulldist` is set to true. For instance, to compute the limited-sampling bias corrected and the plugin Mutual Information between two populations `X1` and `X2` (two-dimensional arrays with neurons in the first dimension and trials in the second dimension), the MI function is called as follows: :

```
[MI_corr, MI_plugin}] = MI({X1, X2}, {'I(A;B)'}, opts)
```

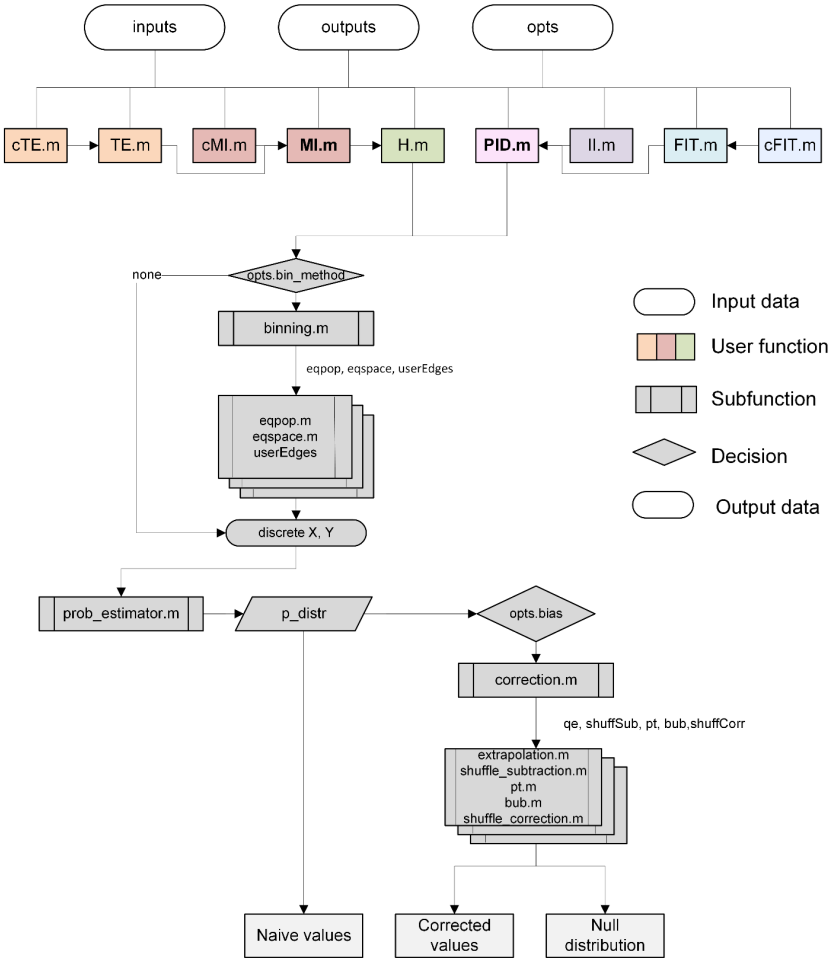

Figure A.1: The flowchart illustrates the structure and workflow of the MI module of the Toolbox, highlighting the steps involved in computing information values.

If the input data for `H.m`, `MI.m`, `cMI.m`, `PID.m` or `II.m` is given as a time series (three-dimensional array, with neuron or brain area ID in the first dimension, time points in the second dimension, and trials in the third dimension), these functions compute the information quantities for each time point and output them as time series.

In the following, we list and synthetically describe the information quantities implemented in MINT.

A.2.1 Shannon Information

The `MI.m` function computes Shannon mutual information between a population of N neurons $\{\mathbf{R_1},\ldots,\mathbf{R_N}\}$ and a task variable $\mathbf{S}$ (such as a sensory stimulus). It is a non-parametric measure that quantifies the full single-trial relationship between $\{\mathbf{R_1},\ldots,\mathbf{R_N}\}$ and $\mathbf{S}$. It is defined as [21]:

$$MI(\mathbf{S};\mathbf{R_1},\ldots,\mathbf{R_N}) = \sum_{s,r_1,\ldots,r_N} p(\mathbf{s},\mathbf{r_1},\ldots,\mathbf{r_N}) \log_2 \left(\frac{p(\mathbf{s},\mathbf{r_1},\ldots,\mathbf{r_N})}{p(\mathbf{s})\, p(\mathbf{r_1},\ldots,\mathbf{r_N})} \right) \tag{A.1}$$

where $p(\mathbf{s},\mathbf{r}_1,\ldots,\mathbf{r}_N)$ is the joint probability, sampled across experimental trials, of observing stimulus value $s \in S$ and the neural responses $(\mathbf{r}_1,\ldots,\mathbf{r}_N) \in \{\mathbf{R}_1,\ldots,\mathbf{R}_N\}$, and $p(\mathbf{s})$ and $p(\mathbf{r}_1,\ldots,\mathbf{r}_N)$ are the marginal probabilities of observing s and $(\mathbf{r}_1,\ldots,\mathbf{r}_N)$, respectively. The sum in Eq. (1) spans all possible events. $MI(\mathbf{S};\mathbf{R}_1,\ldots,\mathbf{R}_N)$ is non-negative and is zero if and only if $\{\mathbf{R}_1,\ldots,\mathbf{R}_N\}$ and $\mathbf{S}$ are independent. To compute the Shannon Mutual Information of two variables, the string to put in the outputs cell array is 'I(A;B)'. Moreover, MINT also allows to compute the mutual information between $\mathbf{S}$ and $\{\mathbf{R}_1,\ldots,\mathbf{R}_N\}$ conditioned on the activity of another population of M neurons $\{\mathbf{R}'_1,\ldots,\mathbf{R}'_M\}$ or another stimulus feature $\mathbf{S}'$ (function `cMI.m`).

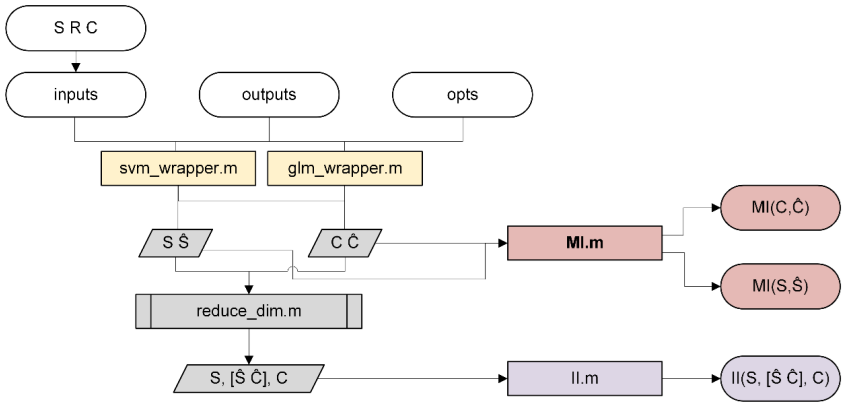

Figure A.2: Example flowchart of the pipeline module used with the MI and II functions.

A.2.2 Information Breakdown

The `MI.m` function can also compute measures that quantify how interactions between neurons contribute to the encoding of **S**. The desired quantities can be passed to the function as specific strings within the outputs cell array input. These quantities include the co-information coI, the difference between the population information and the sum of single-neuron stimulus information quantifying the overall contribution of interactions to population encoding. Additionally, the function can return information breakdown terms [33] quantifying how pairwise correlations contribute to redundancy-synergy index [132]. These terms can also be specified in outputs and include the signal-similarity $I_{sig-sim}$ (output name `'Iss(A)'`) (contribution of the similarity across neurons of trial-averaged responses to different stimuli), the stimulus-independent correlations $I_{cor-ind}$ (contribution of the interplay between the signs of signal similarity and of noise correlations, output name `'Ici(A;B)'`), and the stimulus-dependent correlations $I_{cor-dep}$ (quantifying how much information is gained by the stimulus-modulation of noise correlations, output name `'Icd(A;B)'`).

A.2.3 Partial Information Decomposition (PID)

The `PID.m` function computes PID components. PID decomposes the information jointly carried by a set of source variables (for us, a set of simultaneously recorded neurons; the first N variables in the inputs cell array) about a target S (for us, a sensory stimulus or a behavioral choice; the last variable in the inputs cell array) into non-negative components that capture information about the target that is either redundantly encoded across sources, uniquely encoded by a single source, or synergistically encoded by the combination of sources. For more than two source variables, such components can also represent combinations of redundancy, synergy, and unique information.

In the case of two source variables, once the individual Mutual Information between the target and each source, the joint Mutual Information between the target and the two sources, and one of the PID components (e.g., redundancy) are all computed, algebraic linear relationships (derived from PID "lattices") allow for the computation of all remaining PID components. Thus, a PID with two sources is defined by the choice of a specific redundancy measure. The desired PID components can be passed to the function as strings within the outputs cell array input ('Red' for redundancy, 'Syn' for synergy, 'Unq1' and 'Unq2' for the unique information carried by the first or the second input source, respectively).

For more than two variables, defining a redundancy measure is enough to compute each PID component as a linear combination of redundancy terms (the output name 'PID_atoms' provides all PID components). Therefore, different methods to decompose information differ and are defined by the measure of redundancy they use (which can be specified as the `redundancy_measure` field in the `opts` structure). In MINT, we implemented three possible measures of PID redundancy, which are very popular and respect the so-called pairwise marginal property (redundancy is invariant for distributions preserving the pairwise marginals between each source and the target)[145].

A.2.3.1 I_{min}

We implemented Williams and Beer's PID original redundancy measure called I_{min}. This measure quantifies the information redundantly encoded about S across N source variables $\{\mathbf{R_1}, \ldots, \mathbf{R_N}\}$ as:

$$\mathrm{Red}\left(\mathbf{S}:\mathbf{R_1};\mathbf{R_2};\ldots;\mathbf{R_N}\right) = I_{min}\left(\mathbf{S}:\mathbf{R_1};\mathbf{R_2};\ldots;\mathbf{R_N}\right) = \sum_{\mathbf{s}\in\mathbf{S}} p\left(\mathbf{s}\right) \min_{R_i\in\{\mathbf{R_1},\mathbf{R_2},\ldots,\mathbf{R_N}\}} MI\left(\mathbf{S}=s;R_i\right) \tag{2}$$

where $I\left(\mathbf{S}=\mathbf{s};\mathbf{R_i}\right)$ represents the specific information that source variable $\mathbf{R_i}$ provides about a specific value s of the target variable S. I_{min} captures redundancy as the similarity across the source variables $\mathbf{R_i}$ in distinguishing individual values of S.

A.2.3.2 Minimum Mutual Information (I_{MMI})

MINT also implements the PID based on the redundancy measure introduced in [110], called I_{MMI}. This measure quantifies the information redundantly encoded about S across N source variables $\{\mathbf{R_1}, \ldots, \mathbf{R_N}\}$ as:

$$\mathrm{Red}\left(\mathbf{S}:\mathbf{R_1};\mathbf{R_2};\ldots;\mathbf{R_N}\right) = I_{MMI}\left(\mathbf{S}:\mathbf{R_1};\mathbf{R_2};\ldots;\mathbf{R_N}\right) = \min_{\mathbf{R_i}\in\{\mathbf{R_1},\mathbf{R_2},\ldots,\mathbf{R_N}\}} MI\left(\mathbf{S};\mathbf{R_i}\right) \tag{3}$$

Therefore, I_{MMI} captures redundancy as the minimum amount of information encoded about S by any of the source variables.

A.2.3.3 BROJA

Finally, MINT implements the PID based on the redundancy measure termed BROJA [31], which defines redundancy about S between two source variables $\mathbf{R_1}$ and $\mathbf{R_2}$ as the result of a constrained optimization problem:

$$\mathrm{Red}\left(\mathbf{S}:\mathbf{R_1};\mathbf{R_2}\right) = I_{BROJA}\left(\mathbf{S}:\mathbf{R_1};\mathbf{R_2}\right) = MI\left(\mathbf{S};\mathbf{R_1}\right) + MI\left(\mathbf{S};\mathbf{R_2}\right) - \min_{q\in\Delta_P}\left(MI_q\left(\mathbf{S};\mathbf{R_1},\mathbf{R_2}\right)\right) \tag{A.2}$$

where Δ_P is the space of distributions preserving pairwise marginals between individual source variables and the target variable. BROJA's advantage is that it is additive for independent systems of sources and targets [111], however it is only defined for two source variables. The numerical calculation of Eq. A.2 is performed by MINT through the conic optimization Embedded Conic Solver (ECOS) algorithm. The use of ECOS requires the installation of a C library, which can be implemented by either compiling the C source code with a MATLAB-compatible C Compiler (using MINT's `BuildAndTest.m` function) or copying locally the precompiled binaries that MINT provides for Linux, Windows, and macOS.

A.2.4 Information Transmission Measures

MINT provides a number of measures to quantify the overall or feature-specific directed information transmitted from a putative sender region **X** to a receiver region **Y** from which neural activity was simultaneously recorded. These measures are all based on the Wiener-Granger causality principle, which states that a region **X** causally influences a region **Y** if the past state of **X** predicts the present state of **Y** at time t beyond what can be predicted by the past state of **Y**.

A.2.5 Transfer Entropy (TE)

Transfer Entropy from **X** to **Y**, denoted as $TE(\mathbf{X} \rightarrow \mathbf{Y})$[112], measures the overall information transmitted from region **X** to region **Y** and can be computed using the `TE.m` function in the toolbox. $TE(\mathbf{X} \rightarrow \mathbf{Y})$ is defined as the conditional Mutual Information between the past activity of the sender region $\mathbf{X}_{\text{past}}$ and the present activity of the receiver region $\mathbf{Y}_{\text{pres}}$ given the past of the receiver $\mathbf{Y}_{\text{past}}$. Moreover, MINT also allows for the computation of TE from **X** to **Y** conditioned on the activity of a third node **Z** (function `cTE.m`; **Z** can, in principle, also be the multivariate activity of a set of regions).

Although in the example computations of TE provided in Figs. 2.4, A.5, and A.6 we computed the present of **Y** at a single time point t and the past of **X** and of **Y** at individual time points lagged by a delay Δt, MINT also allows for computing information transfer measures for present and/or past activity as multidimensional variables, potentially spanning several time points. The past-time embedding is specified by the parameters opts `(('tpres'))` and opts `('tau')`. The parameter `('tpres')` specifies the present timepoint in the calculation. The parameter `('tau')` defines the delay (or an arbitrary set of delay numbers if more past points are to be considered) relative to `('tpres')`, expressed as an integer number indicating how many timepoints back the past timepoint is set. These parameters can be specified individually for X and Y, allowing the function to use different timepoints for the present and past for each of X and Y. This allows easy implementation of various embedding techniques such as those described in [73]. Because of the problem of data dimensionality (the bias can be corrected well if the product of the number of bins for the past of X and the present of X and Y is several times smaller than the number of time samples), many studies (e.g. [75]) use only one time point compute the past of X and Y. Some studies [73, 81] have proposed to use a delay of one time-step for the past of the putative receiver Y, to be as conservative as possible when conditioning away the information of the putative receiver Y. Our own recommendation is to consider and plot a wide range of delays, as we do in our analyses (see e.g. Fig. F panel B), to get a better feeling of the robustness of the results as a function of these parameters.

The string to define in reqOutputs to compute Transfer Entropy from the data in the first to the data in the second element of the input cell is `'TE(A->B)'`. In addition, a second output can be requested with `'TE(B->A)'`, in order to reduce computational effort and explore data more efficiently by calculating both directions of Transfer Entropy with a single function call.

A.2.6 Feature-specific information transfer (FIT)

Feature-specific information transfer from **X** to **Y** about a stimulus feature **S**, denoted as $FIT\,(\mathbf{S} \rightarrow \mathbf{X} \rightarrow \mathbf{Y})$, measures the information transmitted from **X** to **Y** about a specific feature **S** and can be computed using the `FIT.m` function in the toolbox. $FIT\,(\mathbf{S} \rightarrow \mathbf{X} \rightarrow \mathbf{Y})$ is defined as the minimum between two PID terms with similar but slightly different interpretations. The first term is the information about **S** that is redundant between $\mathbf{X}_{\text{past}}$ and $\mathbf{Y}_{\text{pres}}$, and is unique to any information encoded by $\mathbf{Y}_{\text{past}}$. The second term is the information about the present activity of the receiver $\mathbf{Y}_{\text{pres}}$ that is redundant between $\mathbf{X}_{\text{past}}$ and **S**, and is unique to any information encoded by $\mathbf{Y}_{\text{past}}$.

To guarantee the nonnegativity of FIT, both terms are computed using the I_{min} measure. Minimizing between the two terms ensures key FIT properties, including that it is upper bounded by the feature information encoded by the past of region **X**, $I\left(\mathbf{S};\mathbf{X}_{\text{past}}\right)$, the information encoded by the present of region **Y**, $I\left(\mathbf{S};\mathbf{Y}_{\text{pres}}\right)$, and by the overall information transmitted from **X** to **Y**, $TE\,(\mathbf{X} \rightarrow \mathbf{Y})$. Moreover, MINT allows for the computation of conditional FIT ($cFIT$, using the `cFIT.m` function), to remove from $FIT\,(\mathbf{S} \rightarrow \mathbf{X} \rightarrow \mathbf{Y})$ the component potentially routed through the past activity of a third recorded region **Z** (where **Z** can, in principle, also be the multivariate activity of a set of regions). $cFIT\,(\mathbf{S} \rightarrow \mathbf{X} \rightarrow \mathbf{Y}|\mathbf{Z})$ is defined as $FIT\,(\mathbf{S} \rightarrow \mathbf{X} \rightarrow \mathbf{Y})$ minus a term capturing feature information transmitted from **X** to **Y** that is also redundantly encoded by the past of **Z**[38].

Similar to TE, the past-time embedding is specified by the parameters opts (‘tpres’) and opts (‘tau’) to specify the timepoints taken as present and past for the computation. Our own recommendation is to consider and plot a wide range of delays, as we do in our analyses (see e.g. Fig. F panel B), to get a better feeling of the robustness of the results as a function of these parameters.

Similar to the TE function, the FIT function can compute bidirectional information transfer about a target by specifying `‘FIT(A->B;S)’` and `‘FIT(B->A;S)’` in the outputs cell array. The same applies to $cFIT$ in `cFit.m` (specifying `‘cFIT(A->B;S|C)’` and `‘cFIT(B->A;S|C)’` in the outputs cell array).

A.2.7 Intersection Information (II)

MINT implements also Intersection Information (II, computed using the `II.m` function), a measure quantifying the amount of sensory information encoded by neural activity that is used to inform behavioral choices. Intersection Information quantifies the part of information in neural responses that is common to both stimulus and choice information. Intersection Information is computed as the minimum between two PID terms with similar but slightly different interpretations. The first term is the information about choice **C** redundant between stimulus **S** and neural response **R**. The second term is the information about stimulus **S** redundant between choice **C** and neural response **R**. By default, these terms are computed using the BROJA redundancy measure. Minimizing between the two terms ensures that Intersection Information satisfies key properties that would be expected from a measure with this interpretation, including that independent **S** and **R** imply null in-

tersection information, that II satisfies the data processing inequality, and that Intersection Information is upper bounded by both $MI(\mathbf{R};\mathbf{S})$ and $MI(\mathbf{R};\mathbf{C})$.

The name to put in the outputs cell array to compute Intersection Information is `'II(A,B,C)'`, which computes the amount of information of input A encoded in input data B and that is also present in input data C.

A.3 Information estimators and limited-sampling bias corrections

MINT implements several types of information estimations. Methods based on estimating probabilities of discrete data or through data discretization include the plug-in method and its bias corrections (Shuffle, QE, Shuffle QE, Panzeri-Treves, Ish and BUB). Methods not requiring discretization of data to estimate probabilities include the NBS and KSG methods.

MINT implements several limited-sampling bias corrections of information theoretic quantities. For Shannon Entropy, Mutual Information, and for the Information Breakdown terms, bias-corrected estimates are computed separately for each quantity. For the PID calculation, bias-corrected estimates can be obtained either by correcting each PID atom individually, or alternatively by correcting for the bias first for the individual and joint mutual information term and one of the PID terms (e.g., redundancy or synergy), and then by correcting for the bias of the other PID components by using algebraic relationships derived from PID "lattices" (see A.3.7). By default, no bias correction is computed (uncorrected, or naïve, information quantities).

A.3.1 Shuffle Limited-sampling bias correction

When this bias correction option is called (setting the bias field in the options structure `opts` to `'shuffSub'`), the bias is estimated by computing the information values after destroying all genuine stimulus information by randomly permuting the stimulus-neural response association in each trial[39, 40], and then it is subtracted out from the original estimate to provide the bias-corrected estimate. MINT allows subtracting these bias estimates from all implemented information theoretic quantities. A non-zero integer in the field `shuff` of the options structure `opts` specifies the number of shuffles to be performed and averaged over to obtain this estimate (by default, 20 shuffles).

A.3.2 QE correction

The Quadratic Extrapolation (QE) procedure [40, 41] (setting the bias field in the options structure `opts` to `'qe'`) assumes that the estimation is performed in a regime with large numbers of trials and approximates the bias of the information quantities as a second-order expansion in the inverse of the number of available trials [42]. This procedure first re-computes the information from fractions (halves and quarters) of the data available

and then fits the dependence of information estimates on the inverse number of trials to a quadratic function. This quadratic fit is then used to estimate the bias-corrected information value as the value that would be obtained with the quadratic scaling law if an infinite number of trials were available (i.e., the intercept term of the fit). MINT allows the use of this bias correction with all its information theoretic measures. Field `xtrp` in the options structure `opts` specifies the number of repetitions of the extrapolation procedure. The function performs the specified number of extrapolations and calculates the final corrected value as the mean of the estimates (by default, 10). Note that MINT allows the option to perform a linear (rather than quadratic) extrapolation, which uses only halves and not quarters of the data. This may be convenient when very small datasets are available and the division into quarters is problematic.

A.3.3 Shuffle-QE correction

MINT also implements a bias correction combining both Shuffle and QE (setting the bias field in the options structure `opts` to `'qe_shuffSub'`). This procedure first computes both the original and the shuffled values, performs QE on both as explained above, and then computes the unbiased estimate by subtracting the QE-corrected shuffled value from the QE-corrected non-shuffled value. The parameters `shuff` and `xtrp` can be set in the options `opts` structure as mentioned above.

A.3.4 Panzeri-Treves correction

This correction technique analytically approximates the linear term of the bias expansion in the inverse of the number of available trials, which is then subtracted from the measured information to obtain bias-corrected information values [42]. The estimation of the bias depends only on the number of response bins with a non-zero probability of being observed, which is estimated using a Bayes approach. It is available only for Shannon Entropy and Mutual Information (setting the bias field in the options structure `opts` to `'pt'`) but not for PID-based quantities.

A.3.5 Ish bias reduction procedure for multi-dimensional data

The Ish procedure is relevant for the reduction of the bias of the information about a task variable (say stimulus **S**) carried by the joint observation of a multivariate neural response with dimension N (e.g., the activity of N neurons). It adds and subtracts to the definition of mutual information two entropy terms which have equal asymptotic value (in the case of exact sampling of the probabilities with an asymptotically large number of trials). For a limited number of trials, the difference between these two terms provides a negative contribution to the mutual information bias. Thus, Ish has a considerably smaller bias (though larger variance) than the direct estimate of mutual information from Eq. (1). Another interesting property of Ish is that it typically has a negative bias, whereas direct estimates of mutual information typically have a positive bias. Thus, the joint calculation of mutual information from Eq. (1) and I_{sh} allows the estimation of upper and

lower bounds to the real information values. This procedure can be applied to the mutual information and to the information breakdown (where it allows computation of upper and lower bounds of some terms) (setting the bias field in the options structure `opts` to `'shuffCorr'`). It can be combined with the QE and the shuffle-subtraction corrections.

A.3.6 Best Universal Bound procedure

This method developed by [43] expresses the information estimation as a polynomial approximation problem, allowing the computation of 'Best Universal Bounds' on the information bias and variance (setting the bias field in the options structure `opts` to 'bub'). Its results depend on the selection of a parameter `kmax` (which is related to degrees of freedom).

A.3.7 Options for PID bias

For QE, shuffle subtraction, and shuffle-QE subtraction applied to PID with two source variables, two bias-correction options are available. The first option is to correct for the bias, with the chosen bias correction procedure, each PID term individually. The second option is to correct for the bias in a way that respects the linear relationships between the PID terms and Shannon information quantities derived from the so-called PID lattices. This is done by correcting for the bias of the individual and joint mutual information term and one PID component of choice, and then correcting for the bias of the other PID components by using the algebraic relationships derived from PID "lattices". The chosen PID atom can be specified with the field `chosen_atom` in the options structure `opts` (by default, synergy). This option is implemented by default (setting the `opts` field 'pidConstrained' to true). If one chooses not to, each PID atom is corrected individually.

A.3.8 Estimators not requiring discrete or discretized data

SM 3.2.1 Kraskov-Stögbauer-Grassberger (KSG) estimator The KSG estimator uses the Kozachenko-Leonenko kth-nearest-neighbor entropy estimator to find structures in the underlying probability distribution [119]. The estimator is included in the toolbox using scripts from the improved version of KSG implemented in [120], which was shown to work very well for real-valued data with am most a handful of dimensions. By varying the parameter k which determines the nearest-neighbor statistics scale, one can also change the scale in which the algorithm looks for the underlying structure. The method can be used by setting the bias field in the options structure opts to `'ksg'` and the parameter k (by default, 6) can be changed by specifying the `k_ksg` field of the opts structure.

SM 3.2.2 Nemenman-Shafee-Bialek (NSB) estimator The NSB estimator is a Bayesian entropy estimator that is designed to work on prior distributions of the stimulus-response distributions that are almost uniform in their expected entropies [121]. This means that the entropy estimate is not strongly biased by the prior assumptions. The method does not require free parameters to be inputted and can be used by setting the bias field in the options structure opts to 'nsb'.

A.4 Hierarchical Permutations for Statistical Testing

A.4.1 Data Shuffling

Shuffling neural data is a useful tool to test hypotheses and gain insights into the information encoding structure of neural populations and how information is transferred. The `hShuffle.m` function provides a range of hierarchical data shuffling methods, allowing for the disruption of neural correlations, temporal patterns, or stimulus information. Each data feature (a neural activity dimension or a task variable) can be shuffled either unconditionally on any other variable or conditionally on the values of other variables.

For example, neural responses can be shuffled unconditionally on any other variable to destroy the information they carry about the stimulus (e.g., to shuffle the first two input variables across trials, the output name is 'AB' and the `opts` field 'dim_shuffle' is set to 'Trials'). Alternatively, neural responses of different neurons can be shuffled conditionally on the stimulus values to provide surrogate data that preserve single-neuron stimulus information but destroy noise correlations (correlations at fixed stimulus between different neurons). Shuffling the neural responses across trials with the same stimulus while keeping the position of each timepoint fixed provides surrogate data that retain time-resolved stimulus information of single neurons but disrupt across-time correlations. For example, to shuffle the first input variable conditioned on the second and third input variables, the outputs cell is 'A_BC' (the variable(s) to condition the shuffling on are specified after the underscore), and the `opts` field 'dim_shuffle' is set to 'Trials'.

Shuffled data are either used within a given function (e.g., `MI.m` or `FIT.m`) to output null hypothesis values or to be separately provided as input to any function in MINT to construct user-defined non-parametric null distributions to empirically estimate the p-value of measures computed from the original data.

Furthermore, MINT enables efficient computation of group-level averaged shuffled quantities by recombining shuffled information values across experiment repetitions using the `create_nullDist_groupLevel` function. This function accepts measures calculated from M independent data shuffles across N experiment repetitions and outputs K distinct realizations of the permuted group average.

A.4.2 Cluster Permutation for Multiple Comparison Correction

It is often important to detect significant information encoded or transmitted across data points that are correlated due to physical proximity (e.g., in space and time, or time and communication delay). MINT implements the rigorous detection of clusters of adjacent significant information values via cluster permutation tests [87, 183] (the `clusterStatistics.m` function). The `clusterStatistics.m` function takes as input a matrix of information values computed from the original data across adjacent space and time points and a set of M analogous matrices obtained from shuffled data. The function computes a cluster-forming threshold as a specified percentile `clusterPercentilThreshold` (provided as input) of the shuffled information values. This threshold is calculated either by pooling shuffled values across all samples (when the input `pool` parameter is set to 1) or indepen-

dently for each sample (when `pool` is set to 0). The latter option provides less statistical power but is recommended when shuffled information is non-stationary over space and time.

The procedure then identifies clusters in both the original and shuffled data by connecting adjacent information values that surpass the cluster-forming threshold and computes the mass of each cluster as the sum of its information values. A cluster-level null distribution is created by taking the maximum cluster mass from each shuffled dataset. Finally, clusters in the original dataset that exceed a specified percentile `significanceThreshold` (provided as input) of this null distribution are classed as significant.

A.5 Interfacing with Dimensionality Reduction Methods

To allow information analysis on datasets with a high number of dimensions, we offer several possibles wrappers that integrate tools or dimensionality reduction in a way that permits the calculation of information measures on processed variables with less dimensionality.

A.5.0.1 Interfacing Information Calculation with Supervised Dimensionality-Reduction Method

Here we describe which supervised dimensionality reduction algorithm we implemented in MINT and we interfaced then with the information theoretic algorithms.

Our routines take input data in the form of a cell array with neural responses $\{\mathbf{r}_1, \ldots, \mathbf{r}_N\} \in \mathbb{R}^N$ across all trials in the first element and the task variables $\mathbf{s} \in \mathbf{S}$ (e.g., sensory stimuli, movement parameters, behavioral choices) across all trials in the second element. It returns an array across all trials of dimensionality reduced representation $\widehat{\mathbf{r}}$ of $\{\mathbf{r}_1, \ldots, \mathbf{r}_N\}$. The dimensionality-reduced representation $\widehat{\mathbf{R}}$ is computed by cross-validated decoding of the behavioral variables $\mathbf{s}$ given the neural data, so that the representation $\widehat{\mathbf{r}}$ for each trial is computed from trials held out from the decoder's training process. The reduced neural representation data are then fed as neural data input to the information calculation routines.

The representation $\widehat{\mathbf{R}}$ obtained through the supervised decoding method can be understood as a representation of the neural activity data that is lower dimensional (typically, one dimensional) but that still captures efficiently the information about $\mathbf{s}$ provided by the joint neural responses. The representation $\widehat{\mathbf{R}}$ can take the form of the value of the behavioral variable decoded as most likely from neural activity (which can be directly fed to the information calculation routines, an approach that is equivalent to computing information from the confusion matrix of the decoder [6]), or the posterior probability of the task variable value given the considered neural activity (which can then be binned and fed to the information calculation routines).

Supervised models implemented in MINT are:

A.5.0.2 Support Vector Machines (SVM)

Support vector machines (SVM) are supervised machine learning methods that find the optimal hyperplane(s) in the data space (in our case, neural data $\{\mathbf{r}_1, \ldots, \mathbf{r}_N\}$) to classify the labels (in our case, $\mathbf{s}$). MINT provides a function `svm_wrapper.m` that trains and tests the SVM with either linear or RBF kernel, using either `fitcsvm.m` (Statistics and Machine Learning Toolbox in MATLAB) or the libsvm toolbox [123] as the underlying SVM implementation. The first element of the output cell array is the lower dimensional representation $\widehat{\mathbf{R}}$ as a list of the cross-validated predicted labels across trials. The second element of the output cell array is the representation $\widehat{\mathbf{R}}$ as a list of cross-validated posterior probabilities of $\mathbf{s}$ given the neural responses in that trial. The function can also output the weights of the decoding model for the linear SVM (which can be used, e.g., to calculate the angle between boundaries as in Fig. A.4) or the trained hyperparameters, in case they were optimized.

A.5.0.3 Generalized Linear Model (GLM)

Generalized linear models are an extension of linear models that can incorporate non-Gaussian-distributed data (including discrete data). In MINT, the `glm_wrapper.m` function trains and tests GLM models using lasso, ridge, or elastic net regularization and either `lassoglm.m` (Statistics and Machine Learning Toolbox in MATLAB) or the glmnet toolbox [124]. This function allows for the training and testing of GLM models with optional regularization methods, including lasso, ridge, or elastic net. It provides flexibility in choosing regularization types and additional options for k-fold cross-validation. Depending on the output list option chosen, it outputs as $\widehat{\mathbf{R}}$ a list (for each trial of the test set) of the predicted labels, or the posterior probabilities of $\mathbf{s}$ given the neural data in that trial.

A.5.0.4 Interfacing Information Calculation with Unsupervised Dimensionality-Reduction Method

Unsupervised methods transform neural data into a lower dimensional output $\widehat{\mathbf{R}}$ that still approximates the data well. Our routines take input data in the form of a list across all trials of neural responses $\{\mathbf{r}_1, \ldots, \mathbf{r}_N\} \in \mathbb{R}^N$, as well as the desired dimensionality of the reduced representation. It returns a list across test-set trials of dimensionality reduced representation $\widehat{\mathbf{r}}$ of $\{\mathbf{r}_1, \ldots, \mathbf{r}_N\}$. The reduced neural representation data are then fed as neural data input to the information routines for information calculation. Unsupervised models implemented in MINT are:

A.5.0.5 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a dimensionality reduction technique that projects the data onto a lower dimensional space with maximal variance (REF). MINT's function `pca_wrapper.m` outputs in each trial the coefficients of the data along the selected number of principal components. It offers the option to perform both cross-validated and non-cross-validated principal component analysis.

A.5.0.6 Non-negative Matrix Factorization (NMF)

Non-negative Matrix Factorization (NMF) is another technique of dimensionality reduction that projects the data onto a lower-dimensional space that still describes the data well. Unlike PCA, it does not require different components to be orthogonal, but requires that the decomposition is performed with nonnegative coefficients and basis functions (which is recommended for reducing the dimensionality of inherently nonnegative data, such as spike counts). MINT's function `nmf_wrapper.m` outputs in each trial the non-negative coefficients of the data along the selected number of principal components. It offers the option to perform both cross-validated and non-cross-validated principal component analysis.

A.6 Details of Simulations

A.6.1 Simulation of Neural Populations Information Encoding

This section presents a detailed description of the simulation and the analysis of information encoding in neural populations presented in Fig. 2.2.

We simulated two scenarios which capture two main ways in which correlations have been reported to influence population coding [53]. For each scenario, we simulated correlated spike trains of a neural population of $N = 20$ neurons responding to two simulated stimuli (200 trials per stimulus, 10 simulation repetitions).

The strength of correlations between neurons was modulated by generating responses to each stimulus as the sum of an independent Poisson process (independent outcome for each neuron) and a shared Poisson process (same outcome across neurons), adjusting the pairwise Pearson noise correlation for each stimulus by varying the contribution of the shared and independent processes to the spike trains. Thus, the spike count of neuron i $(i = 1,2)$ was generated as:

$$r_i(s) = r_{i-\text{individual}}(s) + r_{\text{shared}}(s) \tag{6}$$

where $r_{i-\text{individual}}(s)$ and $r_{\text{shared}}(s)$ are the output of 3 independent Poisson processes for each stimulus, with mean count parameter indicated by the corresponding name.

The first scenario (Fig. 2.2A) was implemented with strong stimulus modulation of the correlation strength, resulting in information-enhancing noise correlations (pairwise Pearson noise correlation 1 for stimulus 1 and 0 for stimulus 2). The individual and shared processes were created such that the resulting total firing rate of each of the neurons is constant across stimulus values, so only the firing correlation is informative about the stimulus. The parameters were $r_{i-\text{individual}}(s = 1) = 1$ sp/s, $r_{\text{shared}}(s = 1) = 1$ sp/s, $r_{i-\text{individual}}(s = 2) = 2$ sp/s, $r_{\text{shared}}(s = 2) = 0$ sp/s.

For the second scenario (Fig. 2.2B), we simulated information-limiting noise correlation. Namely, we simulated a population of neurons all with the same stimulus selectivity (lower spiking rate to the first stimulus and a higher spiking rate to the second stimulus, thus positive signal correlations) and with positive noise correlations that were only

weakly stimulus-dependent (pairwise Pearson noise correlation 0.2 for stimulus 1 and 0.1 for stimulus 2). The parameters were $r_{i-\text{individual}}(s=1) = 0.8$ sp/s, $r_{\text{shared}}(s=1) = 0.2$ sp/s, $r_{i-\text{individual}}(s=2) = 1.9$ sp/s, $r_{\text{shared}}(s=2) = 0.1$ sp/s.

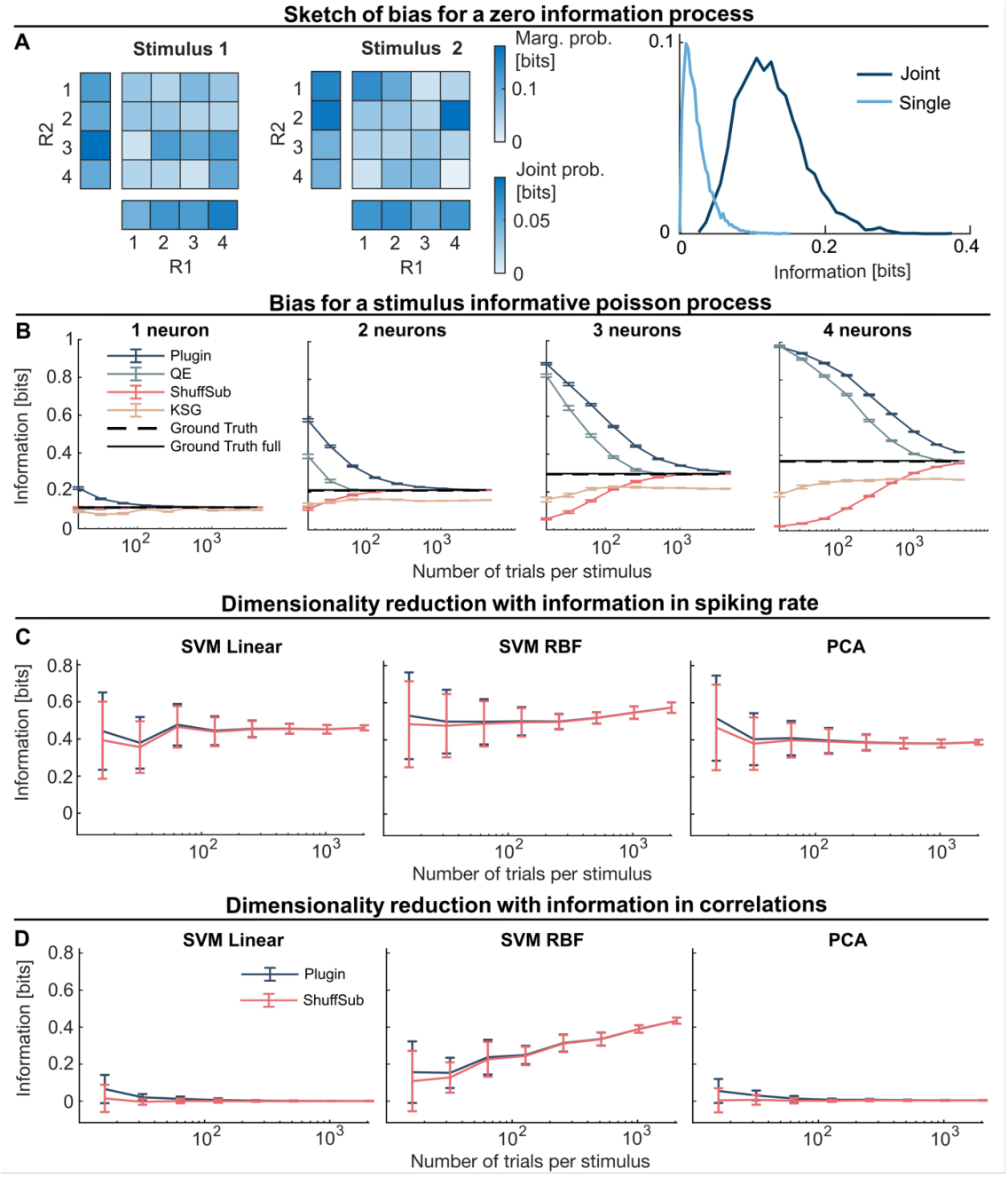

Figure A.3: **A.** Schematic illustration of the limited-sampling bias problem. Two uninformative neurons, responding on each trial with a uniform distribution of spike counts ranging from 1 to 4, regardless of which 2 stimulus values were presented. The empirical response probability heatmaps sampled from 50 trials per stimulus are shown in the left and middle columns (responses to stimuli 1 and 2, respectively). Side heatmaps indicate the marginal probability values. Because of limited sampling, the neural response probabilities look different across stimuli (even if they are not), and much more so for the joint probability than for the marginal response probabilities. Right: distribution (over 5,000 simulations) of the plugin information values obtained with 50 trials per stimulus. Although the information should be zero, it is > 0 because of the random sampling variations illustrated in panel A, left. The bias is the (non-zero) average value of this distribution, as the true asymptotic value should be zero. The bias is larger for the joint information (when neural activity is 2-D) than for the single neuron information (when neural activity is 1-D). **B.** Mean $\pm$ SEM over 25 simulations, each performed with the number of trials per stimulus shown on the x axis, of Mutual Information MI(R;S) between a binary stimulus S and a neural response R made of N Poisson uncorrelated neurons, with N varying from N=1 to N=4 from left to right. (see SI Section SM6.4). Spike counts of each neuron are discretized into R =5 bins (0, 1, 2, 3, >3 spike counts), and then information is computed with four different estimators (plugin, QE, shuffle-subtracted, KSG). We also plot the ground-truth value of information computed from the analytical form of the probability distributions of the Poisson spike counts either unbinned (full line, representing the exact value of information in the simulated process) or binned as described above (dashed line). The QE bias correction was done with 10 repetitions (xtrp=10). The shuffle-subtraction was done with 30 shuffles (shuff= 30). Application of QE or shuffle-subtraction bias correction to the plugin estimators provides an effective elimination of the limited-sampling bias for realistic number of trials < 100) for N=1,2 but not for $N > 2$. The KSG method does not converge to the ground-truth information value even for large trial numbers, with a data-processing bias that grows with N. **C.** Mean ± SEM over 25 simulations, each performed with the number of trials per stimulus shown on the x axis, of Mutual Information MI(R;S) between a binary stimulus S and a neural response R made of the spike counts of N=20 correlated neurons, simulated as in Fig. 2.2B (see Section SM6.1). In this simulation, information is carried by design by the difference across stimuli in spike counts of individual neurons. The activity of the 20 neurons is reduced to one dimension using the `svm_wrapper.m` cross-validated with 2 folds when using the linear or RBF SVM or the `pca_wrapper.m` when considering PCA. The averaged information values do not depend much on the sample size and are almost identical between corrected and uncorrected (plugin) estimates, suggesting that the dimensionality reduction has little bias and that calculation of information from a population of 20 neurons can be performed robustly with limited data sizes when using dimensionality reduction. **D.** Same as panel C, but for a population of N=20 neurons carrying information only by correlations without spike count modulations, simulated as in Fig. 2.2A. Here the information can be recovered only with a complex non-linear decoder (RBF SVM) for any considered number of trials. Limited sampling bias appears small but the increase of RBF SVM suggests that more complex supervised decoders benefit from more training data to reduce the data processing bias.

A third scenario was used for the simulations of the bias in Fig. A.3 panel B and is described next. We simulated a population of Poisson neurons whose mean rate was stimulus-selective similar to the previous stimulation (lower spiking rate to the first stimulus and a higher spiking rate to the second stimulus). The parameters were $r_(i-individual)(s=1)=1sps$, $r_shared(s=1)=0sps$, $r_(i-individual)(s=2)=2sps$, $r_shared(s=2)=0sps$.

Since we did not add any shared process, the neurons were independent from each other at fixed stimulus and there was no noise correlation. This facilitate the numerical computation of the ground-truth values of information for this process, which were used in the study of the bias properties.

The generated spike counts were binned into 5 bins, by leaving spike counts ≤ 4 untouched and setting to 4 all spike count values ≥ 5 (this was done by setting in input options `opts` of `MI.m` the `binning_method` field to `'userEdges'`, which allows binning the data with user-defined bin edges). All MI and PID values were corrected for the limited-sampling bias by using the shuffle-subtraction procedure implemented in the toolbox (averaged over 30 shuffles).

We used the `svm_wrapper.m` function of the toolbox to predict the stimulus based on the population activity by fitting a cross-validated Support Vector Machine (SVM) with 5 folds. Two distinct kernel functions were employed to fit the SVM on the data: linear and radial basis function (RBF). We performed hyperparameter optimization, tuning the parameters C for linear SVM and C and γ for SVM RBF using 5-fold cross-validation. We used Bayesian optimization across a logarithmic range between 1×10^{-3} and 1×10^{3} and a maximum number of iterations of 30. To evaluate the role of correlations in information encoding, we computed the mutual information of the predicted and the true stimuli. To eliminate noise correlations, pseudo-responses were generated by shuffling the simulated response conditionally on the stimulus value, so they have the same single-cell properties as the original data but no noise correlations (using the `shuffle.m` function with output `'A_B'`). We computed the Mutual Information using the `MI.m` function of the toolbox between the actual stimulus and the one predicted from all above-described decoders (linear and RBF kernel SVM for simulated response and pseudo-response) and compared them to gain insights into the effects of noise correlation on the population information.

A.6.2 Simulation of Encoding and Readout of Information in Pairs of Neurons

This section presents a detailed description of the simulation and analysis of information encoding and readout from pairs of neurons presented in Fig. A.4.

We simulated a pair of neurons, independently encoding a binary stimulus $S = \{-1, 1\}$ across 1000 trials. The single-trial firing rate of each neuron $i \in \{1, 2\}$ was determined by a Poisson process $r_i(S) \sim \text{Poisson}(\lambda_i(S))$, with the intensity parameter $\lambda_i(S)$ depending on the stimulus as $\lambda_i(S) = \lambda_0 + \Delta \cdot W_{\text{enc},i} \cdot S$. Here, λ_0 determined the mean firing rate of each neuron across trials, Δ was the separation in the mean firing rates across the two stimuli, and $W_{\text{enc},i}$ was the element i of the 2-dimensional encoding vector $\bar{W}_{\text{enc}}$ determining the tuning of each neuron to the stimulus. In our simulations, we set $\lambda_0 = 4$, $\Delta = 1$, and $\bar{W}_{\text{enc}} = (1, 1)$, so that both neurons had lower firing rates for $S = -1$ and higher firing rates for $S = 1$.

On each trial, we simulated a choice variable C by taking the dot product between the population firing rates and a decoding vector $\bar{W}_{\text{dec}}$, such that

$$C = \bar{r} \cdot \bar{W}_{\text{dec}}$$

and binarized C using equi-populated binning. We obtained the decoding vector $\bar{W}_{\text{dec}}$ by applying the standard two-dimensional rotation matrix $R_2(\theta)$ to the encoding vector $\bar{W}_{\text{enc}}$ as follows:

$$\bar{W}_{\text{dec}} = R_2(\theta) \cdot \bar{W}_{\text{enc}}.$$

We used the `svm_wrapper.m` function of the toolbox to train a cross-validated (5 folds) soft-margin linear SVM (hyperparameter $C = 1$) to decode the stimulus $\hat{S}$ and the choice $\hat{C}$ from the neural population activity. We computed stimulus, choice, and intersection information as $I(S,\hat{S})$, $I(C,\hat{C})$, and $II\left(S,[\hat{S},\hat{C}],C\right)$ respectively, using the `MI.m` and `II.m` functions of the toolbox (with no bias correction).

We simulated two different scenarios, one with a small ($\theta = 20°$) and one with a large ($\theta = 70°$) angle between the encoding and the decoding vectors. We computed the angle between the decision boundaries of the SVMs trained to decode the stimulus $\bar{W}_{\hat{S}}$ and the one to decode the choice $\bar{W}_{\hat{C}}$ as

$$\hat{\theta} = \arccos\left(\frac{\bar{W}_{\hat{S}} \cdot \bar{W}'_{\hat{C}}}{||\bar{W}_{\hat{S}}|| \cdot ||\bar{W}_{\hat{C}}||}\right) \tag{7}$$

where the prime symbol ($'$) indicates the transpose operation, and $||\bar{W}||$ indicates the norm of vector $\bar{W}$. A total of 5 simulations were conducted for each scenario, and information and angle values were averaged across simulations.

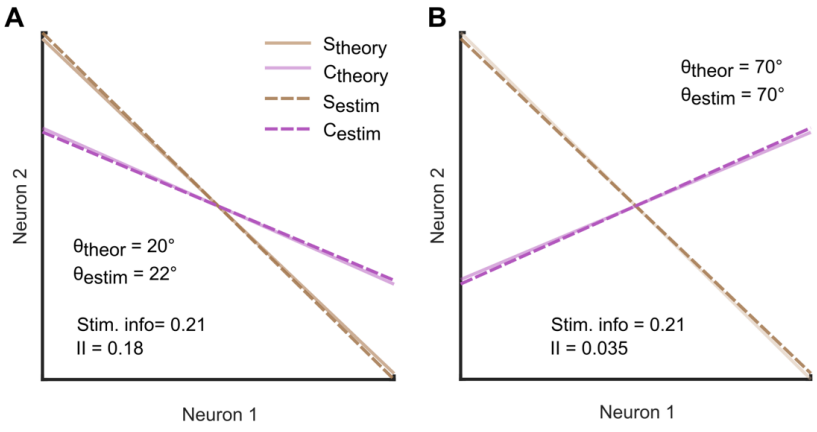

Figure A.4: Real and decoded stimulus and choice boundaries for a population of 2 neurons with a small (A) and a large (B) angle between the stimulus and choice boundaries. The X and Y axes represent the firing rate. Stimulus information Stim. Info $= MI(S;\hat{S})$ is the same for the two scenarios, but the intersection information $II(S;[\hat{S},\hat{C}];C)$ is much larger for the case with a small angle.

A.6.3 Simulation of Aggregate Activity in Interacting Nodes

This section presents a detailed description of the simulation and analysis of aggregate signal activity in a network of interacting nodes (Fig. 2.4 and Fig. A.5).

The network implemented content-specific encoding and transmission of information across four nodes, X_1, X_2, X_3, and X_4, each of them divided into two subnodes $X_{N,1}$ and $X_{N,2}$. A total of 10 simulations were conducted, each including 200 trials with a length of 30 ms. We simulated a stimulus that included two binary features S_1 and S_2 whose values were drawn independently in each trial. The value of $S_1(t)$ was set equal to the value of $s_1 \in [-1,1]$, while the value of $S_2(t)$ was set equal to the value of $s_2 \in [-1,1]$ within a defined stimulus-active time window. Outside this window, the value was set to zero.

Subnodes $X_{1,1}$ and $X_{4,1}$ received input regarding S_1, while subnode $X_{2,2}$ received input regarding S_2. The stimulus-active time window for $X_{1,1}$ and $X_{2,2}$ was defined as $[3,12]$ ms, while $X_{4,1}$ received the input with a delay Δt_2 of 12 ms (stimulus-active time window $[15,24]$ ms). In addition, all subpopulations received stimulus-feature unrelated activity at any time point as zero-mean Gaussian noise $\mathscr{E}_{N,M}(t) = N(0,\sigma_{\text{noise}})$ with standard deviation $\sigma_{\text{noise}} = 0.5$.

To simulate the transfer of feature-related and unrelated activity from one subnode to another, a delay Δt_1 of 5 ms was defined. With that time delay Δt_1, subnode $X_{1,1}$ transmitted its activity to $X_{2,1}$ and $X_{3,1}$, subnode $X_{2,2}$ to $X_{1,2}$, and $X_{3,2}$ transmitted its activity to $X_{4,2}$. The aggregated activity of each node was defined as the summed activity of the two subnodes. The activity of the four nodes at each time point is defined as:

$$X_1(t) = X_{1,1}(t) + X_{1,2}(t) = (\alpha \cdot S_1(t) + \mathscr{E}_{1,1}(t)) + X_{2,2}(t - \Delta t_1) + \mathscr{E}_{1,2}(t) \tag{8}$$

$$X_2(t) = X_{2,1}(t) + X_{2,2}(t) = X_{1,1}(t - \Delta t_1) + \mathscr{E}_{2,1}(t) + \alpha \cdot S_2(t) + \mathscr{E}_{2,2}(t) \tag{9}$$

$$X_3(t) = X_{3,1}(t) + X_{3,2}(t) = X_{1,1}(t - \Delta t_1) + \mathscr{E}_{3,1}(t) + \mathscr{E}_{3,2}(t) \tag{10}$$

$$X_4(t) = X_{4,1}(t) + X_{4,2}(t) = (\alpha \cdot S_1(t - \Delta t_2) + \mathscr{E}_{4,1}(t)) + X_{3,2}(t - \Delta t_1) + \mathscr{E}_{4,2}(t) \tag{11}$$

For all subsequent analyses, we binned the activity of each node into three equipopulated bins and corrected for the limiting sampling bias of information with the QE procedure implemented in the toolbox.

First, we computed the mutual information between each node and the stimulus features at all time points (see Fig. A.5 for the information time course), using the `MI.m` function of the toolbox.

Consistent with the implemented stimulus input and information transfer, X_1 exhibited information about S_1 from 4 to 11 ms, while X_2 and X_3 displayed a 5 ms delayed stimulus-feature informative window. The simulated activity of X_4 contained information regarding S_1 from 11 to 20 ms, consistent with the implemented input delay of 7 ms compared to

X_1. Information about S_2 was only present in X_1 ($[9,16]$ ms) and X_2 ($[4,11]$ ms), in line with the implemented input of S_2 and delayed transfer from X_2 to X_1.

To gain insight into the information transfer within the network, we computed the transfer entropy between all pairs of simulated nodes, using the `TE.m` function provided by the toolbox. Based on the established ground truth, the temporal parameters to compute transfer entropy were defined as $t = 12$ ms and $\Delta t = 5$ ms. Consistent with the implemented network interactions, we found significant transmission of information from X_1 to X_2 and X_3, from X_2 to X_1, and from X_3 to X_4 (see heatmap in Fig. A.5).

In the context of real data analysis, the optimal temporal parameters are typically not known. Therefore, we will demonstrate in the next step how one can assess these parameters by computing the time-delay maps of information transmission. To compute the content of information flow, we measured FIT with the `FIT.m` function implemented in the toolbox. To obtain the temporal profile of content-specific information transmission and to reconstruct the delay of information transmission, we first computed FIT at each time step of the simulation with all possible delays ($\Delta t = [0,29]$ ms) for $X_1 \rightarrow X_2$ related to S_1, $X_1 \rightarrow X_2$ related to S_2, and for $X_2 \rightarrow X_1$ related to S_2 (see Fig. A.5B). Consistent with the implemented ground truth, significant transfer of information was found with a delay of 5 ms from X_1 to X_2 related to S_1 and from X_2 to X_1 related to S_2.

By computing FIT for all pairs of nodes ($t = 12$ ms, $\Delta t = 5$ ms), we found significant information transfer related to S_1 from X_1 to X_2 and X_3, and significant information transfer related to S_2 from X_2 to X_1 (see Fig. 2.4D and Fig. A.5C).

To test for significance in the information theoretic quantities averaged across the $N = 10$ simulations, we used non-parametric permutation tests. For each simulation, we first conducted two different shuffling procedures $M = 20$ times and recomputed TE and FIT from the shuffled data ($t = 12$ ms, $\Delta t = 5$ ms). First, we conditionally shuffled the sender activity at a fixed value of the stimulus to preserve stimulus-induced covariations between the sender and the receiver and destroy single-trial correlations contributed by real communication. Second, we shuffled S for FIT to break any relationship between the stimulus and variables X and Y, and we shuffled X for the TE analysis to break any relationship between X and Y. We then took the pairwise maximum between the information values obtained from the two shuffling procedures to obtain a single, conservative null distribution [38]. Using the `create_nullDist_groupLevel.m` function of the toolbox, we generated $K = 500$ samples of the null distribution of the permuted average across simulations and estimated TE and FIT p-values empirically. To estimate the significance of FIT in the time-delay domain (Fig. A.5C), we implemented the same procedure outlined above, computing FIT at each time step of the simulation with all possible delays. We then used cluster permutation (with the pool option set to 0), setting both `clusterPercentilThreshold` and `significanceThreshold` to the 99th percentile to individuate significant FIT clusters.

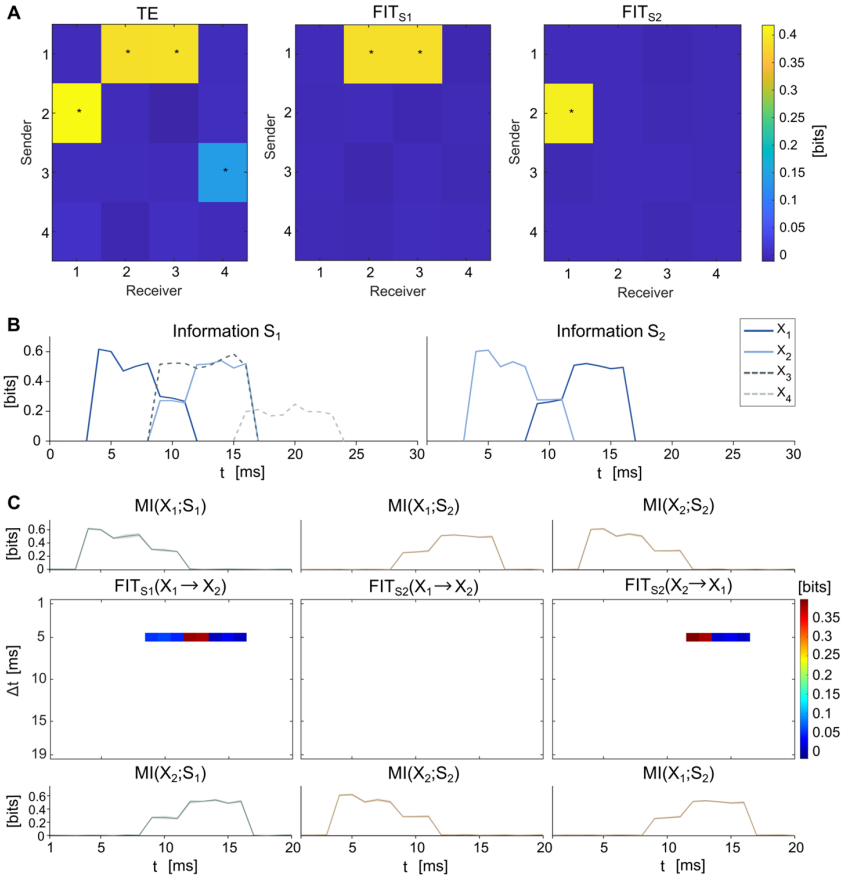

Figure A.5: TE and FIT heatmap, temporal profiles of stimulus-feature information and time delay stimulus-feature FIT maps for X_1 and X_2. A. TE heatmap for all nodes (left), FIT about S_1 for all nodes (middle), and FIT about S_2 for all nodes (right) for time point $t = 12$ ms with a delay of 5 ms. Significant values ($p < 0.01$) are marked with *. **B.** Mutual information time courses between the nodes X and stimulus features S_1 and S_2. **C.** Analysis on specific node pairs: X_1 to X_2 about S_1 (left), X_1 to X_2 about S_2 (middle), and X_2 to X_1 about S_1 (right). *Top.* Mutual information time course between the sender neural nodes and stimulus features. *Middle.* FIT values across post-stimulus time t and delay time Δt from sender to receiver nodes about the stimulus feature. Only the time region with significant ($p < 0.01$) stimulus information, according to a cluster permutation test with 500 null distribution samples, is plotted. *Bottom.* Mutual information between the receiver and stimulus feature. Plots show the mean across 10 simulations.

A.6.4 Simulation study of the limited sampling bias with various estimators and bias correction methods

This section presents a detailed description of the results of simulations of the activity of populations of neurons to illustrate and study the limited-sampling bias on the calculation of mutual information (Fig. A.3).

We first illustrated (Fig. A.3, Panel A, 'Sketch of bias for a zero-information process') the meaning and origin of the limited-sampling bias using a simple simulated example of two uninformative neurons. These neurons respond on each trial with a uniform distribution of spike counts ranging from 1 to 4, regardless of which stimulus values were presented. The colorplots in Fig. A.3, left side of Panel A show the joint and marginal probability distributions sampled from 50 simulated trials per stimulus in a single simulation. Despite the underlying distributions are completely uniform with respect to the stimulus or response value, the stimulus-specific probability distributions of the joint neural-pair or marginal single-neuron responses empirically sampled from a limited number of trials differ across stimuli because of the limited sampling. A naïve interpretation of this difference would make one conclude that since the response distribution changes for each stimulus, the neural response carries some information about the stimulus. When computing many (50000) repeated simulations of this uninformative process with 50 trials per stimulus each, the distribution of plugin information values computed with the direct-method (Fig. A.3, right side of panel A) did not have a mean equal to ground-truth information value of zero, but a different, positive mean. The bias is the difference between the ground-truth value of information and the one obtained on average across different random instantiations of a dataset with a given limited number of trials. In this case, the bias is 0.022 bits for the single neuron information and 0.12 bits for the joint information in the activity of the neuron pair. The reason why the limited-sampling bias is larger for the neuron pair than the single neuron can be appreciated by considering the example single instantiation of the empirical probability Fig. A.3, Panel A, left. The random fluctuations that generate spurious information are larger for the joint probability (the few trials are spread over a larger set of possible responses R = 16) than for the marginal probabilities (the few trials are spread over a smaller set of responses R = 4). As a result of these spurious differences between stimulus-specific response distributions, the values of information computed from individual simulations with the limited number of trials (right panel) would not be distributed around the true value of zero bits, but around a spurious non-zero value (the bias). Since the fluctuations are larger in the joint probability estimation than for the marginal distributions, the bias values would be bigger in this case in comparison to the information computed from the single neuron information distributions.

To further illustrate how the limited-sampling bias scales with the number of available trials, in Fig. A.3 Panel B ('Bias for a stimulus-informative Poisson process') we studied the performance of the estimation algorithms considering Poisson spike counts of populations of neurons ranging from N=1 to N=4, and carrying information about a binary stimulus. We used Poisson neurons because for this process we could evaluate numeri-

cally with high precision the ground-truth value of the information carried by the population. This ground truth value can be computed by inserting the analytical probabilities of the Poisson process into the Shannon formula and stopping the sum over responses at some value of spike counts much higher than the average mean spike count of the Poisson process (probabilities converge to zero for such very high spike counts and thus the sum over responses can be stopped at some point with negligible loss of numerical accuracy). To simulate what we could do in real data, we binned spike counts of each neuron into R =5 bins (0, 1, 2, 3, ¿3 spike counts). We then used the direct-method calculation without bias corrections (plugin) and with two bias-correction methods (QE, shuffle-subtracted). For comparison, we also computed the information estimated through a method popular for the analysis of real-valued data (KSG), which is also available in MINT. We also plotted the ground-truth value of information computed from the analytical form of the probability distributions of the Poisson spike counts after binning the data. The QE bias correction was done with 10 repetitions (xtrp=10). The shuffle-subtraction was done with 30 shuffles (shuff=30).

As documented in previous studies [36], the limited-sampling bias of the bias-uncorrected, plugin, estimator is very high for low numbers of trials and then smoothly decreases with the number of trials. The bias also grows very rapidly with the population size. Application of QE or shuffle-subtraction bias correction to the plugin estimators provides an effective elimination of the limited-sampling bias for number of trials that are realistic with a neuroscience experiment (100) for N=1,2 but not for N¿2. The KSG method, not being designed for discrete spike count processes, does not converge to the ground-truth information value even for large trial numbers, with a data-processing bias that grows with N.

The above simulations indicate a distinction between two types of bias. The first is the (usually upward) limited-sampling bias, which is the difference between the value of information obtained with the considered estimation method (e.g. response binning, KSG, etc.) when considering a finite number of trials vs when considering an infinite number of trials. The second is the data-processing bias, which reflects the possible inability of the estimation method to process or represent the probability distributions of the data correctly. The data-processing bias can be computed as the difference between the ground truth value of information and the one obtained with the considered estimation method if the data were infinite. For example, when discretizing neural responses into a more limited number of possible discrete responses than the ones that could actually happen (for example, binarizing as all/none the output of a Poisson count), the data estimation bias would be negative.

When using the direct-method discrete estimators with binned responses, we found that the direct method had a negligible data-processing bias. The direct-method discrete information estimator without bias correction is however upward biased for large numbers of trials. The bias corrections algorithms QE and shuffle-subtract worked well and made the estimate convergence close to the ground truth value even for a relatively small number of trials. In contrast, the KSG has a relatively small limited sampling bias but at a cost of a large downward data-processing bias, which reflects the fact that its assumption does

not suit discrete data as well as real valued data. We thus would not use the KSG for computing information from neural spike counts.

We next investigated (Fig. A.3, panels C and D) the interplay between data dimensionality reduction and limited sampling bias. We simulated the two scenarios of correlated neural population activity that were used in Figure 2.2A,B and that are detailed in Section SM6.1. We then implemented different dimensionality reduction algorithms to reduce the 20-dimensional neural data to 1 dimension. The activity of the 20 neurons was reduced to one dimension using either `svm_wrapper.m` for linear or RBF SVM or `pca_wrapper.m` for PCA. All dimensionality reduction methods were crossvalidated using 2 folds.

We first considered a case in which correlations between activity of neurons are present but are not much informative and the information about the stimulus is by and large specified by differences in average spike counts of individual neurons across stimuli. This case is equal to the one presented in Fig. 2.2B and the results of the simulations are reported in Fig. A.3, Panel C ('Dimensionality reduction with information in single neurons spike counts'). The information estimates from the reduced 1-dimensional representation show little dependence on the sample size and are almost identical between corrected and uncorrected (plugin) estimates, suggesting that the dimensionality reduction has little bias and that calculation of information from a population of 20 neurons can be performed robustly with limited data sizes when using dimensionality reduction. All dimensionality reduction methods perform similarly, because it is very easy for all methods to find the dimension in neural activity space with maximal information (which is simply the vector of the averaged differences in spike counts between the two stimuli) in a case in which the information is encoded in major differences in spike counts across stimuli.

We finally considered a case in which spike counts of individual cells did not carry information (average spike counts were constant across stimuli) but all information was encoded in the stimulus variations of the strength of correlations between activity of different neurons. This case is equal to the one presented in Fig. 2.2A and the results of the simulations are reported in Fig. A.3, Panel D ('Dimensionality reduction with information in correlations'). In this case, and also shown in Fig. 2.2A, the information can be recovered only with a complex non-linear decoder (RBF SVM) for any considered number of trials. Limited sampling bias appears small but the increase of RBF SVM suggests that more complex supervised decoders benefit from more training data to reduce the data processing bias.

In conclusion, for tens of neurons the limited sampling bias can be well controlled for using dimensionality reduction methods. Most algorithms extract similar amounts of information even with little training data when information is encoded in very easily detectable features of neural activity (e.g. clear spike count separations) while more sophisticated algorithms (e.g. non-linear supervised decoders) and more training data are needed when information is hidden only in more subtle features of neural activity such as the correlations between their activity.

A.7 Supplemental Analysis of Real Neural Data

A.7.1 Analysis of EEG Data

We analyzed a publicly available EEG dataset [131] (available at `datadryad.org/stash/dataset/doi:10.5061/dryad.8m2g3`). Full details are reported in the original publication. Here we summarize them briefly. The EEG data were recorded while participants ($n = 16$) performed a face detection task. Participants were presented with an image hidden behind a bubble mask that was randomly generated in each trial. The presented image was an image of a face in half of the trials and a random texture in the other half of the trials. Participants were instructed to report whether a face was present or not. In our analyses, we only considered trials where the face was correctly detected by the participants (approximately 1000 trials per subject). Following the recommendations of the original publications analyzing these data [131, 132], we excluded one participant from the analysis. All analyses in our paper are based on the $n = 15$ selected participants. EEGs were recorded by fitting participants with a Biosemi head cap comprising 128 EEG electrodes. EEG data were re-referenced offline to an average reference, band-pass filtered between 1 Hz and 30 Hz using a fourth order Butterworth filter, down-sampled to 500 Hz sampling rate, and baseline corrected using the average activity between 300 ms pre-stimulus and stimulus presentation. ICA was performed to reduce blink and eye movement artifacts, as implemented in the infomax algorithm from EEGLAB [184].

For the analyses of TE and FIT, we selected the EEG electrodes in the left and the right Occipito-Temporal regions that had the highest Mutual Information about the visibility of the contra-lateral eye, exactly as done in previous papers [38, 132]. We computed the first derivatives of the EEG signal for both Occipito-Temporal sensors and used both their absolute values and first derivatives to compute the information quantities, for consistency with analyses performed in previous studies [38, 132]. As stimulus feature for the computation of Mutual Information and FIT, we used the visibility of an eye (defined as the fraction of pixels within the eye region that were not hidden by the bubble mask). Both neural and stimulus features were discretized using 2 equi-populated bins. We computed the information quantities for all combinations of directionality of flow across hemispheres (left to right, right to left) and eye identity (left or right eye). As done in previous papers [38], to compute a single TE and FIT value for each participant we selected a rectangular region in the time-delay domain centered around the contra-lateral FIT peaks (time ranging from 140 ms to 240 ms peri-stimulus presentation, delay ranging from 20 ms to 90 ms; same for both eyes, as they were significant in very similar time-delay regions). We computed the average over delays and then picked the maximum over time within this region. We used the same procedure described in Section A.6.3 to compute the significance of the across-participants averaged FIT and TE (Fig. 2.4G-H), generating 500 null samples from 10 shuffles within each participant.

Files that reproduce the analysis of these data are found in `github.com/panzerilab/MINT_figures`, subfolders 'Figure4'.

A.7.2 Analysis of CA1 Data

We reanalyzed a previously published dataset [69] in which the activity of several tens to a few hundreds of neurons was recorded simultaneously using in-vivo two-photon calcium imaging from CA1 neurons in head-fixed transgenic mice during virtual reality navigation of a linear track. This dataset is provided as Supplemental Information file `'CA1_data.mat'`.

We analyzed neurons recorded from $n_{\text{FOV}} = 11$ Fields of View (FOV) from $n_A = 7$ animals. For consistency with the previous study reporting the original data [69], the spatial position of the linear track was computed by binning the space along the track into $S = 12$ equi-populated bins. Also, the neural activity r_i of each neuron was quantified by binning the calcium traces into $R = 2$ equi-populated bins (only raw calcium traces and not deconvolved signals were available from [69]). For the PID analysis, we used all the $n = 870$ individual neurons present in the dataset, leading to $n_{\text{pairs}} = 36158$ pairs of simultaneously recorded neurons used for the pairwise direct information analyses and $n_{\text{sessions}} = 11$ sessions for the population vector analyses. The neural responses dimensionality was reduced using linear and nonlinear (RBF) SVM to predict the position categories using 5-fold cross-validation and hyperparameter optimization (2-fold

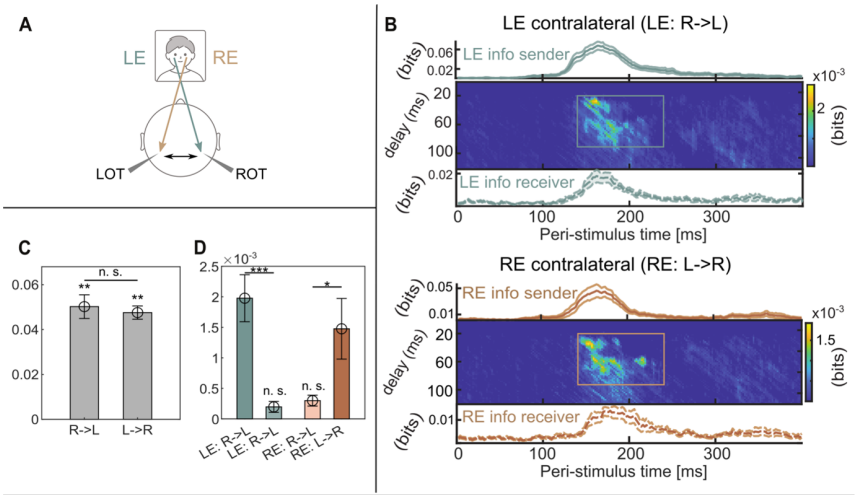

Figure A.6: **A.** Schematic of the putative information flow. LOT (ROT) denote Left (Right) occipito-temporal regions. LE (RE) denotes the Left (Right) Eye visibility feature. **B.** FIT values computed across post-stimulus time t and delay time from sender region to receiver region about the stimulus feature. The region of the time-delay maps used to calculate the final FIT values for Fig. 2.4H are delimited in both heatmaps. *Top.* Mutual Information (lines) carried by the EEG in each region, and FIT (image plot) about LE contra-lateral transfer. *Bottom.* Same as Top for RE. **C.** Transfer entropy between ROT and LOT in both possible directions. **D.** FIT values computed for each stimulus (LE or RE) and direction. In this plot, ROT and LOT are written as R and L, respectively. Dots and image plots represent averages, and error bars represent SEM across participants ($N = 15$). Human face sketch is modified from svgrepo.com/svg/493087/men-in-their-20s-and-30s-face, and head sketch is modified from `doi.org/10.5281/zenodo.3926093`. All resources are under license CC BY 4.0 (`creativecommons.org/licenses/by/4.0`).

cross-validation, Bayesian optimization across a logarithmic range between 1×10^{-3} and 1×10^{3} for C and γ with a maximum number of iterations equal to 30), and later the predicted labels were used jointly to replace the full neural response.

Files that reproduce the analysis of these data, as well as the neural data themselves, are found `github.com/panzerilab/MINT_figures`, subfolder 'Figure2'.

A.7.3 Analysis of A1 Data

We reanalyzed a previously published dataset [129] in which the activity of several tens to a few hundreds of neurons was recorded simultaneously using in vivo two-photon calcium imaging from A1 L2/3 neurons in head-fixed transgenic mice during a pure-tone discrimination task. Data are publicly available at `doi.org/10.13016/m2yt-mfxk`.

The experimental task was structured as follows. After a pre-stimulus interval of 1 s, head-fixed mice were exposed to either a low-frequency (7 or 9.9 kHz) or a high-frequency (14 or 19.8 kHz) tone for a period of 1 s. Mice were trained to report their perception of the sound stimulus by their behavioral choice, which consisted of licking a waterspout in the post-stimulus interval (0.5–3 s from stimulus onset) after hearing a low-frequency tone (target tones) and holding still after hearing high-frequency tones (non-target tones). Two-photon calcium imaging was used to acquire the calcium fluorescence signals from individual A1 L2/3 neurons during the task with an imaging frame rate of 30 Hz. We pre-processed these data as follows to match the pre-processing used by the authors in the original publication [129]. We smoothed the raw calcium fluorescence traces using a zero-phase (MATLAB `filtfilt.m` function) order-2 low-pass Butterworth filter (MATLAB `butter.m` function) with low-pass cutoff frequency of 30 Hz (equal to the original sampling frequency of the calcium imaging data). As in the original publication [129], the resulting traces were deconvolved with a first-order autoregressive model.

We analyzed neurons recorded from $n = 12$ Fields of View (FOV) from $n = 12$ animals. For consistency with the previously published work [129], we only considered individual neurons that carried significant intersection information as described in [129]. This led to selecting $n = 375$ individual neurons (out of the 2792 recorded neurons), leading to $n = 6209$ pairs of simultaneously recorded neurons used for the pairwise direct information analysis and $n = 12$ sessions for the population analyses. For the information analysis of these data, we identified for each neuron the imaging time frame within the trial of maximal intersection information, exactly as in the original publication [129]. We then considered for each neuron a time frame of $n = 10$ imaging frames (corresponding to a window of 333 ms) around the peak intersection information time frame (we call this the peak time window for the neuron). Then we discretized activity for each neuron into $R = 3$ bins according to whether it detected 0, 1, or > 1 spikes in the peak time window (this was done by setting the binning method field in the input options `opts` of `MI.m` to `'userEdges'`, which allows binning the data with user-defined bin edges). The stimulus set used for the stimulus encoding analysis was binary, dividing the presented sound tones into the low- and high-frequency categories. The choice set used for the intersection information analysis was also binary (lick vs no lick). For Figure 2.3, we used the

`svm_wrapper.m` function of the toolbox to train a cross-validated (2 folds) soft-margin linear SVM (hyperparameter $C = 1$) to decode the stimulus $\hat{S}$ and the choice $\hat{C}$ from the neural population activity. We computed stimulus and intersection information as Mutual Information $I\left(S,\hat{S}\right)$ between presented and decoded stimulus and II as the Intersection Information $II\left(S,\left[\hat{S},\hat{C}\right],C\right)$ between the presented stimuli, the mouse choices, and the stimulus and choice decoded from neural activity using the `MI.m` and `II.m` functions of the toolbox. The bias was corrected by applying the shuffle subtraction procedure, setting the `shuff` field of the `opts` structure to 30. Figure A.2 illustrates the pipeline of dimensionality reduction and information measurement used to generate Fig. 2.3. The fitting was performed using a second-order polynomial on the logarithm of the population size.

Files that reproduce the analysis of these data are found in `github.com/panzerilab/MINT_figures`, subfolders 'Figure2' and 'Figure3'.

Appendix B

Supplementary information of Chapter 3

The analyses presented here correspond to those presented as supplementary information in the first submitted version of the paper [138] prior to being revised.

B.1 Definitions of Shannon Information quantities and further relationships with PID components

As described in the main text, once a definition of any of the four PID components is provided, the other three can be obtained as linear combinations between Shannon Information quantities and the defined PID component. To facilitate implementation by users, in the following we report the equations to compute RI and UI as the linear combination of Shannon Information quantities and SI (for which we provided a definition in the main text Eq. (3.7)).

By subtracting Eq. (3.4) from Eq. (3.3) we obtain:

$$I(S;R_1,R_2) - I(S;R_1) = I(S;R_2|R_1) = UI(S:R_2 \setminus R_1) + SI(S:R_1;R_2) \tag{B.1}$$

where $I(S:R_2|R_1)$ is the conditional mutual information about S provided by R_2 given R_1, defined as the joint information carried by R_1 and R_2 minus the information carried individually by R_1 [143]. Analogously, by subtracting Eq. (3.5) from Eq. (3.3) we obtain:

$$I(S;R_1,R_2) - I(S;R_2) = I(S;R_1|R_2) = UI(S:R_1 \setminus R_2) + SI(S:R_1;R_2) \tag{B.2}$$

Finally, by subtracting Eq. (3.4, 3.5) from Eq. (3.3) we obtain:

$$I(S;R_1,R_2) - I(S;R_1) - I(S;R_2) = SI(S:R_1;R_2) - RI(S:R_1;R_2) \tag{B.3}$$

where the quantity on the LHS of Eq. (B.3) is known as the co-information $CoI(S;R_1;R_2)$ of S, R_1 and R_2:

$$CoI(S;R_1;R_2) = I(S;R_1,R_2) - I(S;R_1) - I(S;R_2) \tag{B.4}$$

Linear relationships between the four PID components and Shannon information quantities are depicted in Fig. B.1. While Eq. (B.1-B.3) do not impose any independent constraint on the four PID components additionally to the ones in Eq. (3.3-3.5), they express explicitly RI and UI as combinations of Shannon information quantities and SI, as follows:

$$\begin{aligned} RI(S:R_1;R_2) &= SI(S:R_1;R_2) - I(S;R_1,R_2) + I(S;R_1) + I(S;R_2) \\ UI(S:R_1 \setminus R_2) &= I(S;R_1|R_2) - SI(S:R_1;R_2) \\ UI(S:R_2 \setminus R_1) &= I(S;R_2|R_1) - SI(S:R_1;R_2) \end{aligned} \tag{B.5}$$

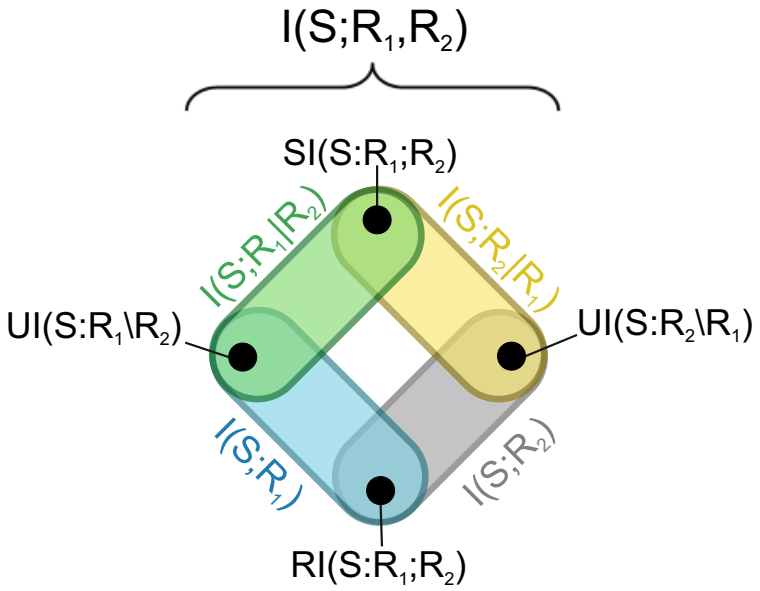

Figure B.1: PID decomposes the joint information that the sources R_1 and R_2 carry about S into four components: the redundancy $RI(S:R_1;R_2)$, the unique information $UI(S:R_1 \setminus R_2)$ and $UI(S:R_2 \setminus R_1)$, and the synergy $SI(S:R_1;R_2)$. The colored outlines represent the four linear Eq. (3.4-3.5) and (B.1-B.2) that relate the four PID components to Shannon information quantities.

Expressions similar to the ones of Eq. (B.5) can be easily obtained to write any three of the PID components as an explicit function of the fourth one. For example, we could have written UI_{R_1}, UI_{R_2}, and SI as a function of RI.

B.2 Definitions and properties of different PIDs

In this section we first provide a mathematical definition of the three measures of PID components used in this paper. Then, we detail important properties of PID of two source variables that are relevant to this study, and discuss which properties these three measures do or do not satisfy. For extensive reviews on available PID measures and their properties, we refer [144, 145].

B.2.1 Different measures of PID components and their properties

B.2.1.1 I_{min}

In the work introducing PID, Williams and Beer proposed a measure of redundant information about S, called I_{min}, that for two source variables R_1 and R_2 is defined as follows:

$$RI(S:R_1;R_2) = \sum_{s\in S} p(s) \min_{R_i\in\{R_1,R_2\}} I(S=s;R_i) \tag{B.6}$$

where $I(S=s;R_i)$ is the specific information that source R_i carries about a specific value of the target variable $s \in S$, and is defined as:

$$I(S=s;R_i) = \sum_{r_i\in R_i} p(r_i|s)\Big[\log \frac{p(s|r_i)}{p(s)}\Big] \tag{B.7}$$

Intuitively, the redundant information computed as in Eq. (B.6) quantifies redundancy as the similarity between R_1 and R_2 in discriminating individual values of the target S.

B.2.1.2 I_{MMI}

Minimum mutual information I_{MMI} was introduced in [110]. This measure is important since both I_{min} and BROJA measures reduce to I_{MMI} for Gaussian systems with an univariate target. This is probably the simplest PID measure, defining the redundant information that R_1 and R_2 carry about S as follows:

$$RI(S:R_1;R_2) = min\{I(S;R_1), I(S;R_2)\} \tag{B.8}$$

I_{MMI} quantifies redundant information by directly comparing the amount of information encoded individually by each source, and results in at least one component of unique information always being null.

B.2.1.3 I_{BROJA}

The BROJA measure [31] is defined as an optimization problem, minimizing Shannon information quantities that depend on the probability distributions $p(S,R_1,R_2)$ defined in the probability space Δ_P of distributions $q(S,R_1,R_2)$ having pairwise marginals between each source and the target equal to the original ones, i.e. $q(S,R_1) = p(S,R_1)$ and $q(S,R_2) = p(S,R_2)$. The rationale of this approach is that the synergy can be conceptualized as the information about the target in the joint space of the two sources that cannot be possibly recovered by observing one source at a time (thus from the marginal probabilities), thus it can be defined operationally as the difference between the original joint information and the minimal joint information about the target that can be found in the space Δ_P of distributions that preserve the marginals (which is the union information, Eq. 3.6):

$$SI(S:R_1;R_2) = I(S;R_1,R_2) - \min_{q\in\Delta_P} I_q(S;R_1,R_2) \tag{B.9}$$

Redundancy and unique information components can then be computed from the synergy and the Shannon information values using the linear constraints as explained above (see Eq. (B.5)). Among several advantages of the BROJA PID, we note that this definition insures that the redundancy and the unique information values will have the same value for all the distributions in the space Δ_P of distributions that preserve the marginals (the so called pairwise marginals property of PIDs).

B.2.2 Key properties of PID components for two source variables

In their paper introducing PID for the first time, Williams and Beer [30] proposed a set of properties that a measure of *RI* should satisfy. In the following years, other authors introduced new properties which they thought to be important for measures of PID components [111, 141, 145].

Three particularly important properties of PID measures with two source variables are: non-negativity, symmetry and additivity [45].

- Non-negativity: each PID component should be non-negative, i.e. $RI \geq 0$, $UI_{R_1} \geq 0$, $UI_{R_2} \geq 0$ and $SI \geq 0$. This property is important since it allows interpreting PID components as fractions of the joint mutual information about the target S encoded by the source variables (i.e., the average reduction of the entropy of S obtained when measuring the two sources simultaneously). All three BROJA, I_{min} and I_{MMI} measures satisfy non-negativity [30, 31, 110].

- Symmetry: *RI* and *SI* are symmetric under reordering of source variables, i.e. $RI(S:R_1;R_2) = RI(S:R_2;R_1)$ and $SI(S:R_1;R_2) = SI(S:R_2;R_1)$. This property is important since also Shannon information quantities which can be decomposed in terms of *RI* and *SI* (including the joint information and the co-information) are symmetric under reordering of R_1 and R_2. All three BROJA, I_{min} and I_{MMI} measures satisfy symmetry [30, 31, 110].

- Additivity: given two independent systems of random variables (S_1, X_1, Y_1) and (S_2, X_2, Y_2), then

$$\begin{aligned} RI((S_1,S_2):(X_1,X_2);(Y_1,Y_2)) &= RI(S_1:X_1;Y_1) + RI(S_2:X_2;Y_2) \\ SI((S_1,S_2):(X_1,X_2);(Y_1,Y_2)) &= SI(S_1:X_1;Y_1) + SI(S_2:X_2;Y_2) \\ UI((S_1,S_2):(X_1,X_2)\setminus(Y_1,Y_2)) &= UI(S_1:X_1\setminus Y_1) + UI(S_2:X_2\setminus Y_2) \\ UI((S_1,S_2):(Y_1,Y_2)\setminus(X_1,X_2)) &= UI(S_1:Y_1\setminus X_1) + UI(S_2:Y_2\setminus X_2) \end{aligned} \tag{B.10}$$

  This property is important since it guarantees that PID components of two independent systems can be computed independently and then summed together. BROJA satisfies additivity, while I_{min} and I_{MMI} do not [111].

An additional property satisfied by all three measures is the *pairwise marginals* property [145], i.e. *RI*, UI_{R_1} and UI_{R_2} only depend on pairwise marginal distributions between

the sources and the target $p(S,R_1)$ and $p(S,R_2)$. This implies that all dependencies between the target S and the sources (R_1,R_2) that go beyond the dependency between the individual sources and the target are quantified as synergy. Even though the pairwise marginals is a fundamental property of the three PID measures, it has been questioned whether PID measures should always satisfy it [145, 185].

B.3 Illustration of bias and details of implementation of the bias correction procedures

B.3.1 Illustration of the origin of the bias

In this section we illustrate the effect of limited sampling on information calculation (both joint and single neuron information). We simulated two completely stimulus uninformative neurons, responding on each trial with a uniform marginal distribution of spike counts ranging from 1 to 4, regardless of which of 2 stimuli was presented. The neurons are also uncorrelated, so that their joint distribution is uniform across all possible 16 joint responses, and equal between the two stimuli.

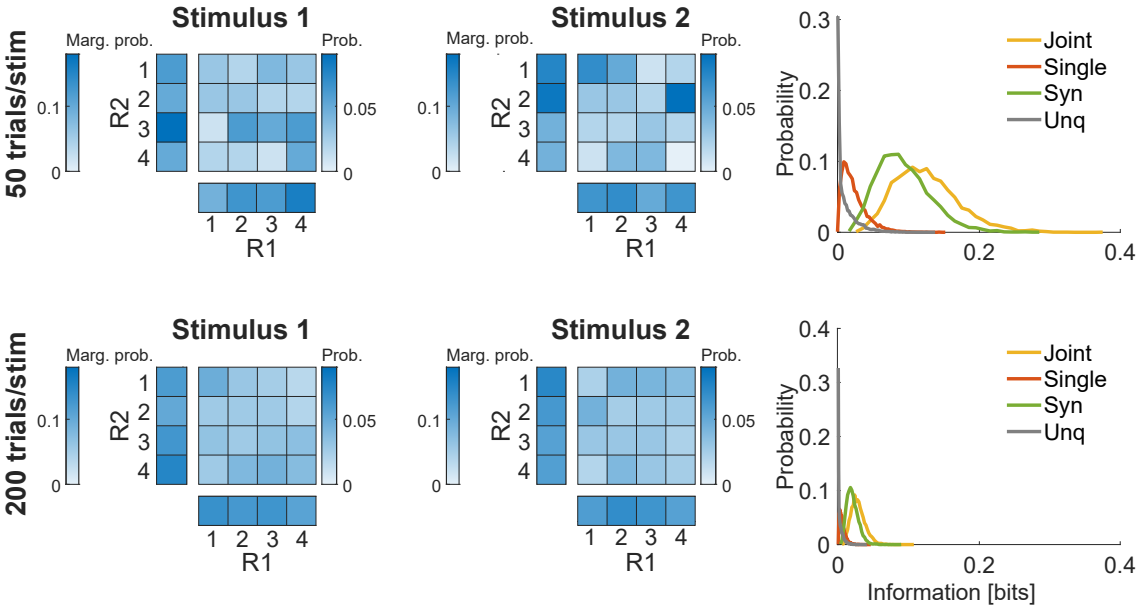

Figure B.2: Schematic of the limited sampling bias problem. Two uninformative neurons, responding on each trial with a uniform distribution of spike counts ranging from 1 to 4, regardless of which of 2 stimuli was presented. Examples of empirical response probability heatmaps sampled from 50 and 200 trials per stimulus (top and bottom rows, respectively) are shown in the left and middle columns (responses to stimuli 1 and 2, respectively). Each of the heatmaps has two more heatmaps indicating the marginal probability values. Right: distribution (over 5,000 simulations) of the plugin information values obtained with 50 (top) and 200 (bottom) trials per stimulus respectively.

Examples of marginal and joint stimulus-conditional neural response empirical response probability histograms sampled from a finite number of trials (50 trials per stimulus in the top row and 200 trials per stimulus in the bottom row respectively) are shown in the left and middle columns (responses to stimuli 1 and 2, respectively). Despite all the true response distributions being uniform across response bins and stimuli, a single instantiation of the probabilities from simulated data will have differences across stimuli that are not real but just due to random fluctuations. With smaller numbers of trials, we can appreciate that the random fluctuations that generate spurious information are larger for the joint probability (the few trials are spread over a larger set of possible responses $R = 16$) than for the marginal probabilities (the few trials are spread over a smaller set of responses $R = 4$). As a result of these spurious differences between stimulus-specific response distributions, the values of information computed from individual simulations with the limited number of trials (right panel) would not be distributed around the true value of zero bits, but around a spurious non-zero value (the bias). The bias is smaller for the single-neuron information values obtained from the marginal probabilities (unique information and single neuron information) than for the information obtained from the joint distribution. With larger number of trials (bottom row), these random fluctuations get smaller (the single instantiation in the bottom row looks more the same across stimuli), but much more so for the marginal than for the joint distribution, and thus the bias reduces more for the quantities obtained from the joint distribution (unique, single neuron) than for the joint information and the synergy.

B.3.2 Quadratic Extrapolation (QE)

The Quadratic Extrapolation (QE) procedure [41] assumes we are in the asymptotic sampling regime, so that the bias of the entropies and information can be accurately approximated as second order expansions in $1/N$ [42]. That is, we assume that

$$I_{plugin}(S;R_1,R_2) = I_{unbiased}(S;R_1,R_2) + \frac{a}{N} + \frac{b}{N^2} \tag{B.11}$$

where a and b are free parameters estimated from the data. This is done by re-computing the information from fractions ($N/2$ and $N/4$) of the data available and then fitting (using a least-square-error procedure) the plugin information values obtained with fractions of data to the preceding quadratic function of $1/N$. This provides the best-fit estimates of the parameters a and b and consequently the estimate of $I_{unbiased}(S;R_1,R_2)$. In the numerical implementation of QE, we allowed for a user-selected number K of data partitions into halves and quarters. We performed a separate extrapolation for each partition and then we averaged the resulting estimates of $I_{unbiased}(S;R_1,R_2)$ to obtain our final estimate of $I_{unbiased}(S;R_1,R_2)$. The procedure is explained in the above for the joint information but we applied it to each and every PID term.

B.3.3 Shuffling-subtraction bias correction

To estimate the level of the bias, we computed PID after randomly permuting the stimulus values across trials. In this way, all information about the stimulus is lost and all PID terms should be zero, but for finite sampling effect. Thus, the resulting PID terms can be taken as an estimation of their bias, which we subtracted from their plugin estimates. In our implementation, we allowed the user to perform a number V of random permutations and then we took as a measure of the bias (to be subtracted from the plugin estimate to obtain the unbiased estimates) the average of the PID term computed over all random realizations of the permutation.

In simulations in the main text and supplement, we used $V = 20$ and $K = 20$. In the analyses of real data, we used $V = 1$ and $K = 1$.

B.4 Details of the binning procedures used to compute the PID

For all simulated data, we estimated the PID components based on the following binning approach. First, in each simulation, we generated a large number ($n = 2048$) of trials per stimulus for each set of model parameters. Then, we discretized the probability distribution of spike counts for each neuron in the chosen number of bins. We computed the bin edges that made the partition of trials as equally populated as possible, following the procedure of Ref. [85]. These bin edges were then used for all simulations computing PID as a function of the number of trials per stimulus. We chose to fix the bin edges for each simulation to ensure that differences across trial numbers in the estimated values of the PID components were due only to trial numerosity and not to other reasons. This was important to study the properties of the information estimates as a function of the number of available trials.

For real data, the binning procedure is specified in the SM section describing their information analysis (see SM Section B.10).

A table with all the number of bins used to discretize single neuron activity in each figure is reported in the Table below.

Number of bins	2	3	4	8
Figures	Fig.3.4C Fig.B.4	Fig.3.4A Fig.3.4B	Fig.3.1 Fig.3.2 Fig.3.3 Fig.B.2 Fig.B.3 Fig.B.6 Fig.B.7 Fig.B.8 Fig.B.9 Fig.B.10 Fig.B.11 Fig.B.12 Fig.B.13	Fig.B.5

Table B.1: Number of bins used to discretize neural activity for information-theoretic analyses of simulated and real data, for both the main text and SM figure

B.5 Details of simulations used to test the bias properties and further results of these simulations

B.5.1 Details of simulations

To test the bias of the PID algorithms, we developed a simulation of the spiking activity of two neurons responding to a set of stimuli, as follows. The spike count r_1 and r_2 of each of the two simulated neurons for each of the four stimulus values was the sum of two Poisson processes, one Poisson process that was independently drawn for each neuron expressing the variability of responses that is "private" to each neuron, and a second Poisson process that is shared between the two simulated neurons and which gives rise to noise correlations.

The equations for the mean rates (that is, mean spike counts) of each Poisson process were as follows:

$$\begin{aligned} rate_{shared} &= \gamma \\ rate_{individual-1} &= B + \alpha\beta s_1 + \alpha(1-\beta)s_2 \\ rate_{individual-2} &= B + \alpha(1-\beta)s_1 + \alpha\beta s_2 \end{aligned} \tag{B.12}$$

In the above mean rate equation used for our simulations of the spike counts of pairs of neurons in response to the stimulus, we had the following free parameters, which were varied across simulations. We had (i) a parameter α regulating the strength of stimulus tuning of the spike counts of the individual neurons or in other words the separation in response strength between least and most effective stimuli (higher values of α leading to higher values of individual neuron information and thus to higher values of joint information); (ii) a baseline parameter B expressing the overall baseline level of activity common to all stimuli (higher values of B in general decreased information of individual neurons and thus the joint information because they increased the standard deviation of responses at fixed stimulus and thus made the rate separation α between most and least effective stimuli smaller in standard deviation units); (iii) a parameter β which increased the amount of redundancy because it regulated the dissimilarity of tuning of the spike

count of each individual neurons to the two stimulus features (β close to zero means independent tuning, i.e. each neuron encodes a different feature; higher values of β close to $\beta = 0.5$ indicate that both neurons encode the features similarly and redundantly, i.e. the two neurons encode the same linear combination of features); (iv) and a parameter γ regulating the strength of the shared process and thus the strength of correlated firing (increasing γ increased synergy because it made the two neurons more strongly correlated, and the fact that the proportion of shared spikes differed between stimuli that were more or less effective for each individual neuron made the overall correlation strength stimulus dependent thereby increasing the amount of synergistic information about the stimulus that can be gained only measuring the joint responses of the two neurons [33]).

For the three scenario presented in the top, middle and bottom rows of Fig. (3.1, 3.2, B.3, B.4, B.5, B.6, B.7, B.8, B.10, B.11) the simulations parameters were as follows. For all scenarios, α was set to 7 for the lower information case and to 10 for the higher information case, and B was set to 5. For the uncorrelated scenario (top row), we set $\gamma = 0$ and $\beta = 0$. For the high redundancy scenario (middle rows), we set $\gamma = 2$ and $\beta = 0.4$. For the low redundancy scenario (bottom rows), we set $\gamma = 20$ and $\beta = 0.1$. For the figures with the parameter sweeps (Fig 3.3,B.9), we used the same B, α, β values as above for the three scenarios but we varied α from 1 to 15 in steps of 1.

B.5.2 Further results of simulations

Here we present and collect all additional analyses of simulated data reporting the bias properties of the PID.

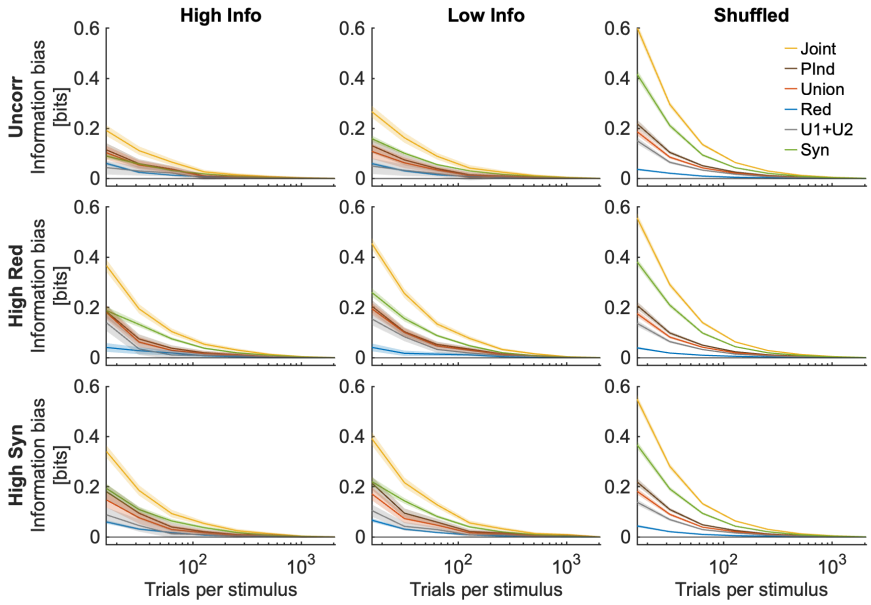

Figure B.3: Joint information and PID quantities as a function of the number of simulated trials used to compute them. Here we used the plugin method. In each panel we plot (rather than the information value itself) the information component bias, computed as the information component estimated with the considered number of simulated trials minus the asymptotic information component estimated with the largest number of simulated trials, that is 2048 trials per stimulus). Top, central and bottom row plot the simulated scenarios with no interaction, high redundancy and high synergy, respectively (see SM Section B.5). Left, center and right columns represent simulations with higher information ($\alpha = 10$), lower information ($\alpha = 7$) and with shuffled low-information data. "Syn": synergy. "Red": redundancy. "U1+U2": sum of the two unique information of each neuron. We used $R = 4$ discretization bins for each neuron (Table B.1). Each panel plots mean ± 2 SEM over $n = 96$ simulations.

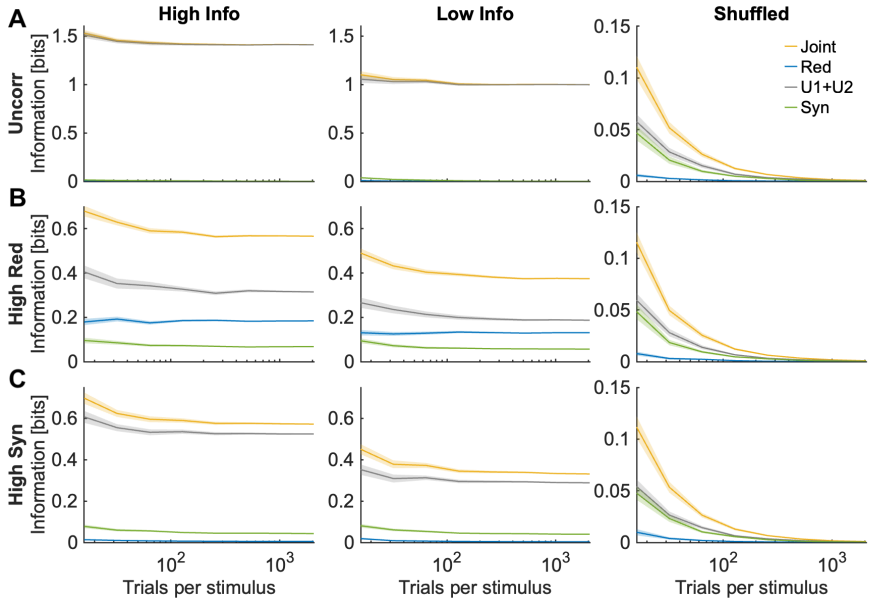

Figure B.4: Joint information and PID quantities using BROJA as the redundancy measure as a function of the number of simulated trials used to compute them. Plotting conventions are exactly as in Fig. 3.1. We used $R = 2$ discretization bins per each neuron (see Table B.1). Results in each panel are plotted as mean ± 2 SEM over $n = 96$ simulations.

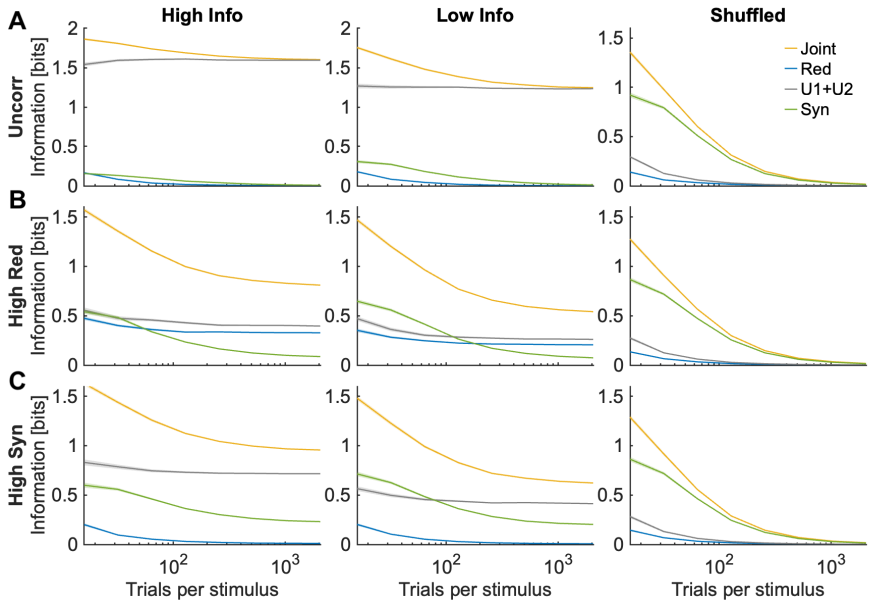

Figure B.5: Joint information and PID quantities using BROJA as the redundancy measure as a function of the number of simulated trials used to compute them. Plotting conventions are exactly as in Fig. 3.1. We used $R = 8$ discretization bins per each neuron (see Table B.1). Results in each panel are plotted as mean ± 2 SEM over $n = 96$ simulations.

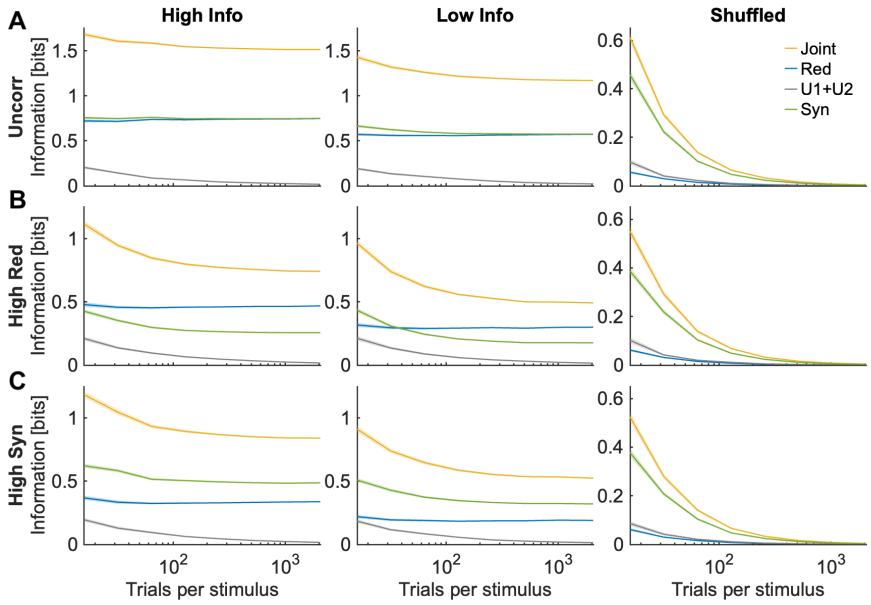

Figure B.6: Joint information and PID quantities using I_{min} as the redundancy measure as a function of the number of simulated trials used to compute them. Plotting conventions are exactly as in Fig. 3.1. We used $R = 4$ discretization bins per each neuron (see Table B.1). Results in each panel are plotted as mean ± 2 SEM over $n = 96$ simulations.

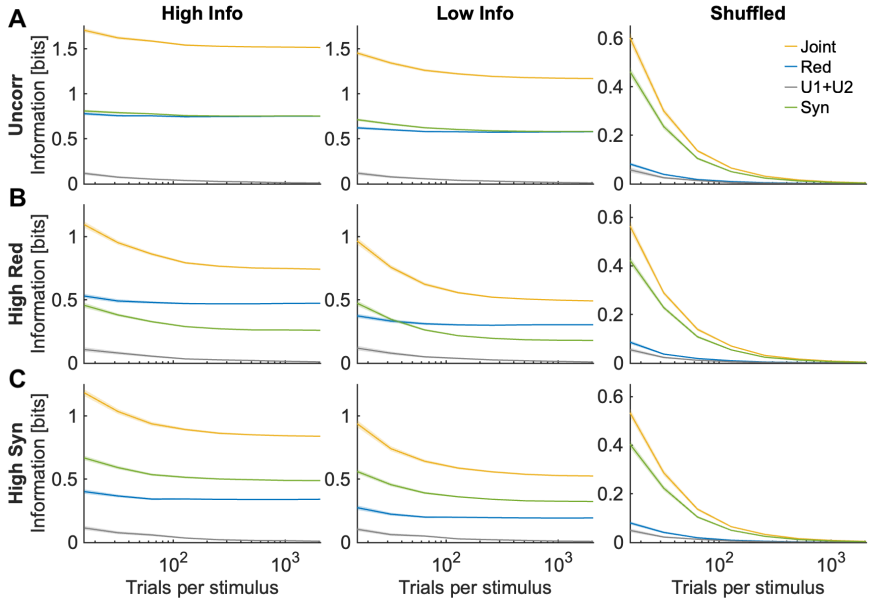

Figure B.7: Joint information and PID quantities using I_{MMI} as the redundancy measure as a function of the number of simulated trials used to compute them. Plotting conventions are exactly as in Fig. 3.1. We used $R = 4$ discretization bins per each neuron (see Table B.1). Results in each panel are plotted as mean ± 2 SEM over $n = 96$ simulations.

B.6 Further Results Bias Corrections

In this subsection we collate figures reporting additional results of the PID bias correction methods on simulated data.

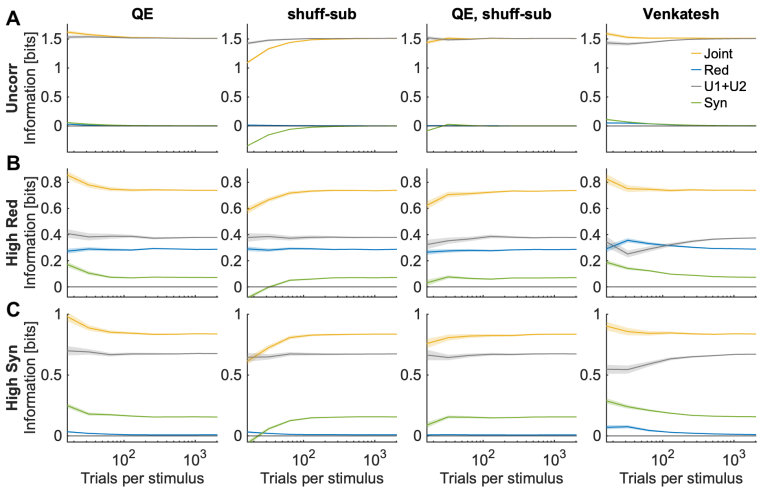

Figure B.8: Bias corrections for the case of high information. Joint information and PID quantities as a function of the number of simulated trials used to compute them. Plotting conventions are exactly as in Fig. 3.2. We used 4 discretization bins for each neuron (see Table B.1). Results in each panel are plotted as mean ± 2 SEM over $n = 96$ simulations.

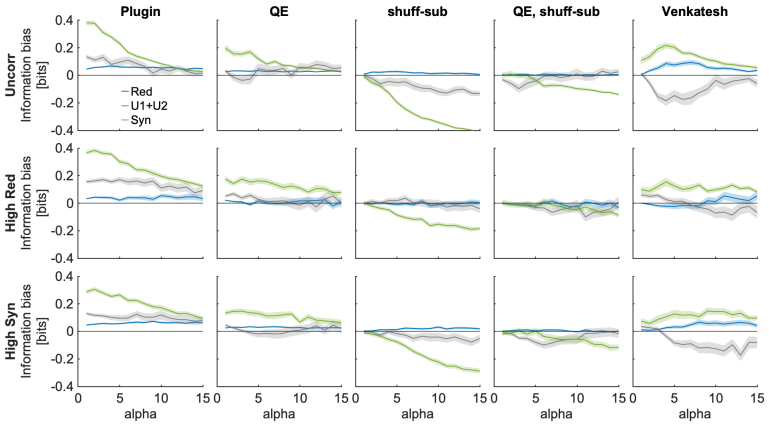

Figure B.9: Performance of bias corrections using $N_s = 16$ trials per stimulus with 4 discretization bins for each neuron, as function of the parameter α increasing single-neuron information in the simulated data. In each panel we plot (rather than the information value itself) the information component bias, computed as the information component estimated with the considered number of simulated trials minus the asymptotic information component estimated with the largest available number of simulated trials, that is 2048 trials per stimulus). Plotting conventions are exactly as in Fig. 3.3.

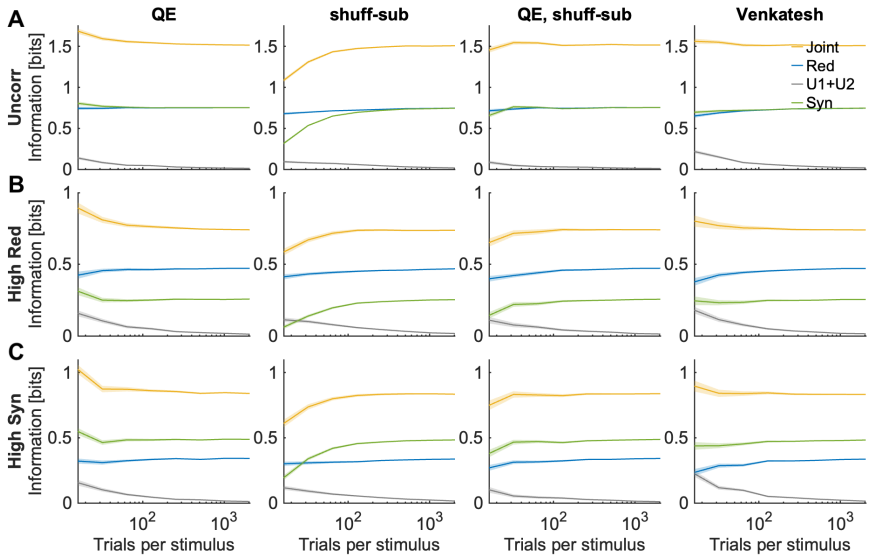

Figure B.10: Bias corrections for the case of high information using I_{min} as the redundancy measure. Joint information and PID quantities as a function of the number of simulated trials used to compute them. Plotting conventions are exactly as in Fig. 3.2. We used $R = 4$ discretization bins per each neuron (see Table B.1). Results in each panel are plotted as mean ± 2 SEM over $n = 96$ simulations.

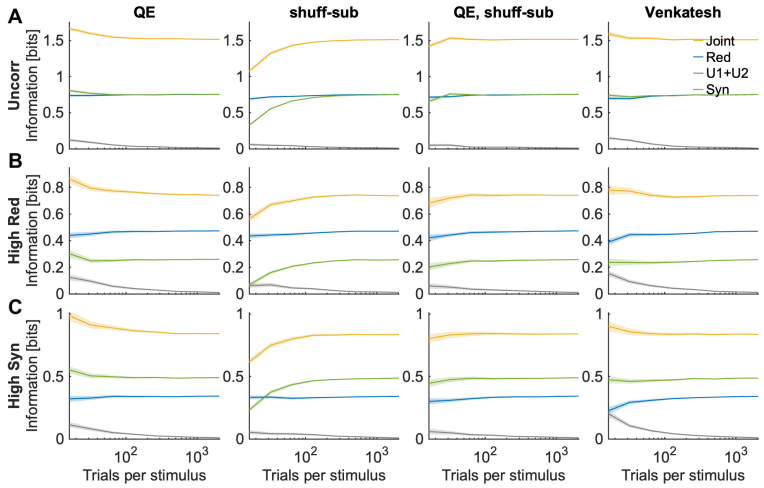

Figure B.11: Bias corrections for the case of high information using I_{MMI} as the redundancy measure. Joint information and PID quantities as a function of the number of simulated trials used to compute them. Plotting conventions are exactly as in Fig. 3.2. We used $R = 4$ discretization bins per each neuron (see Table B.1). Results in each panel are plotted as mean ± 2 SEM over $n = 96$ simulations.

B.7 Simulation showing that the Gaussian approximation for Shannon information fails for neural spike rates with realistically low numbers of spikes

In this subsection, we use simulations of neural activity to exemplify why the Gaussian approximations for information, often used in PID, are not suitable for computing information about stimuli carried by neurons.

We consider here the mutual information about a stimulus s carried by the response of one neuron r_i, which can be quantified as the difference between the entropy $H(R_i)$ of the distribution $p(r_i)$ of responses r_i across of stimuli, called response entropy in the neural literature, and the conditional entropy $H(R_i|S)$ of the stimulus-specific distributions $P(r_i|s)$ referred to as noise entropy in the neural literature:

$$I(S;R_i) = \sum_{s\in S, r_i\in R_i} p(s,r_i)\log\frac{p(s,r_i)}{p(s)p(r_i)} \tag{B.13}$$

$$= -\sum_{r_i\in R_i} p(r_i)\log p(r_i) + \sum_{s\in S, r_i\in R_i} p(s)p(r_i|s)\log p(r_i|s) \tag{B.14}$$

$$= H(R_i) - H(R_i|S) \tag{B.15}$$

It is well documented that the number of spikes emitted by a neuron in response to different stimuli follows approximately a Poisson distribution [147]. Moreover, the time windows in which neurons process and transmit information are usually short of the order of tens to hundreds of milliseconds [55]. Given that the firing rate of cortical neurons ranges from one or few spikes per second (when considering the less effective stimuli) to several tens of spikes per second (when considering the most effective stimuli), firing rate distributions relevant for neural information processing can be conceptualized as near-Poisson with relatively small average mean spike counts. Under these conditions, probability of neural responses conditional to the stimuli are not well approximated by Gaussian distributions, as often assumed by implementations of PID [45, 76, 110, 186]. As a result, the noise entropy will be overestimated (remember that the Gaussian distribution is the one with the highest entropy across all distributions with a given variance). If the different stimuli presented are few in number (that is, S is small) the probability $p(r)$ of response r across all stimuli will be very far from a Gaussian, and as a consequence the response entropy will be overestimated even more than the noise entropy, and as a result the information will be overestimated substantially. Hence, Gaussian PIDs cannot be used to estimate information reliably in typical experiments involving the recording of neural activity, and estimation methods that respect the discrete nature of neural activity will be better suited.

To illustrate this, we simulated spike count responses of an individual neuron in response to a set of $S = 4$ different stimuli defined by two independently-drawn binary features $s1, s2$. We simulated stimulus-specific neural spike counts in response to a stimulus defined by the two feature values $(s1, s2)$ as a Poisson process with average count r,

as follows:

$$r = \alpha s_1 + \alpha s_2 + B \tag{B.16}$$

The parameter B represents the baseline firing and the parameter α represent the separation of the mean responses between the least effective and most effective stimuli. The parameter α increases the information (as it increases the separation between least and more effective stimuli), whereas the parameter B decreases the information (it increases of standard deviations of stimulus-specific responses, which in turn decreases the separation between responses to different stimuli in units of these standard deviations). We used three approaches to compute information. In the first, we computed the ground-truth information by computing the probabilities of each single possible spike count value, which captures perfectly the information expressed by the above Poisson process; the information computed by discretizing the responses into $R = 4$ equipopulated bins (at it could be done reliably also in experiments with more limited numbers of trials) and using the Gaussian approximation (the latter is computed using the information that would be obtained if all probabilities $P(r|s)$ and $P(r)$ were Gaussian with the standard deviation σ and σ_s of measured empirically from the empirical $P(r_i)$ and $P(r_i|s)$, as done in Gaussian PID [45, 76, 186]:

$$I_{gauss}(S;R_i) = \frac{1}{2}\log_s(2\pi\sigma^2) - \frac{1}{2}\sum_{s\in S}\log_s(2\pi\sigma_s^2) \tag{B.17}$$

In Fig. B.12 we plot the values of information obtained with these 3 approaches simulating neural responses across values of the parameters α and B (α and B were varied between 2 and 14 in steps of 1). We computed information using a large number (10000) of simulated trials per stimulus to avoid bias issues. The Gaussian information presented a large overestimation of information with respect to the ground truth information encoded by the Poisson process. On average across all simulated cases, the values of Gaussian information overestimated the ground truth values by 21%.

In contrast, a simple discretization into $R = 4$ equipopulated bins of the neural responses led to a very precise estimation of the ground truth information. On average across all simulated cases, the values of the $R = 4$ discretization underestimated the ground truth values by only 2% (the slight underestimation due to a coarser discretization of discrete data is due to the data processing inequality).

B.8 Asymptotic expansion of the bias of the PID components in the limit of large numbers of experimental trials

To derive analytical approximations to the bias, we make the assumption of a large N limit (where N is the total number of experimental trials available to compute the probabilities), defined formally as the case when $N_s P(r_1, r_2|s) \gg 1$ for each stimulus s and joint response r_1, r_2, where N_s is the number of trials for each stimulus used to compute the stimulus-specific neural response probabilities. (If this is satisfied then it is also satisfied that

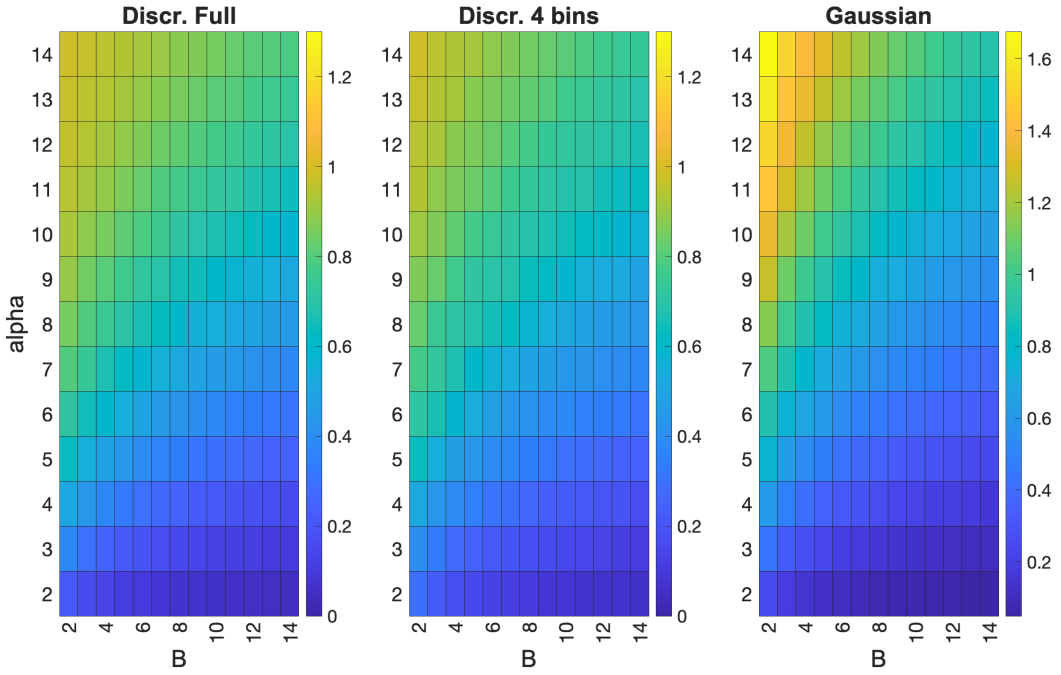

Figure B.12: Comparison between discrete and Gaussian computations of the stimulus information carried by an individual Poisson neuron. We report the information values obtained when evaluating the stimulus information carried by a Poisson neuron simulated with mean spike count set Eq. (B.16) as a function of the parameters B and α. We simulated 10000 trials for each of the 4 stimuli. Left: ground truth information of the Poisson neuron obtained with discrete probabilities considering the probabilities of all possible spike count values. Middle: information computed with the discretized approach by binning the possible spike counts into 4 equipopulated bins. Right: information values computed with the Gaussian approximation of the distributions of spike counts. The Gaussian approach strongly overestimates the ground-truth values of information (by 21% on average across all simulated parameters), whereas the discrete approach using the coarse binning is far more precise (within 2% on average across all simulated parameters)

$NP(r_1, r_2) \gg 1$ for each joint response r_1, r_2, and that the same applies to all the marginal single-neuron probabilities, e.g. $N_s P(r_1|s) \gg 1$). While it is unrealistic that this large N limit case will be formally satisfied in practical experiments, this type of derivation has proven very useful to understand the properties of the bias and to design strategies to cope with it [42, 43, 163].

Suppose we have M trials (for us, M will be either N_s or N depending on whether we will consider noise or response entropy) and we want to compute the entropy in a certain discrete space X (for us, this space will be the joint space of pairwise neural activity or the space of marginal distributions of individual neural activity). The entropy of the probability distribution is defined as

$$H(X) = \sum_x - p(x) \log_2(p(x)) \tag{B.18}$$

where it is important to bear in mind that the sum over possible values of x for the discrete entropy includes only values of x with probability larger than zero.

The values of the probabilities are not known a priori in experimental situations. They could be perfectly computed from the data if we had infinite amounts of experimental trials, but we assume that we have only a finite amount of trials M. If we compute the probabilities from the empirical occurrences $p_M(x)$ from the M available trials, we have the value of the entropy H_M obtained with a finite number of trials M:

$$H_M(X) = \sum_x - p_M(x) \log_2(p_M(x)) \tag{B.19}$$

We would like to compute the finite sampling bias of the entropy, defined as the difference between the average value $\langle H_M \rangle_M$ of H_M (over many instantiations with M trials) and the true value H obtained when the true distributions are known. In what follows, we will compute an approximation to the bias in the large M limit, following the procedure of [42]. However, other methods lead exactly to the same result for the leading approximation to the bias in the large M limit [43, 163]. To compute analytical approximations to the bias of the entropy, we will make use of the logarithm expansion:

$$-\log_2(p_M(x)) = \frac{1}{\ln 2} \sum_{j=1}^{\infty} \frac{(1-p_M(x))^j}{j} \tag{B.20}$$

This is convergent for any value of $0 < p_M(x) < 1$ (note that values of $p_M(x) = 0$ provide a vanishing contribution (thus do not enter) the above sum and if $p_M(x) = 1$ for one response then the entropy is trivially zero). This allows to rewrite the entropy computed from M samples as a sum over powers of the probabilities:

$$H_M(X) = \frac{1}{\ln 2} \sum_x \sum_{j=1}^{\infty} p_M(x) \frac{(1-p_M(x))^j}{j} = \frac{1}{\ln 2} \sum_x \sum_{j=1}^{\infty} \sum_{k=0}^{j} \frac{(-1)^k}{j} \binom{j}{k} p_M(x)^{k+1} \tag{B.21}$$

From the above equation, it follows that averages over all instantiations with M trials of the entropy computed with M trials can be obtained from the averages of the powers of the probabilities over instantiations with M trials. Because we use a discrete computation, we assume without loss of generality that the neural response and the stimulus-specific response probabilities follow a multinomial probability with arbitrary parameters. The value of the probability $p(x)$ computed from the empirical occurrences $p_M(x)$ using M trials on average overall all possible outcomes is unbiased, as its value $\langle p_M(x) \rangle_M$ coincides with the true value $p(x)$that would be estimated with an infinite number of trials:

$$\langle p_M(x) \rangle_M - p(x) = 0 \tag{B.22}$$

When computing with M trials the power k (for $k > 1$) of a discrete probability $p(x)$, on average over all possible outcomes the power of the empirical probability has instead a bias. Under the assumptions that the number of trials M is so large that $Mp(x) \gg 1$ for all

x, the bias of the power k of the probability has the following asymptotic expression [42, 187]:

$$\langle p_M(x)^k \rangle_M - p(x)^k = \frac{1}{M}\binom{k}{2} p(x)^{k-1}[1-p(x)] + o\left(\frac{1}{Mp(x)}\right) \tag{B.23}$$

Higher orders terms of the bias of the probability powers can be computed analytically and are available in Ref. [187]. By inserting the above expansion of the average probability with M trials into the expansion of the entropy one can obtain by some algebra an expansion of the bias of the entropy in inverse powers of $1/M$

$$\langle H_M \rangle_M - H = \frac{C_1}{M} + \frac{C_2}{M^2} + \cdots \tag{B.24}$$

with the leading term $1/M$ in the bias being

$$\frac{C_1}{M} = -\frac{(R_x - 1)}{2M \ln 2} \tag{B.25}$$

where R_x denotes the number of relevant events (responses) for the considered distribution, defined as the number of responses x with probability $p(x) > 0$. Thus, although the bias is not the same for all probability distributions, in the asymptotic sampling regime, its leading term depends on some remarkably simple details of the response distribution (the number of trials and the number of relevant responses). This makes it possible to derive simple rules of thumb for estimating the bias magnitude and compare the relative bias in different situations. As we will discuss now, this equation can predict very effectively how bias properties of PID terms differ between them within a simulated scenario and how they change across different simulated scenarios.

The leading bias terms of the joint information can be computed by writing it as the difference between joint response entropy and joint noise entropy,

$$\begin{aligned} I(S;R_1,R_2) &= H(R_1,R_2) - H(R_1,R_2|S) \\ &= -\sum_{r_1,r_2 \in R} p(r_1,r_2)\log p(r_1,r_2) + \sum_{s\in S, r_1 \in R_1, r_2 \in R_2} p(s)p(r_1,r_2|s)\log p(r_1,r_2|s) \end{aligned} \tag{B.26}$$

and using the equations above with M equal to N_s for the noise entropy and to N for the response entropy. In this way we obtain

$$\langle I_N(S;R_1,R_2)\rangle_N - I(S;R_1,R_2) = \frac{1}{2N\ln(2)}\left[\sum_s (R_{J-s}-1) - (R_J - 1)\right] \tag{B.27}$$

The above bias of $I(S;R_1,R_2)$ originates from finite sampling fluctuations of the joint probabilities and thus is function of R_{J-s}, the number of relevant bins of the joint stimulus-specific distribution in the joint r_1, r_2 space, R_J is the number of relevant bins of the joint stimulus-unconditional distribution in the joint r_1, r_2 space, and N the total number of trials available across all stimuli.

Consider the bias of the single cell entropies (Eq. (B.15) above). They depend on the marginal distribution $p(r_i|s)$ and $p(r_i)$. Thus this bias depends on the finite sampling

fluctuations of the marginal probabilities, which will be smaller than those for the joint probability because the available trials are concentrated in a smaller space (Fig. B.2). Thus, the bias of the single cells will be much smaller than that of the joint information and its leading term is as follows:

$$\langle I_N(S;R_i)\rangle_N - I(S;R_i) = \frac{1}{2N\ln(2)}\left[\sum_s (R_{i-s}-1) - (R_i-1)\right] \tag{B.28}$$

In the above R_{i-s} and R_i are the number of bins relevant for the marginal stimulus conditional and stimulus-unconditional probabilities. These bin numbers will be less than the corresponding ones for the joint probabilities R_{J-s} and R_J. For example, if we discretized responses into 5 bins per neurons, we expect the joint response bins R_{J-s} and R_J to be 25 or less, and we expect R_{i-s} and R_i to be 5 or less. Thus we expect in this case a factor of 5 difference between the joint and individual information bias. In general we can expect the bias of the individual neuron information to be considerably smaller than that of the joint information, with the difference between the two biases getting larger when more bins are used to discretize the activity of each neuron.

Having understood the bias of Shannon information measures relevant for PID, we now use this knowledge to evaluate the bias of the PID terms. We focus first on the bias of the synergy *SI* as defined in BROJA. We remind that the synergy equation is defined as a difference between the joint and the Union information, with the Union information defined as follows:

$$SI(S:R_1;R_2) = I(S;R_1,R_2) - Union \tag{B.29}$$

$$Union = \min_{q\in\Delta_P} I_q(S;R_1,R_2) \tag{B.30}$$

where Δ_P is the set of all joint probability distributions $q(S,R_1,R_2)$ that have the same pairwise marginals $q(S,R_1) = p(S,R_1)$ and $q(S,R_2) = p(S,R_2)$ as the original distribution $p(S,R_1,R_2)$, and $I_q(S;R_1,R_2)$ is the joint information computed for distribution $q(S;R_1,R_2)$. Consider now the Union information. When applied to limited data M, the minimization within the Δ_P space when applied to finite sampling data will tend to select a minimum information within the space set by the marginals $p_M(r_1,s)$ and $p_M(r_2,s)$ computed with M trials. It will thus select probability distributions with a low finite sampling information value. Because this bias is by and large positive, the minimization procedure will tend to select probabilities with lower bias. Among the probabilities with low bias in the Δ_P space we have the probability $p_{ind}(r_1,r_2s)$ defined directly from the marginals such that the neurons would have the same single cell probabilities but no interactions at fixed stimulus (noise correlations)

$$p_{ind}(r_1,r_2|s) = p(r_1|s)p(r_2|s) \tag{B.31}$$

We call the I_q computed on $q = p_{ind}$ the conditionally-independent information $I_{ind}(S;R_1,R_2)$, defined as the joint mutual information with stimulus-conditional probabilities $p_{ind}(r_1,r_2|s)$ and stimulus unconditional probabilities $p_{ind}(r_1,r_2) = \sum_s p_{ind}(r_1,r_2|s)p(s)$. (The information is the information carried by the population about the stimulus if the single neuron responses were the same as the original data but there were no noise correlations

(no stimulus conditional interactions between the neurons), and it has been used as a reference distribution for understanding whether noise correlations increase or decrease information, see [33, 64].) Importantly for our PID bias understanding, we can use the above bias expansion to compute that in the large N limit $I_{ind}(S;R_1,R_2)$ has a low bias, very close to the one of the sum of the biases of individual information:

$$\langle I_{ind-N}(S;R_1,R_2)\rangle_N - I_{ind}(S;R_1,R_2) = \frac{1}{2N\ln(2)}\left[\sum_s\sum_{i=1}^{2}(R_{i-s}-1) - R_{ind}\right] \quad \text{(B.32)}$$

where R_{i-s} is the number of bins relevant for the marginal stimulus conditional and R_{ind} is the one relevant for the stimulus-unconditional independent probabilities of the i-th individual neuron ($i = 1,2$). This will be smaller than the corresponding one for the joint information. For example, if we discretize responses into 5 bins per individual neuron, we would expect 25 bins or less relevant for the joint information and 10 or less (5 for neuron 1 and 5 for neuron 2). In other words, the bias of the joint information grows quadratically with the number of responses of the individual neurons, whereas the bias of $I_{ind}(S;R_1,R_2)$ grows linearly. Because the minimal set of finite sampling variations in the Δ_P space is set by the finite sampling fluctuations of the marginals, we expect and make the assumption that the minimization in the Union calculation will select distributions roughly with the same bias as $I_{ind}(S;R_1,R_2)$. To test this, we performed our simulations of pairs of neurons. We found (see Fig. B.3) across all conditions and trial numbers that the bias of the Union information is very similar to the bias of $I_{ind}(S;R_1,R_2)$. It is actually slightly less which is compatible with the understanding that the minimum selects a similar bias and of course smaller statistical fluctuations.

The synergy is the difference between the joint and the union information. Because the former has a large positive bias (growing quadratically with the number of single-neuron discrete responses) and the latter has a smaller positive bias (growing linearly with the number of single-neuron discrete responses), the synergy is expected to have a relatively large positive bias, smaller than that of the joint information but still large. This prediction is confirmed in all numerical simulations (Fig. 3.1, 3.2, B.3, B.4, B.5, B.6, B.7, B.8, B.10, B.11).

The redundancy is the difference between the sum of single neuron information and the Union information. In the asymptotic regime, the leading term of the bias of the Union is equal to the sum of the leading term of the biases of single neuron information (see Eq. (B.32) above). Thus the redundancy is expected to have relatively small bias. This prediction is also confirmed in all numerical simulations (Fig. 3.1, 3.2, B.3, B.4, B.5, B.6, B.7, B.8, B.10, B.11).

The unique information is the difference between the single neuron information and the redundancy. Given that the redundancy has little bias the unique information should be biased like the single cell information, which grows linearly with the number of discrete single cell responses. Thus the bias of the sum of the unique information should be similar to that of the Union and grow linearly with the number of single neuron responses. Thus, the unique information should have an intermediate bias between synergy and re-

dundancy. Once again, these predictions are too confirmed in all numerical simulations (Fig. 3.1, 3.2, B.3, B.4, B.5, B.6, B.7, B.8, B.10, B.11).

In sum, the above explains why synergy is more biased than the other terms and gives an idea about the bias of the unique information and of the redundancy, suggesting that the bias of synergy is positive, the largest and grows quadratically with the number of single cell responses, the bias of the unique information is positive, the second largest and grows linearly with the number of single cell responses, and redundancy has the smallest bias which grows sub-linearly. It explains why the bias of synergy and joint get proportionally bigger for large numbers of possible discretized single cell responses.

Importantly, the analytical expression also can explain why the synergy bias of the shuffled data (and in general of the bias with lower information levels) are larger than the bias of the original distribution and give an idea of the situations in which they are expected to be tight. Take for example the joint information for which the bias is mostly dictated by the number of relevant response bins R_{J-s} of the stimulus-specific joint response distribution. Informative cases will have stimulus-specific distributions that are restricted to a fewer number of relevant bins than the total number of bins (because informative cases have less variability at fixed stimulus so more restricted and entropic stimulus-specific distributions) and also the bins that are relevant for a specific stimulus would not all be relevant for other stimulus-specific distributions (because informative cases will also show diversity of response distributions to different stimuli). When we randomly shuffle the trials combining trials to different stimuli, the shuffling operation will mix up responses and thus create larger stimulus-specific distributions (larger numbers of stimulus-specific relevant bins) which will then increase the bias, as found in our simulations. This prediction is confirmed by all our simulations. This result is important because it implies that bias corrections based on subtracting shuffled estimators lead a residual downward bias, which is useful to produce lower bounded or conservative estimates. It also implies that the shuffled-subtracted bias corrected estimates will be precise and not too conservative when the information level in the original data is low (because in this case the number of relevant bins of the shuffled and original data will be similar) and it will be too conservative when the information levels in the data are high (because in this case the number of relevant bins of the shuffled distribution will be much larger than the corresponding one for the original distribution). This leads to the design of tighter downward biased estimators of synergy that may be helpful to draw conservative yet sufficiently accurate conclusions.

While we derived these bias properties for the BROJA PID, we expect that the same conclusions would hold across PIDs. In particular, we would expect them to hold for the two other PIDs that we implemented, that is the I_{min} and the I_{MMI}. We verified with simulations that these predictions indeed hold with I_{min} and the I_{MMI} (Fig. B.6, B.7, B.10, B.11). In fact in both these decompositions the Union information depends on the two marginal probabilities $p(r_1,s)$ and $p(r_2,s)$ (because they satisfy the so called pairwise marginals property). Thus all the considerations we made for the bias size of the BROJA Union are expected to hold also for the Union of these other definitions, because Union information definitions that satisfy the pairwise marginals property should have

bias determined by the size of the finite size fluctuations of the marginals. The properties discussed for the BROJA held also for these other PIDs. If anything, we found that the bias of the synergy was even closer to the one of the joint information as the I_{min} and the I_{MMI} have the well documented problem of producing small values of unique information [110, 141].

B.9 Venkatesh et al NeurIPS 2023 procedure for PID bias correction reformulated the discrete case

We tested the effectiveness in correcting for the bias of the PID bias correction procedures developed by Venkatesh and colleagues [45]. The procedure was formulated for Gaussian PID. We report it here for completeness, describing also the straightforward adaption of it that we did to use it in the discrete PID. In the following we use the subscript N to indicate the information values obtained by plugin of the empirical probabilities estimated from N trials s without using any bias correction, and (following Ref. [45]) we use the subscript $bias-corr$ to indicate the information quantities corrected for the bias with the considered specific procedure.

The Venkatesh bias correction procedure [45] focuses first on the bias of the Union information. We consider union information $Union(S:R_1;R_2)$ which is related to the other PID quantities from the following equation

$$Union(S:R_1;R_2)_{plugin} = UI(S:R_1 \setminus R_2)_{plugin} + UI(S:R_2 \setminus R_1)_{plugin} + RI(S:R_1;R_2)_{plugin} \tag{B.33}$$

and is computed in the BROJA definition from Eq (3.6). In the following, we will indicate union information by omitting their dependency on source and target variables, as $Union$, for brevity. The Venkatesh procedure corrects for the bias by first computing the bias-corrected union information (see Eq. (17) and the rectification formulae Eq. (106, 107) of [45]) by computing the bias of the joint information with any of the method available for Shannon information (e.g. QE), then making the assumptions that it has the same bias as the joint information and rescaling it accordingly (Eq. (17) in [45]):

$$Union_{(0)} = Union_{plugin} \frac{I(S;R_1,R_2)_{bias-corr}}{I(S;R_1,R_2)_{plugin}} \tag{B.34}$$

Note that, to follow precisely what was done by Venkatesh and colleagues [45], when applying the Venkatesh procedure if a Shannon information quantity had a negative value after the bias correction procedure, we set it to zero. Eq. (B.34) assumes that the fraction of bias of the union information is the same as the one of joint information. As we found in our analytical calculations and simulations (Fig. B.3), this assumption is incorrect, as union information is much less biased than joint information. Thus this first step will in most cases and for low trial numbers lead to a severe underestimation of union information. Then, to make sure all PID quantities after bias correction are non-negative and satisfy the PID linear constraints, the Venkatesh procedure applies a double post-hoc

rectification of the bias-corrected value of the union information:

$$Union_{(1)} = \max\{Union_{(0)}, I(S;R_1)_{bias-corr}, I(S;R_2)_{bias-corr}\} \tag{B.35}$$

$$Union_{(2)} = \min\{Union_{(1)}, I(S;R_1)_{bias-corr} + I(S;R_2)_{bias-corr}, I(S;R_1,R_2)_{bias-corr}\} \tag{B.36}$$

The resulting $Union_{(2)}$ is the value we take in the following for bias corrected union $Union_{bias-corr}$. From this the Venkatesh procedure computes the 4 bias-corrected PID terms using the bias-corrected union information and the bias corrected joint and individual information using the 4 linear PID constraints:

$$\begin{aligned} RI(S:R_1;R_2)_{bias-corr} &= I(S;R_1)_{bias-corr} + I(S;R_2)_{bias-corr} - Union_{bias-corr} \\ UI(S:R_1 \setminus R_2)_{bias-corr} &= Union_{bias-corr} - I(S;R_2)_{bias-corr} \\ UI(S:R_2 \setminus R_1)_{bias-corr} &= Union_{bias-corr} - I(S;R_1)_{bias-corr} \\ SI(S:R_1;R_2)_{bias-corr} &= I(S;R_1,R_2)_{bias-corr} - Union_{bias-corr} \end{aligned} \tag{B.37}$$

The problems of the Venkatesh procedure can be readily understood by considering scenarios in which both ground truth unique information components are larger than zero. Then, the major underestimation of $Union_{(0)}$ in Eq. (B.34) due to the incorrect assumption that the upward bias of the union information scales like the bias of the joint information (whereas in reality, as demonstrated in the previous section and in Fig. 3.1, B.3, B.4, B.5, B.6, B.7, the former is small and scales linearly with the number of discretized single-neuron responses and the latter is large and scales quadratically) will lead to very low values of $Union_{(0)}$. Then the first rectification in Eq. (B.35) would take $Union_{(1)}$ as the maximum between the information carried individually by the two sources, when in reality the unbiased union information should be larger than both single-source information, since both unique info are larger than zero. This would ultimately lead to a grossly underestimated $Union_{bias-corr}$, which would result, from Eq. (B.37) into an underestimation of the bias corrected unique information components (for which $Union_{bias-corr}$ has a positive sign in Eq. (B.37)) and an overestimation of the redundant and synergistic components (for which $Union_{bias-corr}$ has a negative sign in Eq. (B.37)). The major overestimation of synergy and redundancy and the major underestimation of unique information of the Venkatesh procedure have been consistently confirmed by our simulations (Fig. 3.2, 3.3, B.8, B.9, B.10, B.11).

B.10 Details of experimental procedures for real neural data recorded from the mouse brain

B.10.1 Details of experimental procedures of mouse auditory cortex data recorded during a sound discrimination task

We reanalysed a previously published dataset [129] in which the activity of several tens to a few hundreds of neurons was recorded simultaneously using in vivo two photon calcium imaging from A1 L2/3 neurons in head-fixed transgenic mice during a pure-tone

discrimination task (Fig. 3.4A). The experimental task was structured as follows. After a pre-stimulus interval of 1 s, head-fixed mice were exposed to either a low-frequency (7 or 9.9 kHz) or a high-frequency (14 or 19.8 kHz) tone for a period of 1 s. Mice were trained to report their perception of the sound stimulus by their behavioural choice, which consisted of licking a waterspout in the post-stimulus interval (0.5–3 s from stimulus onset) after hearing a low-frequency tone (target tones) and holding still after hearing high-frequency tones (non-target tones). Two-photon calcium imaging was used to continuously acquire the calcium fluorescence signals from individual A1 L2/3 neurons during the task with an imaging frame rate of 30 Hz. Calcium fluorescence traces were deconvolved as reported in [129] to estimate spike rates in each imaging frame. Full details of the experimental procedures and details of the ethical approval are reported in full in the original publication [129].

We analysed neurons recorded from $n = 12$ Fields of View (FOV) from $n = 12$ animals. For consistency with the previously published work [129] for the PID analysis reported in this paper (Fig. 3.4) we only considered individual neurons that carried significant intersection information [156], that is information about the stimuli that is used to inform the behavioral discrimination, according to the statistical permutation tests described in [129]. This led to selecting for PID analysis $n = 375$ individual neurons (out of the 2792 recorded neurons) leading to $n = 6209$ pairs of simultaneously recorded neurons used for the PID analysis.

For the PID analysis of these data, we identified for each neuron the imaging time frame within the trial of maximal intersection information [37] exactly as described in [129]. We then considered for each neuron a time frame of $n = 10$ imaging frames (corresponding to a window of 333 ms) around the peak intersection information time frame (we call this the peak time window for the neuron). Then we discretized activity for each neuron into $R = 3$ bins according to whether it was detected 0, 1 or > 1 spikes in the peak time window. The stimulus set used for this analysis was binary ($S = 2$), dividing the presented sound tones into the low- ($s = 0$) and high-frequency ($s = 1$) categories. For consistency with the original experimental publication [129], which analyzed joint stimulus information carried by pairs of neurons without breaking it up with the PID, we estimated the total stimulus information that was jointly carried by pairs of neurons following a time-lagged approach. Namely, we computed the time-lagged stimulus information carried jointly by the activity of each pair of neurons using Eq. (3.2) and using as responses r_1 or r_2 the discretized responses of each neuron measured as detailed above in their respective peak time windows.

B.10.2 Details of experimental procedures of mouse posterior parietal cortex data recorded during a decision-making task requiring sound localization

We analyzed previously published [35] neural recordings from a sound localization task in which mice reported perceptual decisions about the location of an auditory stimulus by navigating through a visual virtual reality T-maze (Fig. 3.4C). As mice ran down the

T-stem, a sound cue was played from one of eight possible locations in head-centered coordinates. Mice reported whether the sound originated from their left or right by turning in that direction at the T-intersection. During each session, the activity of approximately 50 (range 37-69) layer 2/3 neurons was imaged simultaneously using two photon microscopy and the GCamp6f indicator. Calcium fluorescence traces were deconvolved as described in Ref. [35] to estimate spike rates in each imaging frame.

We computed the mutual information between the joint activity of two simultaneously recorded neurons r_1 and r_2 and the stimulus category of the sound location ($S = 2$; in other words we computed information about whether the sound came from left or right of the midline, which is what the mice were asked to categorize in their task). This information is relevant for posterior parietal cortex, because this is an area that has been described as a sensory-motor interface that converts sensory information into signals relevant for decision-making. For the PID analysis of these data, for consistency with the above reported analysis of auditory cortex data, we identified for each neuron the imaging time frame within the trial of maximal stimulus information. We then considered for each neuron a time frame of $n = 5$ imaging frames (corresponding to a window of 320 ms) around the peak information time frame (we call this the peak time window for the neuron). Then we discretized activity for each neuron into $R = 3$ bins according to whether it was detected 0, 1 or > 1 spikes in the peak time window. Again, for consistency with the auditory cortex analysis, we estimated the total stimulus information that was jointly carried by pairs of neurons following a time-lagged approach. Namely, we computed the time-lagged stimulus information carried jointly by the activity of each pair of neurons using Eq. (3.2) and using as responses r_1 or r_2 the discretized responses of each neuron measured as detailed above in their respective peak time windows.

We analyzed neurons recorded from $n = 11$ Fields of View (FOV) from $n = 11$ animals. For consistency with the original publication reporting these data, we did not perform any selection of these neurons and we used all $n = 713$ neurons available in the published database, which allowed us to use $n = 10750$ pairs of simultaneously recorded neurons for the PID analysis.

B.10.3 Details of experimental procedures of mouse hippocampus data recorded during spatial navigation in a virtual environment

We reanalysed a previously published dataset [69] in which the activity of several tens to a few hundreds of neurons was recorded simultaneously using in-vivo two-photon calcium imaging from CA1 neurons in head-fixed transgenic mice during virtual reality navigation of a linear track (Fig. 3.4A). Calcium fluorescence signals were obtained using the jRCaMP1a calcium indicator and an imaging frame rate of $0.333s$. Full details of the experimental procedures and details of the ethical approval are reported in full in the original publication [69].

We analyzed neurons recorded from $n = 11$ Fields of View (FOV) from $n = 7$ animals. Because hippocampal neurons encode position in space, we computed the mutual

information between the spatial position S in which the animal was located in the linear track and the joint activity of two simultaneously recorded neurons r_1 and r_2. For consistency with the previous study reporting the original data [69], published study, the spatial position s along the linear track was computed by binning the space along the track into $S = 12$ equipopulated bins. For consistency with the previous study [69], to quantify the neural activity r_i of each neuron, we took the calcium traces and binned them into $R = 2$ equipopulated bins (only raw calcium traces and not deconvolved signals were available from Ref. [69]). For the PID analysis, we used all the $n = 870$ individual neurons present in the dataset leading to $n = 36158$ pairs of simultaneously recorded neurons used for the PID analysis.

B.11 Simulated example of application of PID to within-network synergistic information transfer

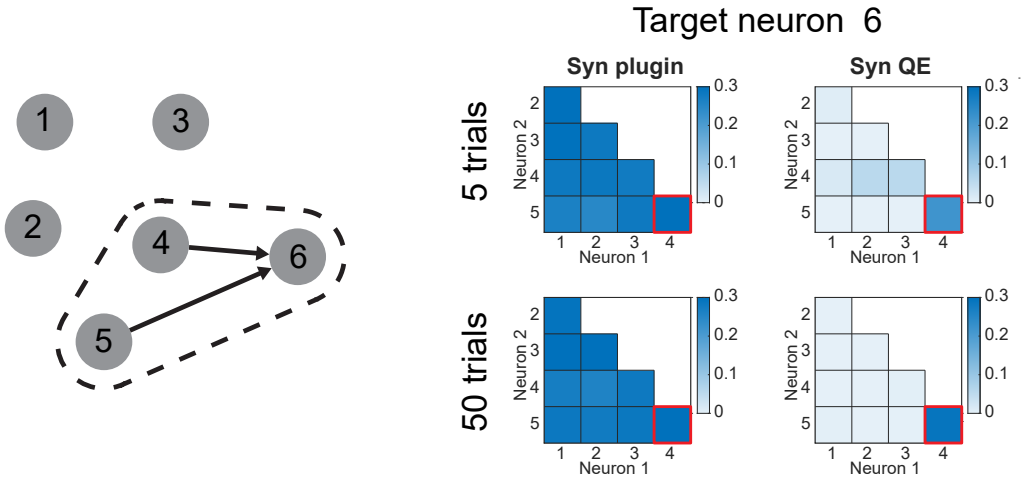

Figure B.13: Example of using PID to describe triplet interactions in a network. On the left is shown a schematic of the network, where only neuron 6 is influenced by other neurons (neurons 4 and 5). On the right, the values obtained by setting neuron 6 as the PID target and going through all other possible cell pairs as sources. On the top row, the values are shown for the mean of 5 trials and on the bottom row, the mean across 50 trials. The left panels show the synergy values obtained naively without any bias correction. On the right panels, the corrected values using QE are shown. A red square points to the cell pair that has a true influence on cell 6.

To illustrate and test the usefulness of using PID to map information transfer within a network [78, 137], we simulated a small network of neurons and computed the PID using the activity of another neuron (and not a sensory stimulus (S) as a target variable). We compared results using or not using sampling bias corrections.

We simulated a network (Fig. B.13 of 6 nodes (to be conceptualized as different neurons or brain regions). Each node was modeled as Gaussian processes across time (100 timesteps). Nodes 1 to 5 are independent of each other with mean zero and unit

standard deviation, while the activity of neuron 6 at time t is defined as the sum of the activities of neurons 4 and 5 at time $t-1$ plus noise. Thus, the ground truth in this simulated dataset is that nodes 4 and 5 exert a synergistic information transmission on neuron 1, and there is no transmission of information between any other set of nodes. Thus, if we select neuron 6 as a target for our PID analysis, only the cell pair 4-5 should have significant synergy values [110].

The PID analysis was done using the BROJA redundancy measure [31], binning the neural activities with 4 equipopulated bins. The target was the activity of neuron 6 from time 2 to 100 and the sources were taken from the other cells from time 1 to 99 to account for the time-lagged nature of the interactions. The first row shows the result of running 5 trials of the same network and averaging its values. The second row shows the mean results of 50 trials.

The top left heatmap on Fig. B.13 shows the synergy values for each possible cell pair targeting neuron 6 averaged across 5 trials. All the cell pairs have comparable, positive synergy values, which are significant according to a one-tailed t-test. When applying a QE correction of one instantiation of the simulation, only the pair corresponding to the real connections is still significantly above zero (one-tailed t-test $p < 0.01$). When increasing the number of simulations to 50, the plugin PID is still unable to assign a significant synergy value only to the real source neurons (Fig. B.13 bottom left, all pairs have significant values with $p < 0.001$). After the QE correction, only the real cell pair remains significantly above zero (one-tailed t-test $p < 0.001$).

Together, these results show that the bias corrections are useful also to obtain better estimates of information about target variables being the activity of other neurons (as often done in neuroscience [78, 137]) and that the bias corrections are useful to identify patterns of synergy within larger networks.

B.12 Data and code availability

Code for simulating and analysis is available from the corresponding author upon reasonable request. The real neural data used here in Fig. 3.4 were published before. Data of [129] were released with the original publication and can be found at `https://doi.org/10.1101/2021.08.31.458395`. Data of [35] were republished in [130] and can be found at `https://doi.org/10.12751/g-node.tqbad8`. Data of [69] are available from the corresponding author upon reasonable request.