Neuroinformatics (2025) 23:40
https://doi.org/10.1007/s12021-025-09738-1

RESEARCH

Scikit-NeuroMSI: A Generalized Framework for Modeling Multisensory Integration

Renato Paredes[1,2] · **Juan B. Cabral**[3,4] · **Peggy Seriès**[5]

Accepted: 14 July 2025 / Published online: 24 July 2025

Abstract
Multisensory integration is a fundamental neural mechanism crucial for understanding cognition. Multiple theoretical models exist to account for the computational processes underpinning this mechanism. However, there is an absence of a consolidated framework that facilitates the examination of multisensory integration across diverse experimental and computational contexts. We introduce Scikit-NeuroMSI, an accessible Python-based open-source framework designed to streamline the implementation and evaluation of computational models of multisensory integration. The capabilities of Scikit-NeuroMSI were demonstrated in enabling the implementation of multiple models of multisensory integration at different levels of analysis. Furthermore, we illustrate the utility of the software in systematically exploring the model's behavior in spatiotemporal causal inference tasks through parameter sweeps in simulations. Particularly, we conducted a comparative analysis of Bayesian and network models of multisensory integration to identify commonalities that may enable to bridge both levels of description, addressing a key research question within the field. We discuss the significance of this approach in generating computationally informed hypotheses in multisensory research. Recommendations for the improvement of this software and directions for future research using this framework are presented.

Keywords Multisensory integration · Causal inference · Scientific software · Computational neuroscience · Computational models

Introduction

Multisensory integration is the neural process by which signals originating from distinct sensory modalities (such as visual, tactile, or auditory) are merged. As a result, the multisensory response can differ significantly from the responses elicited by stimuli confined to a single sensory modality (Stein and Stanford, 2008; Stein et al., 2010). Disturbances in multisensory processing can impact various cognitive domains (Wallace et al., 2020). For example, alterations in multisensory function are observed in various neuropsychiatric and neurological disorders (e.g. SCZ, ASD, dementia, sensory loss, dyslexia) (Martin et al., 2013; Haßet al., 2017; Zvyagintsev et al., 2017; Paredes et al., 2022; Cascio et al., 2012; Stevenson et al., 2014; Hahn et al., 2014; Zhou et al., 2018; Noel et al., 2022a; Wu et al., 2012; Festa et al., 2017; Ramkhalawansingh et al., 2017). Furthermore, a substantial body of research is exploring the potential of multisensory markers to predict future clinical manifestations or to serve as key focal points for therapeutic interventions (Bolognini et al., 2005; Sánchez et al., 2013; Gieseler et al., 2018).

Computational modeling is crucial for the advancement of the field due to its potential to develop formal theories of the neural mechanisms of multisensory integration (Meijer and Noppeney, 2020; Colonius and Diederich, 2020). It

✉ Renato Paredes
renato.paredes@pucp.edu.pe

1 Departament of Psychology, Pontifical Catholic University of Peru, Lima, Peru

2 Instituto de Investigaciones Psicológicas, Facultad de Psicología, Universidad Nacional de Córdoba, Córdoba, Argentina

3 Grupo de Innovación y Desarrollo Tecnológico, Gerencia De Vinculación Tecnológica, Centro Espacial Teófilo Tabanera, Comisión Nacional de Actividades Espaciales (CONAE), Córdoba, Argentina

4 Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Buenos Aires, Argentina

5 School of Informatics, University of Edinburgh, Edinburgh, United Kingdom

forces scientists to analyze, specify, and formalize their ideas, while also allowing them to generate precise quantitative predictions suitable for testing in future experiments (Blohm et al., 2020; Guest and Martin, 2021). Consequently, the body of literature on computational models of multisensory integration has experienced substantial growth over the past two decades (Colonius and Diederich, 2020).

The main modeling approaches are optimal cue combination (Ernst and Banks, 2002; Alais and Burr, 2004; Fetsch et al., 2013; Parise and Ernst, 2016), Bayesian causal inference (Körding et al., 2007; Shams and Beierholm, 2010; Rohe and Noppeney, 2015; Rohe et al., 2019; Meijer and Noppeney, 2020), race (Diederich, 1992, 1995; Colonius and Diederich, 2004; Colonius et al., 2017), and network (Ma and Pouget, 2008; Ma and Rahmati, 2013; Ohshiro et al., 2011; Miller et al., 2017; Cuppini et al., 2017; Ursino et al., 2019) models. Despite being based on very general mechanisms, these models are typically restricted to a specific experimental paradigm (e.g. spatial localization, orientation judgments, temporal order judgments, rate detection, sound-induced flash illusion, among others) and vary in their level of description (e.g. computation, implementation or algorithm) and sophistication.

Overall, the field lacks a unified theoretical approach to multisensory integration that allows for model testing in different experimental and computational paradigms. There is a growing need for scientific software specifically designed to represent the distinctive concepts and mechanisms that characterize the integration of information from different sensory modalities. To our knowledge, there is as yet no software that can provide a unified computational environment designed to facilitate the examination of differences in model predictions. Multisensory integration modelers currently rely on packages built for general-purpose computational neuroscience modeling, such as Brian Stimberg et al. (2019), HDDM (Wiecki et al., 2013), TAPAS (Frässle et al., 2021) or PyRates (Gast et al., 2019), BrainPy (Wang et al., 2023), among others. There are also alternatives for specific multisensory integration models, such as the Bayesian Causal Inference Toolbox (Zhu et al., 2024a), but no frameworks encompassing more than one modeling approach.

Here we present Scikit-NeuroMSI (Paredes et al., 2023), an open-source Python (Rossum and Drake, 2010) framework that simplifies the implementation of neurocomputational models of multisensory integration. The package currently allows to run seminal computational models of multisensory integration and to easily implement new models defined by users. As an illustration, we show how this framework facilitates the analysis of spatiotemporal multisensory integration at different levels of description using Bayesian and network models.

This paper targets a diverse audience in neuroscience and computational fields, including computational neuroscientists and researchers in sensory processing. It is also beneficial for software developers and engineers interested in multisensory integration models. While basic neuroscience and programming knowledge is useful, proficiency in Python is needed to understand the implementation of Scikit-NeuroMSI, as it is developed in this language. We make the paper accessible by covering theoretical foundations and practical implementations, with explanations and code examples to aid comprehension. Those with Python skills can dive directly into the implementation details, while newcomers will find enough context to grasp the concepts. This approach highlights the challenge of standardizing multisensory integration models, which requires a blend of neuroscience, mathematics, and software engineering.

Software

Design Overview

Experiments on multisensory integration investigate how the brain combines information from multiple sensory modalities to create a unified perception of the environment. These studies often involve manipulating the reliability or congruency of sensory cues from different modalities, such as vision and touch, to examine how the brain weighs and integrates this information (Colonius and Diederich, 2020). Experimental paradigms include spatial localisation tasks for cross-modal spatial interactions (Ernst and Banks, 2002; Alais and Burr, 2004), temporal order judgments for timing perception of multisensory events (Ferri et al., 2016), or the double flash illusion for auditory influence on visual perception (Shams et al., 2002), among others.

Computational models play a crucial role in multisensory research by providing frameworks to predict and explain behavioral and neural responses in multisensory contexts (Chandrasekaran, 2017). These models help researchers understand the underlying principles of sensory integration, such as reliability-based cue weighting and inverse effectiveness (Stein et al., 2020). By comparing model predictions with empirical data, researchers can test hypotheses about the neural mechanisms of multisensory integration and gain insights into how the brain creates coherent perceptions from diverse sensory inputs (Blohm et al., 2020).

A wide range of computational models for multisensory integration is available (Colonius and Diederich, 2020), providing descriptions at computational, algorithmic, and implementation levels within information-processing theory (Marr, 2010). Nevertheless, it is unusual to find formal evaluations of multisensory integration using more than one model at the time to bridge across levels of analysis (Ursino et al., 2014).

Scikit-NeuroMSI was designed to meet three fundamental requirements in the computational study of multisensory integration:

1. **Modeling Standardization:** Researchers need to compare different theoretical approaches (Bayesian, neural network, maximum likelihood estimation, among others) using consistent analysis methods. Our framework provides a standardized interface for implementing and analyzing different types of models.
2. **Data Processing Pipeline:** The framework handles multidimensional data processing across:
   - Spatial dimensions (1D to 3D spatial coordinates)
   - Temporal sequences
   - Multiple sensory modalities (e.g., visual, auditory, touch)
3. **Analysis Tools:** We provide integrated tools for:
   - Parameter sweeping across model configurations
   - Result visualisation and export
   - Statistical analysis of model outputs

Furthermore, the software processes fundamental attributes of experimental inputs pertinent to multisensory research: spatial coordinates (e.g., degrees of visual angle and sound source location), temporal properties (encompassing stimulus onset, duration, and inter-stimulus intervals), stimulus intensity, and spatiotemporal reliability (random noise). Each input is validated to ensure compliance with the formatting standards, followed by a transformation to standardized internal representations that facilitate effective model processing. The software systematically applies type checking and data validation protocols to ensure the integrity of the inputs before proceeding with processing.

Consequently, the software requires a uniform output for each multisensory integration model. This output incorporates the activity values from all participating modalities, necessitating at least two unisensory modes along with one multisensory mode. There is an optional provision for output related to causal inference responses, as determined by the user. These prerequisites guided the technical design choices detailed in the next section.

A Formal and Technical Approach for the Model Standardization

We outline the theoretical and mathematical basis of multisensory integration models. We first describe the formal properties and mathematical framework needed to understand these computational problems theoretically, which aids in addressing software engineering challenges. Our development of `Scikit-NeuroMSI` aimed to articulate these problems with minimal conceptual differences, removing programming overhead to focus on solving multisensory integration issues (Brooks and Kugler, 1987). Specific implementation details and practical considerations are discussed in Section "Implemented Models", with examples of applying these principles in the software framework.

Computational Model Formalities

This work aims to create a unified framework for multisensory integration models (Colonius and Diederich, 2020) through Scikit-NeuroMSI, facilitating interoperability between analysis, comparison, and explanation tools, regardless of the specific model.

Consider two distinct model outputs $r0$ and $r1$, such as Bayesian models (Körding et al., 2007; Shams and Beierholm, 2010), neural networks (Cuppini et al., 2014, 2017), or maximum likelihood estimators (Ernst and Banks, 2002; Alais and Burr, 2004), with different modalities and dimensions. Any processing function f should handle both $r0$ and $r1$ equally well.

In general, our goal is to create models (m) that produce results (r) compatible with any processing function f, ensuring that the new m or f are mutually compatible (see Appendix A for a mathematical formulation). For example, Code 1 shows that neural network models (Cuppini et al., 2017) and maximum likelihood estimation models (Alais and Burr, 2004) can be analyzed with the same parameter sweep tools.

Code 1 Example showing how Scikit-NeuroMSI enables the use of consistent analysis tools across different types of multisensory integration models, regardless of their underlying theoretical framework.

```
# import all the needed models and tools
>>> from skneuromsi import neural, mle, sweep

# we create two models that are totally different in nature
>>> m0 = neural.Cuppini2017()
>>> m1 = mle.AlaisBurr2004()

# the utility sweep works for any model
>>> sweep.ParameterSweep(m0, target="auditory_position")
>>> sweep.ParameterSweep(m1, target="auditory_position")
```

Another goal was to allow users to choose the sensory modality they simulate, reflecting real-world scenarios. For example, a researcher might study visual-auditory stimuli interaction in one experiment and visual-tactile in another. This allows models to adapt parameters to dynamically match selected modalities. As demonstrated in Code 2 an audio-visual setup reveals parameters such as `"auditory_position"`, while a visual-tactile setup replaces them with parameters such as `"tactile_position"`, keeping the core model structure and logic intact.

Code 2 Example of mode configuration in a model: The same neural network model (`Cuppini2017`) can be configured for different sensory modalities. Compare how the run function parameters automatically adapt between auditory-visual (top) and tactile-visual (bottom) configurations while maintaining the same underlying model architecture. Note that the outputs of `m0.run` and `m1.run` show the function signatures since they are not executed. To execute the function, the code should be `m0.run()` or `m1.run()`

```
# import all the needed models and tools
>>> from skneuromsi import neural, mle, sweep

# The same model with two different modes0 configurations
# the models can have an arbitrary number of modes
>>> m0 = neural.Cuppini2017(mode0="auditory")
>>> m1 = neural.Cuppini2017(mode0="tactile")

# check the parameters of the run function (all auditory)
>>> m0.run
<function skneuromsi.neural._cuppini2017.Cuppini2017.run(*,
 ↪  auditory_position=None, visual_position=None, auditory_sigma=32,
 ↪  visual_sigma=4, auditory_intensity=28, visual_intensity=27,
 ↪  auditory_duration=None, auditory_onset=0, auditory_stim_n=1,
 ↪  visual_duration=None, visual_onset=0, visual_stim_n=1,
 ↪  noise=False, feedforward_weight=18, cross_modal_weight=1.4,
 ↪  causes_kind='count', causes_dim='space')>

# 'auditory' change for tactile
>>> m1.run
<function skneuromsi.neural._cuppini2017.Cuppini2017.run(*,
 ↪  tactile_position=None, visual_position=None, tactile_sigma=32,
 ↪  visual_sigma=4, tactile_intensity=28, visual_intensity=27,
 ↪  tactile_duration=None, tactile_onset=0, tactile_stim_n=1,
 ↪  visual_duration=None, visual_onset=0, visual_stim_n=1,
 ↪  noise=False, feedforward_weight=18, cross_modal_weight=1.4,
 ↪  causes_kind='count', causes_dim='space')>
```

The final design requirement ensures multidimensional analysis of data. Each model output point is associated with a specific mode, time, and space, with the spatial dimension possibly encompassing one to three dimensions. Consequently, the model output can comprise up to five dimensions (5D).

Object-Oriented Design

Selecting an ecosystem for a tool is a subjective decision. Most data analysis projects use Python due to its rich scientific ecosystem (Perez et al., 2010). We chose Python to exploit object-oriented mechanisms, enhancing simplicity and tool extensibility. In object-oriented languages such as Python, classes structure data types by grouping attributes (state) and methods (behavior), enabling cohesive code organization and providing inheritance for sharing functionalities without code duplication (Booch, 1982).

A conceptual mechanism available in object-oriented programming languages that we employed is abstract classes. These represent data types that, although possessing a complete protocol (i.e., their functions, the data they receive, and the return types are fully defined), have certain behaviors that remain unspecified. For example, as demonstrated in Code 3, the `Foo` class utilizes the functionality already established in the `method0()` method without necessitating redundancy in the source code. Furthermore, this establishes a hierarchical structure, or inheritance, between the `FooABC` and `Foo` types, indicating that any object created or instantiated by `Foo` is an instance of the `FooABC` type. The final aspect to note is that `FooABC` is explicitly declared as an incomplete entity and is thus non-instantiable, achieved by adorning `method1()` with `@abstractmethod`.

Code 3 Example of abstract vs concrete class implementation in Python: The abstract class (`FooABC`) defines a protocol with both concrete and abstract methods, while its concrete implementation (`Foo`) provides the required implementation of the abstract method. Note how creating an instance of the abstract class fails, while the concrete class is instantiated successfully.

```
# modules that expose abstract class
# functionality in Python
from abc import ABC, abstractmethod

class FooABC(ABC):  # FooABC is an Abstract-Base-Class
    def method0(self):
        print("This is a concrete method")

    @abstractmethod
    def method1(self): ...

class Foo(FooABC): # Foo inherits from FooABC
    def method1(self):
        return self.method0()

# Creating objects
foo = FooABC() # This fails, method1 is not defined
foo = Foo()    # Works

# calling the methods
foo.method0()  # Works
foo.method1()  # Works
```

We followed the "Dependency Inversion Principle" (DIP), an essential SOLID principle (Martin, 2000), which establishes two fundamental rules:

1. High-level modules should not depend on low-level modules. Both should depend on abstractions.
2. Abstractions should not depend on details. Details should depend on abstractions.

To illustrate its importance, we compare two model processing tool implementations. Code 4 is tightly coupled, relying directly on a specific model, while Code 5 is more flexible, using dependency injection via an abstract base class.

Code 4 Example of a tool without Dependency Injection: The tool function is tightly coupled to a specific model implementation (`Model1`), making it inflexible and difficult to extend to other model types.

```
# Example without Dependency Inversion
class Model1:
    def process(self, data):
        return data * 2

def tool_for_model1(data):
    # Tool is tightly coupled to Model1
    model = Model1()
    return model.process(data)
```

The main distinction between these methodologies lies in that Code 4 contains a tool function specifically hard-coded to operate solely with `Model1`, thereby precluding its application to other model types unless the function is modified. In contrast, Code 5 is designed to accommodate any model inheriting from `ModelABC`, facilitating the seamless integration of new model types, allowing runtime model substitution, enhancing testability through mock objects, and ensuring a more robust separation of concerns. This pattern is essential for Scikit-NeuroMSI, as it enables the consistent implementation of tools across various multisensory integration models.

Code 5 Example of a tool with Dependency Injection: Depending on an abstraction (`ModelABC`), the tool function can work with any model that implements the required interface, allowing flexible model substitution and easier testing.

```
# Example with Dependency Inversion
from abc import ABC, abstractmethod

class ModelABC(ABC):
    @abstractmethod
    def process(self, data):
        pass

class Model1(ModelABC):
    def process(self, data):
        return data * 2

def tool(model: ModelABC, data):
    # Tool depends on abstraction (ModelABC)
    return model.process(data)
```

We propose a set of classes to reduce the "semantic gap" that separates the ideas of how a multisensory integration problem is expressed and how the code that represents these ideas is written. Two main classes form the core architecture of the framework:

- `ModelABC`: An abstract base class that defines the standard interface for all multisensory integration models
- `NDResult`: A result object responsible for storing multi-dimensional stimulus information and providing analysis tools for research

This two-class design separates the model implementation logic from the data handling and analysis capabilities, following standard software engineering principles. The entirety of the classes, along with their interconnections, is presented in a diagram employing UML language (Jacobson et al., 2000) to formally depict the relationships among the core modules of the project (see Fig. 1).

ParameterSweep is a tool for performing parameter sweeps over models, offering high flexibility and employing interchangeable strategies to process results, thereby enabling efficient memory management. NDResultCollection serves as an auxiliary class designed to aggregate and compress results into organized collections. This facilitates advanced functionalities, such as the analysis of spatiotemporal disparity effects. Furthermore, NDResultCollection is the standard output format of ParameterSweep when using the default Processing Strategy. For a complete description of the implemented classes, refer to Appendix B.

Implemented Models

Any model of multisensory integration must define the link between responses to unisensory signals, such as visual and auditory, and responses to cross-modal signals such as visual-auditory. This connection varies according to spatial and temporal characteristics, experimental configuration, and level of description (e.g. single neurons, neural populations, neuroimaging, behaviors). This opens a broad spectrum of approaches for modeling the observations derived from multisensory integration experiments.

A prevalent paradigm in this field is the Ventriloquist Effect (Thurlow and Jack, 1973). This phenomenon arises when incongruent visual and auditory stimuli are simultaneously presented, leading the observer to perceive a singular origin for both visual (the movements of a puppet's face) and auditory (speech) stimuli, attributing them to the same source (the puppet's speaking) (see Fig. 2 for an illustration). This effect is systematically investigated in laboratory environments through tasks designed to assess the spatial localization of an auditory source within a combined visual-auditory setting (Alais and Burr, 2004).

In general, these models aim to explain common "empirical rules" derived from multisensory integration paradigms (see Colonius and Diederich 2020 and Stein et al. 2020 for a detailed review). The "spatio-temporal rule" suggests that stimuli near in space and time are more likely to be integrated. The "inverse-effectiveness rule" states that integration is stronger when the unimodal input intensity decreases. Additionally, the "reliability rule" emphasizes a greater weighting of more reliable modalities (e.g., with lower noise). These empirically observed "rules" have been demonstrated to arise from common computational principles (Ohshiro et al., 2011) and do not necessarily require alternative models for their explanation. In consequence, the key challenge is the development of models capable of accounting for empirical observations of multisensory integration across various levels of description.

Here we present the main models of the Ventriloquist Effect currently available in the Scikit-NeuroMSI package. Appendix C contains a comprehensive mathematical exposition of the models, as well as the code necessary for their execution. The package currently includes models pertinent to additional paradigms, such as the Sound-Induced Flash

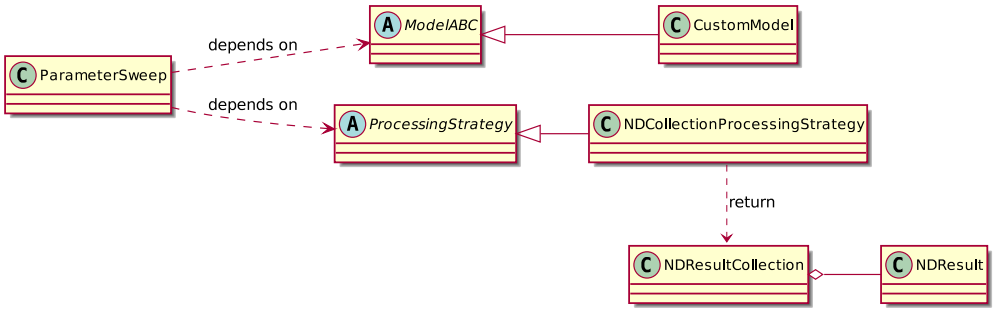

Fig. 1 **Reduced Class Diagram of *Scikit-NeuroMSI***. Empty arrows represent inheritance, the diamond-headed arrow indicates that multiple `NDResult` objects are aggregated into a single `NDResultCollection`, and all other relationships have explanatory labels

Illusion (Cuppini et al., 2014; Zhu et al., 2024b). These models are not elaborated upon in depth in this article, but are thoroughly delineated in the user documentation. Furthermore, we provide guidance on how to streamline the incorporation of novel models into the package. We actively encourage the research community specializing in multisen-

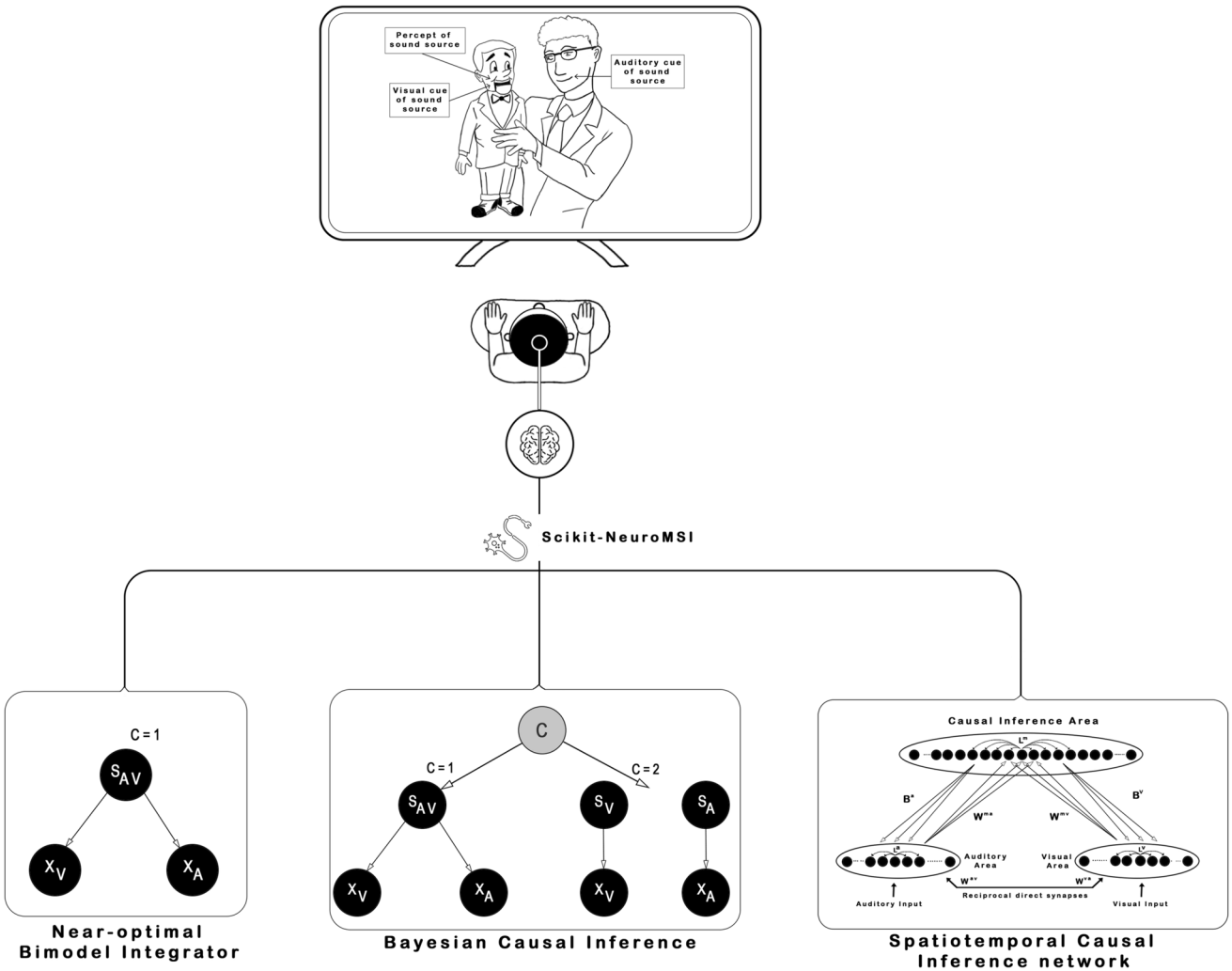

Fig. 2 **Implemented models in *Scikit-NeuroMSI***. The illustration represents how the software package allows to model the Ventriloquist Effect (i.e. spatial integration under audio-visual disparities) using three different approaches: Near-optimal Bimodal Integrator, Bayesian Causal Inference and Spatiotemporal Causal Inference network

sory integration to contribute to this initiative by developing and sharing their own models (refer to Contributing Guidelines).

Near-Optimal Bimodal Integrator

An early model proposes that the process of cue combination from different modalities resembles a maximum-likelihood integrator (Ernst and Banks, 2002). The Near-optimal Bimodal Integrator (Alais and Burr, 2004) for auditory (A) and visual (V) signals in the context of an auditory spatial localization task (e.g. Ventriloquist effect) can be computed by adding the unisensory estimates ($\hat{S}_A$ and $\hat{S}_V$) weighted by their reliability. Consequently, the integrated percept is predisposed to align more closely with the signal that exhibits lower variability.

Bayesian Causal Inference

In the previous model, cue integration is essential for accurately estimating a cross-modal stimulus. However, if there is a significant difference between the subjective assessments $\hat{S}_A$ and $\hat{S}_V$ from a visual-auditory stimulus, the observer cannot determine if this difference is due to random noise in neural signal processing or systematic signal divergence.

Originally designed to explain the Ventriloquist Effect, the Bayesian Causal Inference model (Körding et al., 2007) distinguishes whether $\hat{S}_A$ and $\hat{S}_V$ arise from a single audiovisual event (integration) or separate events (segregation). To do so, the observer considers the likelihood of $\hat{S}_A$ and $\hat{S}_V$ given a common or separate event and the prior probability of a common source. A higher likelihood occurs if the two unisensory signals are similar, which in turn increases the probability of inferring that the signals have a common cause.

Network Model for Audio-Visual Integration and Causal Inference

Bayesian causal inference models provide a high-level description (i.e. computational level of analysis according to Marr (Marr, 2010)) of the computations carried out by the brain to integrate unisensory signals. Recently, neural network models have been proposed as an alternative to model causal inference in multisensory integration paradigms (Cuppini et al., 2017; Fang et al., 2019; Rideaux et al., 2021), providing a low-level description of such a mechanism.

The audio-visual integration and causal inference network (Cuppini et al., 2017) features two unisensory regions for processing noisy auditory and visual stimuli, interconnected by cross-modal excitatory synapses. Here, rate-coded neurons are spatially organized, with closer neurons responding to nearer spatial positions. These regions emulate sensory processing in the brain's unisensory cortex and determine the spatial location of the stimuli by computing the barycenter of activity in the auditory and visual regions. Cross-modal connections cause the spatial localization of one modality to be influenced by the concurrent presentation in another, even if processed separately.

Multisensory Spatiotemporal Causal Inference Network

Our research group developed the Multisensory Spatiotemporal Causal Inference Network to account for the sound-induced flash illusion (Paredes et al., 2025). This model was built upon preceding network models for spatial (Cuppini et al., 2017) and temporal (Cuppini et al., 2014) multisensory integration to inform two levels of causal inference processing. Our model consists of three layers: two encode auditory and visual stimuli separately and connect to a multisensory layer via feedforward and feedback synapses. At the unisensory areas, the model computes the spatiotemporal position of the external stimuli. In addition, at the multisensory area the model computes causal inference. This neural architecture allows iterative computation of spatiotemporal causal inference across the network (Rohe et al., 2019).

In summary, our model retains neural connectivity (lateral, cross-modal, feedforward) and inputs as detailed in the previously discussed network (Cuppini et al., 2017), while integrating feedback connectivity. Additionally, this model introduces latency into the cross-modal and feedforward-feedback neural inputs in accordance with literature indicating that early and late interactions during multisensory processing. Furthermore, in line with Cuppini et al. (2014), temporal filters have been incorporated for auditory, visual, and multisensory neurons to replicate the temporal progression of neural input and synaptic dynamics. These filters account for specific time constants that determine the temporal characteristics of each group of neurons, namely auditory, visual, or multisensory.

Example Applications

Modeling Spatiotemporal Causal Inference with Scikit-NeuroMSI

Causal inference is a highly relevant computation for multisensory integration (Körding et al., 2007; Shams and Beierholm, 2010, 2022). Causal inference in multisensory integration is examined through implicit or explicit tasks. Explicit causal inference involves tasks in which participants are required to directly assess the causal relationship between stimuli (unity judgment) in a multisensory setting, whereas implicit causal inference involves tasks where participants are required to estimate the spatiotemporal location of the stimuli. An in-depth investigation of how the causal mecha-

nism operates in both types of tasks is currently in progress (Acerbi et al., 2018) and has been found to be distinct in neurodiverse populations (e.g. Noel et al. (2022b)).

In general, multisensory causal inference models rely mainly on Bayesian inference (Körding et al., 2007), offering a high-level description (as described by Marr's computational level (Marr, 2010)) of how the brain integrates sensory signals (French and DeAngelis, 2020). Recently, neural network models have emerged as an alternative for modeling causal inference in multisensory contexts (Cuppini et al., 2017; Fang et al., 2019; Rideaux et al., 2021), providing a tentative implementation of the mechanism. Yet, these models have not been rigorously tested across multiple experimental frameworks or dimensions (i.e. space and time), nor have they been compared with other models. In the following, we ask: 1) Are the probabilistic and network models comparable in their performance when fitting data? 2) Do these models accurately account for both implicit and explicit causal inference responses? 3) Do these models show performance differences when working with spatial or temporal disparities?

Modeling Setup

We used the models currently implemented in Scikit-NeuroMSI to reproduce human responses in audio-visual causal inference tasks (see Experiments 2, 3 and 4 in Noel et al. (2022b) for details on the experimental setup) and qualitatively compared their performance. For each task, we fitted the implemented models to behavioral responses of healthy control participants using the differential evolution algorithm (Storn and Price, 1997) available in the SciPy library for the Python programming language (Virtanen et al., 2020). Details about the fitting procedure and model readout for each task can be found in the Appendix D.

First, we model an auditory spatial localization task with disparate visual cues (Experiment 2 in Noel et al. (2022b)) to examine the implicit causal inference performance of the models. Our main focus is the modulation of auditory spatial perception by visual stimuli, a phenomenon known as auditory bias. We present each model with an auditory stimuli at a fixed position (45°) and visual stimuli at different positions relative to the auditory cue: ±3, ±6, ±12, and ±24°(see Fig. 3, left). For simplicity, we present each possible combination of audio-visual stimuli only once, and all noise sources are eliminated for this approach. Following Körding et al. (2007) and Cuppini et al. (2017), we compute the auditory bias as the spatial disparity between the position of the auditory stimulus and the position detected by each model, divided by the distance between the auditory and visual stimuli. We fit the auditory bias responses of the model to correspond with human responses under the

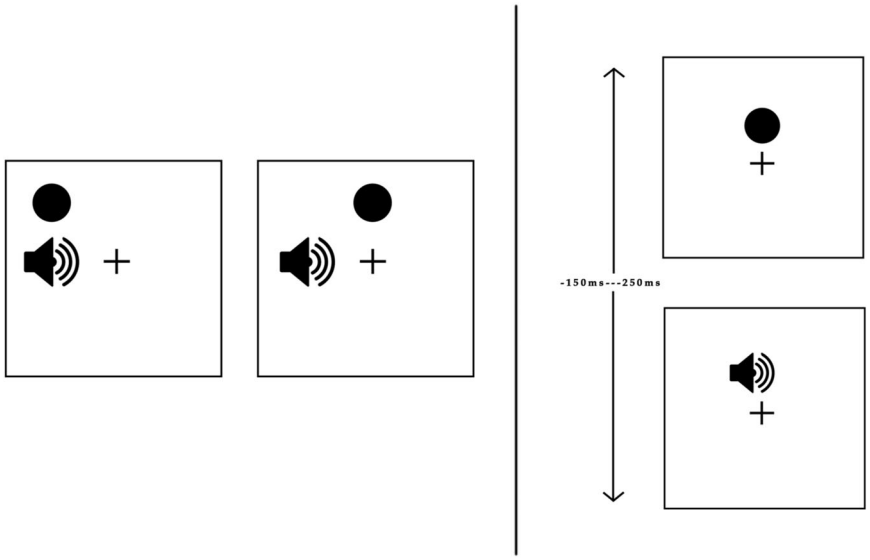

Fig. 3 Causal inference tasks. The figure shows the causal inference tasks reported in Noel et al. (2022b) that were simulated in this study. Left panel: Spatial audio-visual disparity task. Participants viewed a visual disk and heard an auditory tone at different locations and with different small disparities (top = no disparity, bottom = small disparity). We present each model with an auditory stimuli at a fixed position (45°) and visual stimuli at different positions relative to the auditory cue: ±3, ±6, ±12, and ±24°. The models had to determine the position of the auditory stimuli and report the number of causes (1 or 2). Right panel: Temporal audio-visual disparity task. Participants viewed a visual disk and heard an auditory stimuli at different onsets relative to the visual cue. We present each model with a visual stimulus at a fixed onset (160 ms) and auditory stimuli at different onsets relative to the visual cue: 0, ±20, ±80, ±150, and +250 ms. The models had to determine the number of causes of the stimuli (1 or 2)

high visual cue reliability condition (see Noel et al. 2022b for details).

Next, we model an audio-visual common cause report task under spatial disparities (Experiment 3 in Noel et al. (2022b)) to examine the explicit causal inference performance of the models. Audio-visual stimuli are delivered to each model at the same positions as in the previous simulation. Here, our focus is the modulation of the unity judgments (common cause reports) of the models by the spatial disparity of the stimuli. For the Bayesian model, we determine the proportion of reports indicating a common cause by calculating the posterior probability of a common cause (refer to Eq. C7). In contrast, for network models, this proportion is identified through the maximal neural activation (refer to Eq. C9) observed within the multisensory neurons.

Finally, we model an audio-visual common cause report task under temporal disparities (Experiment 4 in Noel et al. (2022b)) to examine the explicit temporal causal inference performance of the models. Here our focus is on how temporal disparities in stimuli affect the unity judgments of the models. We present each model with a visual stimulus at a fixed onset (160 ms) and auditory stimuli at different onsets relative to the visual cue: 0, ±20, ±80, ±150, and +250 ms (see Fig. 3, right). Each possible combination of audio-visual stimuli is presented only once, and all noise sources are eliminated for this approach. We calculate the proportions of the common cause reports of the models as in the previous simulation.

Simulation Results

We compared the performance of the implemented models in the aforementioned causal inference tasks. Auditory bias responses are shown in Fig. 4a. We observe that both Bayesian (Körding et al., 2007) and Network models (Cuppini et al., 2017) provide a good approximation to behavioral data, while the maximum likelihood estimation model (MLE) (Alais and Burr, 2004) fails to reproduce audio-visual disparities beyond ±6°.

Furthermore, spatial causal inference responses are shown in Fig. 4b. We observe that all the evaluated models provide a fair approximation to behavioral responses, with the neural network models (Cuppini et al., 2017) outperforming the Bayesian Causal Inference model (Körding et al., 2007) in audio-visual disparities below ±6°.

Temporal causal inference responses are shown in Fig. 4c. We observe that both the Bayesian Causal Inference model (Körding et al., 2007) and the Multisensory Spatiotemporal Causal Inference Network (Paredes et al., 2025) provide a good approximation to behavioral responses, whereas the network model for audio-visual integration (Cuppini et al., 2017) fails to reproduce temporal disparities beyond ±100 ms.

The findings from the current series of simulations indicate that the Bayesian Causal Inference and Spatiotemporal Causal Inference network models offer the most accurate representation of the participants' data. However, the efficacy of

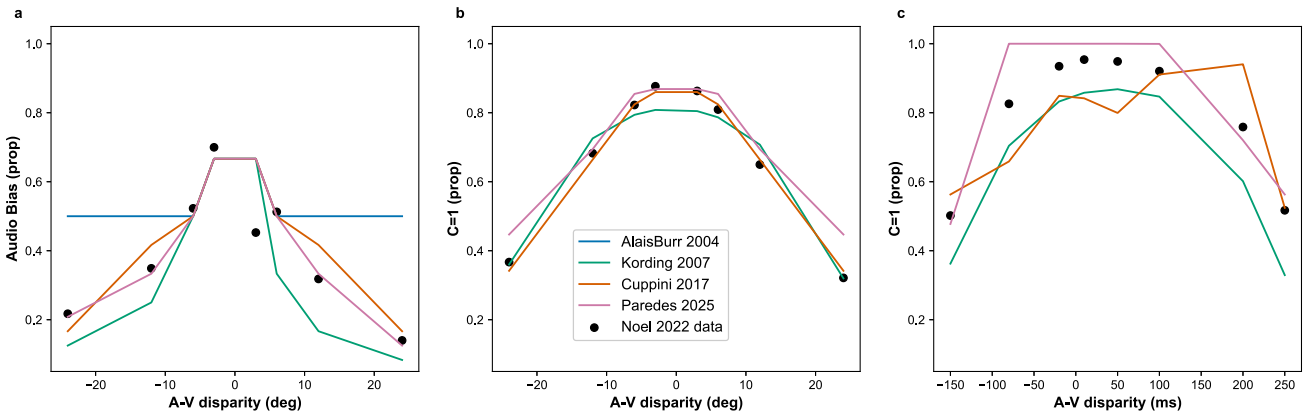

Fig. 4 Models of audio-visual causal inference tasks. The figure shows optimal responses of the implemented models in multisensory causal inference in the audio-visual disparity tasks reported in Noel et al. (2022b). **(a)** Performance of the models in spatial localization within an implicit causal inference task. This graph reveals that Bayesian and network models provide optimal performance, whereas the MLE model fails to reproduce audio-visual disparities beyond ±6°. **(b)** Performance of the models in common source reports under spatial disparities within an explicit causal inference task. These simulations show that the network models outperform the Bayesian Causal Inference model in audio-visual spatial disparities below ±6°. **(c)** Performance of the models in common source reports under temporal disparities within an explicit causal inference task. This graph shows that both the Multisensory Spatiotemporal Causal Inference Network and the Bayesian Causal Inference models provide fair approximations to human performance, whereas the network model for audio-visual integration fails to reproduce disparities beyond ±100 ms

both models diminishes when addressing temporal disparities, underscoring the need for new models that effectively incorporate causal inference within the temporal domain.

Comparing network and Bayesian models of Spatiotemporal Causal Inference with Scikit-NeuroMSI

To our knowledge, this is the first time that implicit and explicit spatiotemporal causal inference is computationally modeled using a combined Bayesian and Neural Network approach (Ursino et al., 2014; Acerbi et al., 2018). This methodology facilitates the concurrent assessment of the influence of each model parameter and the identification of similarities among them. We are now equipped to conduct comprehensive sweeps of parameters within each model and assess their effects on model responses to discern commonalities that may allow us to bridge both levels of description. This process enables the exploration of interindividual variability by combining different modeling approaches, offering new ways to study multisensory integration differences seen in psychiatric or neurological conditions (Martin et al., 2013; Haßet al., 2017; Zvyagintsev et al., 2017; Paredes et al., 2022; Cascio et al., 2012; Stevenson et al., 2014; Hahn et al., 2014; Zhou et al., 2018; Noel et al., 2022a; Wu et al., 2012; Festa et al., 2017; Ramkhalawansingh et al., 2017). In the following, we ask: 1) What would the correlates of the components of Bayesian models be in a more neural implementation? 2) Do the parameters of each model have the same impact on implicit and explicit causal inference responses? 3) Are the effects of model parameters the same for tasks involving spatial or temporal disparities?

Modeling Setup

For the Bayesian Causal Inference model, we explored the impact of varying the prior probability of a common cause (p_{common}) and the precision of the unisensory estimates (σ_a or σ_v). For Spatiotemporal Causal Inference network model, we explored the impact of manipulating the weights of cross-modal ($W_0^{av,va}$), feedforward (W_0^{mc}), feedback (W_0^{cm}) and excitatory lateral ($L_{0_{ex}}^{c}$) synapses. These parameters were selected due to its relevance in explaining individual differences in multisensory integration found in psychiatric conditions, as shown by recent computational research (Karvelis et al., 2018; Noel et al., 2022a, b; Paredes et al., 2022; Chrysaitis and Seriès, 2023; Noel and Angelaki, 2023).

First, we explored the impact of manipulating these parameters on the auditory bias responses of the selected models in the implicit causal inference task (Experiment 2 in Noel et al. (2022b)). For simplicity, we computed the auditory bias at the -6°disparity point for each value of the explored parameters. Next, we examined the effects of sweeping parameters on the proportion of synchronous reports across spatial and temporal disparities (Experiments 3 and 4 in Noel et al. (2022b)). Following Noel et al. (2022b), these differences were systematically quantified by fitting Gaussian functions to the proportion of common source responses as a function of audio-visual disparities (Δ). The Gaussian fits provide three parameters that characterize the responses of the computational models: (1) amplitude, denoting the maximum proportion of common source reports by the model; (2) mean, indicating the Δ at which the proportion of common source reports was maximal; and (3) width (standard deviation), reflecting the extent of Δ within which the model was prone to report a common source.

Simulation Results

The simulated auditory bias responses are shown in Fig. 5a. We found that both p_{common} and σ_v within the Bayesian framework are inversely related to $L_{0_{ex}}^{c}$ and $W_0^{av,va}$ within the network level. In line with previous network modeling of audiovisual integration (Ursino et al., 2017, 2019), our results suggest a possible neural correlate of the prior probability of the co-occurrence of audio-visual stimuli in the cross-modal synapses, with such neural mechanism impacting unisensory precision as well.

In addition, the simulated common source responses in the spatial causal inference task are shown in Fig. 5b. We found that σ_v within the Bayesian model shows an opposite impact in the amplitude and width of the common source reports compared to $W_0^{av,va}$ at the network level. This highlights the impact of cross-modal connectivity in explicit causal inference judgments, suggesting that its observed association with sensory precision estimates potentially scale up towards higher order cortical areas responsible for causal inference computations (Rohe and Noppeney, 2015; Rohe et al., 2019).

The simulated common source responses in the temporal causal inference task are shown in Fig. 5c. In contrast to observations in the spatial domain, we found that the p_{common} within the Bayesian model displays a similar impact in the amplitude and width of the common source reports compared to W_0^{mc} at the network level. This discrepancy opens up questions about potential differences in the mechanisms driving temporal and spatial causal inference, or at the very least, in the foundational assumptions under which these models were initially formulated. Notably, most of the modeling efforts have been carried out in spatial (static) multisensory integration tasks, whereas models of causal inference in the temporal domain at different levels of description have recently begun to accumulate (Cuppini et al., 2014; Pesnot Lerousseau et al., 2022; Zhu et al., 2024b).

Overall, the Bayesian parameter p_{common} representing prior beliefs about common causes could be mapped to neural parameters such as $L_{0_{ex}}^{c}$, $W_0^{av,va}$ and W_0^{mc} representing

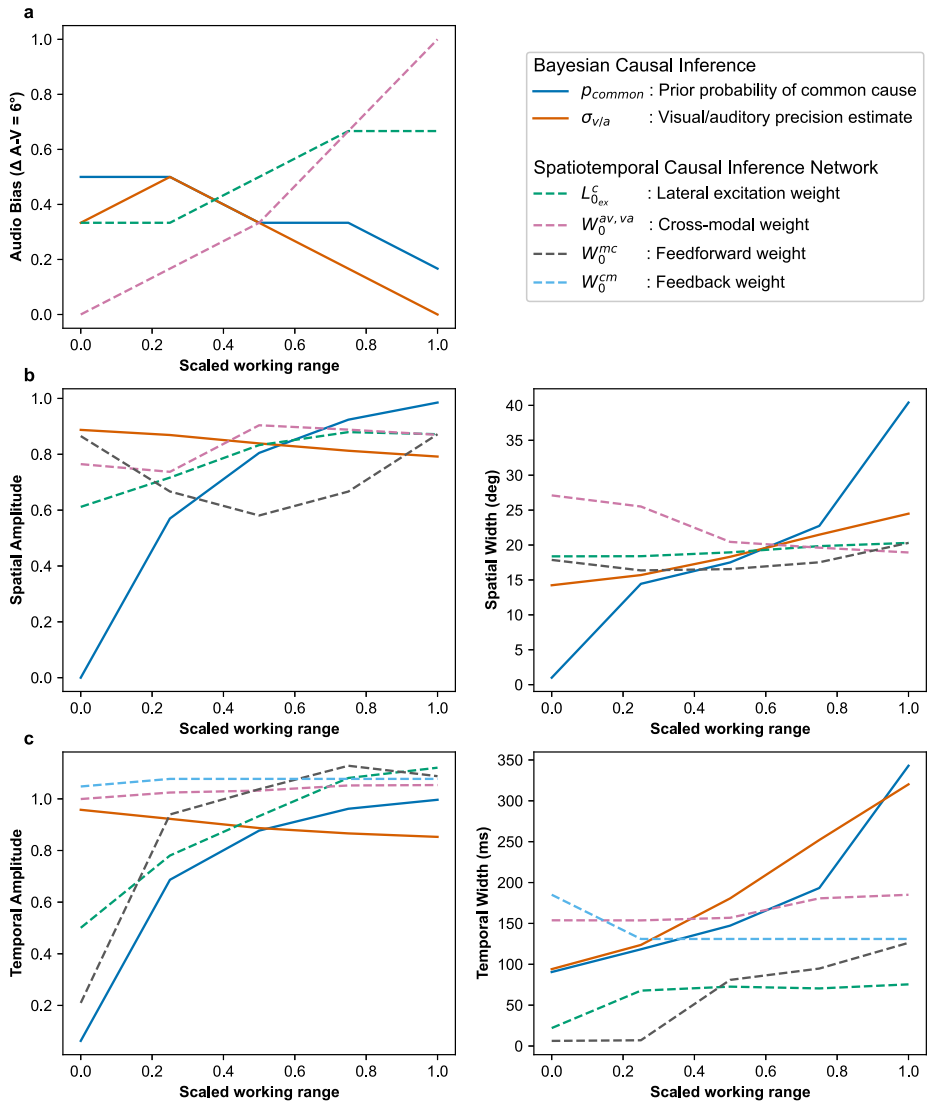

Fig. 5 Impact of model parameters on causal inference responses in Bayesian and network models. **(a)** Parameter sweeps on the implicit causal inference task. The simulations indicate that the prior probability of a common cause (p_{common}) and visual estimate precision (σ_v) reduce auditory bias in the Bayesian model, while lateral excitation ($L^{c}_{0_{ex}}$) and cross-modal weights ($W_0^{av,va}$) enhance it in our network model. **(b)** Parameter sweeps on the explicit spatial causal inference task. The proportion of common source responses relative to spatial disparities fitted to a Gaussian function for analysis. The simulations show that in the Bayesian Causal Inference model the parameter σ_v shows an opposite impact in the amplitude and width of the common source reports compared to $W_0^{av,va}$ in the model. **(c)** Parameter sweeps on the explicit temporal causal inference task. The simulations show that in the Bayesian model p_{common} displays a similar impact in the amplitude and width of the common source reports compared to W_0^{mc} in the network model

inter and intra-areal synaptic strengths, although no exact parallel could be found across domains (spatial or temporal) or metrics (bias, amplitude and width). Similarly, $\sigma_{v/a}$ representing uncertainty in sensory information could be inversely mapped to $W_0^{av,va}$, reflecting the strength of cross-modal connectivity at the network level, due to their opposite impact in the amplitude and width of the common source reports during explicit tasks. The observed discrepancies, including the differing effects of Bayesian and network models on implicit versus explicit causal inference tasks, indicate the presence of additional neural complexities that may not be fully encapsulated by Bayesian modeling or, conversely, by network approaches. The acquired understanding of the similarities of these models opens up the possibility of extending the current theoretical accounts of multisensory integration (Colonius and Diederich, 2020).

Discussion

We have addressed the objective of developing scientific software specifically designed for the computational modeling of multisensory integration, attending a key necessity in the field (Colonius and Diederich, 2020; Shams and Beierholm, 2022). We demonstrated the capabilities of Scikit-NeuroMSI in facilitating the implementation of multisensory integration models and systematically investigating their behavior by sweeping parameters across simulations (see Code 1 and Fig. 4). We have also demonstrated the utility of the software by modeling spatiotemporal causal inference at different levels of analysis using Bayesian (Körding et al., 2007) and network models of multisensory integration (see Fig. 5), addressing a fundamental inquiry necessary for advancing the field (Ursino et al., 2014; French and DeAngelis, 2020; Shams and Beierholm, 2022).

With software tools such as a Scikit-NeuroMSI we are now able to approximate multisensory integration at different levels of analysis (Marr, 2010) (e.g. computational, algorithmic, and neural) simultaneously and extend our possibilities of generating computationally informed hypotheses. This enables the formulation of more precise predictions that can be evaluated with neurobiological and behavioral measurements, a factor crucial for the consolidation of emerging theories of multisensory integration in neuroscience (Colonius and Diederich, 2020; Shams and Beierholm, 2022).

An immediate application for our new modeling framework is the study of multisensory integration differences in psychiatric and neurological disorders (Martin et al., 2013; Haßet al., 2017; Zvyagintsev et al., 2017; Paredes et al., 2022; Cascio et al., 2012; Stevenson et al., 2014; Hahn et al., 2014; Zhou et al., 2018; Noel et al., 2022a; Wu et al., 2012; Festa et al., 2017; Ramkhalawansingh et al., 2017). For example, a novel quantitative theory on ASD (Noel and Angelaki, 2023) suggests that ASD could be interpreted as a multisensory causal inference disorder (computational level), where this process may be facilitated by divisive normalization (algorithmic level) and potentially disrupted by excitatory/inhibitory imbalances (neural implementation level). However, there is as yet no formal evaluation of experimental data of this disorder using more than one multisensory integration model at the time to bridge across levels of analysis.

We acknowledge that our modeling effort represents a first step towards achieving a general solution for multisensory integration formalization. We have shown the capabilities of our software in the simulation of multiple models in a group of three similar tasks (Noel et al., 2022b). However, our software framework requires the incorporation of model comparison and validation metrics to facilitate the critical assessment of each model implementation (Wilson and Collins, 2019; Blohm et al., 2020). Overall, we propose a software environment as a first approach to a generalized framework for multisensory integration, needed for the theoretical advancement of the field.

Availability and Future Directions

The entire source code is under a BSD 3-Clause License and available in a public repository: https://github.com/renatoparedes/scikit-neuromsi. Scikit-NeuroMSI is available for installation on the Python Package-Index (PyPI)[1]. User documentation is automatically generated from Scikit-NeuroMSI docstrings and published in the Read the Docs service[2].

In Spanish, there is a phrase "*Con el diario del Lunes*" (literally, "With Monday's newspaper"), which shares the same meaning as the English expression "Monday-morning quarterback" - indicating that something becomes obvious only after it has happened. While Scikit-NeuroMSI has successfully achieved its technical objective of standardizing existing multisensory integration models, our experience has revealed opportunities for improved computational modeling.

Specifically, the architecture could be enhanced by decomposing the models into two distinct entities:

- A stimulus processing component that handles individual sensory inputs
- An integration component that consolidates the results into a unified modality

Or mathematically speaking:

$$E : S \times I \rightarrow R \tag{1}$$

Where:

- S represents the stimulus source(s) from one or multiple modalities
- I is the integrator that combines the stimuli from S

We envision a Python implementation similar to what is presented in Code 6. By decoupling stimulus processing/generation from integration mechanisms, new integration models can be easily implemented and tested without modifying the underlying stimulus code. The flexible architecture simplifies the implementation of sophisticated integration strategies and enables straightforward extension to handle additional modalities or stimulus types, making the framework particularly valuable for emerging research in areas such as brain-computer interfaces and robotics. From a software perspective, this new design would promote code

[1] https://pypi.org/project/scikit-neuromsi/

[2] https://scikit-neuromsi.readthedocs.io/

reusability and make the codebase more maintainable, allowing the scientific community to contribute new models and extensions to the framework more easily.

Code 6 Proposed implementation using composition-based design: The same integration component (`integrators.SomeIntegrator`) can process both cross-modal (audio-visual) and unimodal (audio-audio) combinations through a flexible stimulus composition interface. This design separates stimulus handling from integration logic, enabling more modular and extensible implementations.

```
from skneuromsi import stimulus, integrators

# one sound and one visual stimulus
m0 = Model(
    stimulus.Auditory(...), stimulus.Visual(...),
    integrators.SomeIntegrator(...),
)

# two sounds
m1 = Model(
    stimulus.Auditory(...), stimulus.Auditory(...),
    integrators.SomeIntegrator(...),
)
```

Information Sharing Statement

The code and data used to generate the simulations presented in this article is available at: https://github.com/renatoparedes/NeuroMSI-Network.

Appendix A Formal Model Standardization

Formally, our aim is to define a framework with the following property:

$$E : \mathcal{M} \to \mathcal{R} \tag{A1}$$

Where:

- E is the execution of a model m.
- $\mathcal{M}$ is the set of all existing models.
- $\mathcal{R}$ is the space of all possible results of $\mathcal{M}$.

Then for a specific model $m \in \mathcal{M}$

$$r = E(m)$$

where $r \in \mathcal{R}_m \subseteq \mathcal{R}$.

Then we define $f \in \mathcal{F}$ where $\mathcal{F}$ represents all result processing tools, and we define a processing function:

$$f : \mathcal{R} \to \mathcal{Y}$$

where $\mathcal{Y}$ is the output space of the tool; and with this we define the universal property

$$\forall r \in \mathcal{R}, \exists y \in \mathcal{Y} : y = f(r), \forall f \in \mathcal{F} \tag{A2}$$

Appendix B Classes and Objects

The abstract class `ModelABC` defines the functionalities responsible for configuring mode names and establishing a common protocol for all models defined in the package. In terms defined in the previous equations, any model that inherits from `ModelABC` belongs to the set M of the Eq. A1.

The class `NDResult` is the implementation of any result from models inherited from `ModelABC` and from a mathematical point of view, they are the code realization of the set R from Eq. A1. As a data structure, `NDResult` (N-Dimensional Result) is a thin layer that adds multisensory integration analysis capabilities on top of the `DataArray` structure from the *XArray* library (Hoyer and Hamman, 2017). *XArray* was originally designed to work with high-dimensional data from NetCDF format (Rew and Davis, 1990), commonly used in satellite data processing. Additionally, since *XArray* provides persistence facilities, a `.ndr` persistence format was defined, which compresses a *NetCDF* file along with its metadata in a ZIP container (PKWARE Inc., 2022). Furthermore, `NDResult` also supports compression, as model results typically consume significant memory resources.

The `ParameterSweep` class is the primary tool designed to perform parameter sweeps over a model, automatically executing model multiple times while systematically varying the value of a target parameter within a specified range. The class allows configuration of the number of repetitions for each parameter value, supports parallel execution using multicore architectures, and implements memory management to prevent overflow conditions. Notably, `ParameterSweep` is model-agnostic and can work with any model that inherits from `ModelABC`, making it highly flexible and reusable. This is the first use of DIP.

The second use case implements the Strategy Pattern (Gamma et al., 1995), allowing interchangeable algorithms to process the results of each model execution. By default, `ParameterSweep()` employs a strategy called `NDCollectionProcessingStrategy`, which acts as a processing pipeline that first compresses each individual result and then aggregates them into a coherent data structure called `NDResultCollection`, facilitating subsequent analysis. This architectural decision was driven by the memory footprint of the results, which typically occupy hundreds of megabytes, making concurrent storage of multiple results computationally expensive. The design allows for the implementation of custom strategies that can selectively store specific data points from each result, thus optimising memory utilisation.

`ParameterSweep` supports any model that instantiates from a class inheriting from `ModelABC`, and knows how to process the `NDResult` objects it generates by delegating

them to the chosen strategy. In other words, it is the code implementation of the property defined in Eq. A2.

`NDResultCollection` is the final piece of the puzzle that makes up Scikit-NeuroMSI. It is an auxiliary class that allows grouping many compressed results from the same model into a single collection and extends them with classic area functionalities such as cross-modal bias and causal analysis (Körding et al., 2007). Additionally, it is the default data type returned by the `ParameterSweep()` tool when used with the default strategy. Similar to `NDResult`, it offers the capability to export each data point to NetCDF format in a ZIP compressed file with its metadata in a format we call `*.ndc`.

Appendix C Implemented Models

C.1 Near-Optimal Bimodal Integrator

The Near-optimal Bimodal Integrator (Alais and Burr, 2004) for auditory (A) and visual (V) signals in the context of an auditory spatial localization task can be computed as:

$$\hat{S}_{AV} = w_A \hat{S}_A + w_V \hat{S}_V \tag{C3}$$

where $\hat{S}_A$ and $\hat{S}_V$ are unimodal auditory and visual estimates, respectively, and $\hat{S}_{AV}$ is the multimodal estimate.

In addition, w_A and w_V are the relative weights for each modality, defined as:

$$w_A = \frac{\sigma_V^2}{\sigma_A^2 + \sigma_V^2} \tag{C4}$$

$$w_V = \frac{\sigma_A^2}{\sigma_V^2 + \sigma_A^2} \tag{C5}$$

where σ_A and σ_V are the variances of each unimodal stimuli, respectively.

These equations show that the optimal multisensory estimate adds the unisensory estimates weighted by their normalized reciprocal variances.

The Near-optimal Bimodal Integrator model can be called using Code 7.

Code 7 Example of Near-optimal Bimodal Integrator instantiation and default execution parameterisation within the Scikit-NeuroMSI package.

```
# import the model
>>> from skneuromsi.mle import AlaisBurr2004

# create the model object
>>> model = AlaisBurr2004(position_range=(0, 90), position_res=1)

# explore the model run default parameterisation
>>> model.run
<function skneuromsi.mle._alais_burr2004.AlaisBurr2004.run(*,
↪  auditory_position=-5, visual_position=5, auditory_sigma=3.0,
↪  visual_sigma=3.0, noise=None)>
```

C.2 Bayesian Causal Inference

The Bayesian Causal Inference model (Körding et al., 2007) uses the following formulation:

$$p\left(C \mid x_1, x_2\right) = \frac{p\left(x_1, x_2 \mid C\right) p\left(C\right)}{p\left(x_1, x_2\right)} \tag{C6}$$

where x_1 and x_2 are two unimodal signals and C is a binary variable that represents the number of causes in the environment.

The posterior probability of the signals having a single cause in the environment is defined as follows:

$$p\left(C = 1 \mid x_1, x_2\right) = \frac{p\left(x_1, x_2 \mid C = 1\right) p\left(C = 1\right)}{p\left(x_1, x_2 \mid C = 1\right) p\left(C = 1\right) + p\left(x_1, x_2 \mid C = 2\right)\left(1 - p\left(C = 1\right)\right)} \tag{C7}$$

and the likelihood is computed as:

$$p\left(x_1, x_2 \mid C = 1\right) = \iint p\left(x_1, x_2 \mid X\right) p\left(X\right) dX \tag{C8}$$

Here $p\left(C = 1\right)$ is the prior probability of a common cause (by default 0.5). X denotes the attributes of the stimuli (e.g. distance), which are then represented in the nervous system as x_1 and x_2.

These equations show that the inference of a common cause of two unisensory signals is computed by combining the likelihood and prior of signals having a common cause. A higher likelihood occurs if the two unisensory signals are similar, which in turn increases the probability of inferring that the signals have a common cause.

The Bayesian Causal Inference model can be called using Code 8.

Code 8 Example of Bayesian Causal Inference model instantiation and default execution parameterisation within in the Scikit-NeuroMSI package.

```
# import the model
>>> from skneuromsi.bayesian import Kording2007

# create the model object
>>> model = Kording2007(position_range=(0, 90), position_res=1)

# explore the model run default parameterisation
>>> model.run
<function skneuromsi.bayesian._kording2007.Kording2007.run(*,
↪  auditory_position=-15, visual_position=15, auditory_sigma=2.0,
↪  visual_sigma=10.0, p_common=0.5, prior_sigma=20.0, prior_mu=0,
↪  strategy='averaging', noise=True, causes_kind='count',
↪  dimension='space')>
```

C.3 Audio-Visual Integration and Causal Inference Network

The audio-visual integration and causal inference network (Cuppini et al., 2017) consists of three layers: two encode

auditory and visual stimuli, separately, and connect to a multisensory layer where causal inference is computed. Each of these layers consists of 180 neurons arranged topologically to encode a 180°space. In this way, each neuron encodes 1°of space and neurons close to each other encode close spatial positions.

Each neuron will be indicated with a superscript c indicating a specific cortical area (a, v or m for the auditory, visual or multisensory area, respectively). Similarly, each neuron will have a subscript j referring to its spatial position within a given area. Neurons in each layer have a sigmoid activation function and first-order dynamics:

$$\tau^c \frac{dy_j^c(t)}{dt} = -y_j^c(t) + F\left(u_j^c(t)\right), \quad c = a, v, m \tag{C9}$$

Here, $u(t)$ and $y(t)$ are used to represent the net input and output of a given neuron at time t. τ^c denotes the time constant of neurons belonging to a given area c. $F(u)$ represents the sigmoidal relationship:

$$F\left(u_j^c\right) = \frac{1}{1 + exp^{-s\left(u_j^c - \theta\right)}} \tag{C10}$$

Here, s and θ denote the slope and the central position of the sigmoidal relationship, respectively. Neurons in all regions differ only in their time constants, chosen to mimic faster sensory processing for stimuli in the auditory region compared to visual stimuli.

These neurons are recurrently connected in a "Mexican hat" pattern within each layer. Such connectivity pattern consists of defining a central excitatory area surrounded by an inhibitory ring for each neuron, so that the entire layer generates excitation for spatially close stimuli and inhibition for distant stimuli:

$$L_{jk}^s = \begin{cases} L_{ex}^c \cdot exp\left(-\frac{(D_{jk})^2}{2\cdot(\sigma_{ex}^c)^2}\right) - L_{in}^c \cdot exp\left(-\frac{(D_{jk})^2}{2\cdot(\sigma_{in}^c)^2}\right), & D_{jk} \neq 0 \\ 0, & D_{jk} = 0 \end{cases} \tag{C11}$$

Here, L_{jk}^c denotes the weight of the synapse from the pre-synaptic neuron at position k to post-synaptic neuron at position j. D_{jk} indicate the distance between the pre-synaptic neuron and the post-synaptic neurons within a given area:

$$D_{jk} = \begin{cases} |j-k|, & |j-k| \leqslant N/2 \\ N - |j-k|, & |j-k| > N/2 \end{cases} \tag{C12}$$

This defines a circular structure where each neuron receives the same number of lateral connections.

On the other hand, neurons in each unisensory layer (e.g. auditory) are reciprocally connected with neurons in the opposite layer (e.g. visual). These connections are excitatory and modify the spatial perception of unisensory stimuli. These synaptic weights are symmetrically defined ($W_0^{av} = W_0^{va}$ and $\sigma^{av} = \sigma^{va}$) by the Gaussian function:

$$W_{jk}^{cd} = W_0^{cd} \cdot exp\left(-\frac{\left(D_{jk}\right)^2}{2\left(\sigma^{cd}\right)^2}\right), cd = av \; or \; va \tag{C13}$$

W_0 denotes the highest level of synaptic efficacy and D_{jk} is the distance between neuron at position j in the post-synaptic unisensory region and the neuron at position k in the pre-synaptic unisensory region. σ^{cd} defines the width of the cross-modal synapses.

In addition, neurons in the unisensory layers have excitatory connections to the multisensory layer. These synapses are used to encode information about the mutual spatial coincidence of cross-modal stimuli and the probability that two stimuli were generated by a common source (i.e. they solve the problem of causal inference). The weights of these feedforward synapses are symmetrically defined as:

$$W_{jk}^{mc} = W_0^{mc} \cdot exp\left(-\frac{\left(D_{jk}\right)^2}{2\left(\sigma^{mc}\right)^2}\right), c = a, v \tag{C14}$$

Here W_0^{mc} denotes the highest value of synaptic efficacy, D_{jk} the distance between the multisensory neuron at position j and the unisensory neuron at position k, and σ^{mc} the width of the feedforward synapses. The distance is defined as:

$$i_j^m(t) = \sum_{k=1}^{N} W_{jk}^{ma} \cdot y_k^a\left(t - \Delta t_{feed}\right) + W_{jk}^{mv} \cdot y_k^v\left(t - \Delta t_{feed}\right) \tag{C15}$$

Here, Δt_{feed} represents the latency of feedforward inputs between the unisensory and multisensory regions. W_{jk}^{ma} and W_{jk}^{mv} are the synapses connecting the pre-synaptic neuron at position k in a given unisensory area and the post-synaptic neuron at position j in the multisensory area.

Finally, the visual and auditory stimuli used as input to the network are defined with a Gaussian function to mimic the spatially localized external stimuli filtered by the receptive fields of the neurons. The stimulus from the external world is simulated as a 1-D Gaussian function to represent the uncertainty in the detection of external stimuli:

$$e_j^c(t) = E_0^c \cdot exp\left(-\frac{\left(d_j^c\right)^2}{2\left(\sigma^c\right)^2}\right) \tag{C16}$$

Here, E_0^c denotes the strength of the stimulus, d_j^c the distance between the neuron at position j and the stimulus at

position p^c, and σ^c the degree of uncertainty in sensory detection. The distance d_j^c is defined as:

$$d_j^c = \begin{cases} |j - p^c|, & |j - p^c| \leqslant N/2 \\ N - |j - p^c|, & |j - p^c| > N/2 \end{cases} \quad \text{(C17)}$$

The central point of the Gaussian function corresponds to the point of application of the stimulus in the external world, while the standard deviation of the Gaussian function reflects the width of the receptive fields of the neurons and the reliability of the external input. This parameter is used to represent the different spatial acuities of the auditory and visual sensory modalities.

The net input of a neuron is the sum of an inside (i.e. within region) component (l_j^c) and an outside (i.e. extra-area) component ($o_j^c(t)$):

$$u_j^c(t) = l_j^c(t) + o_j^c(t) \quad \text{(C18)}$$

The within region component l_j^c is defined as:

$$l_j^c(t) = \sum_k L_{jk}^c \cdot y_{jk}^c(t) \quad \text{(C19)}$$

Here L_{jk}^c represents the strength of the lateral synapse from a presynaptic neuron at position k to a postsynaptic neuron at position j in the region c. y_{jk}^c is the activity of the presynaptic neuron at position k.

Importantly, the extra-area input is defined differently for unisensory and multisensory areas. The extra-area input for the unisensory areas includes a stimulus from the external world ($e_j^c(t)$), a cross-modal component coming from the other unisensory area ($c_j^c(t)$) and a noise component (n_j^c). Furthermore, the cross-modal input is defined as:

$$\begin{aligned} c_j^a(t) &= \textstyle\sum_{k=1}^N W_{jk}^{av} \cdot y_k^v \\ c_j^v(t) &= \textstyle\sum_{k=1}^N W_{jk}^{va} \cdot y_k^a \end{aligned} \quad \text{(C20)}$$

The noise component n_j^c is extracted from a standard uniform distribution in the interval $[n_{max} + n_{max}]$. Here, n_{max} is defined as the 40% of the strength of the external stimulus for each modality.

The Audio-visual Integration Network model can be called using Code 9:

Code 9 Example of the Audio-visual Integration Network model instantiation and default execution parameterisation within in the Scikit-NeuroMSI package.

```
# import the model
>>> from skneuromsi.neural import Cuppini2017

# create the model object
>>> model = Cuppini2017(neurons=90, position_range=(0, 90),
↪  position_res=1)

# explore the model run default parameterisation
>>> model.run
<function skneuromsi.neural._cuppini2017.Cuppini2017.run(*,
↪  auditory_position=None, visual_position=None, auditory_sigma=32,
↪  visual_sigma=4, auditory_intensity=28, visual_intensity=27,
↪  auditory_duration=None, auditory_onset=0, auditory_stim_n=1,
↪  visual_duration=None, visual_onset=0, visual_stim_n=1,
↪  auditory_soa=None, visual_soa=None, noise=False, noise_level=0.4,
↪  feedforward_weight=18, cross_modal_weight=1.4,
↪  causes_kind='count', causes_dim='space',
↪  causes_peak_threshold=0.15, causes_peak_distance=None)>
```

C.4 Multisensory Spatiotemporal Causal Inference Network

The model consists of three layers: two encode auditory and visual stimuli separately and connect to a multisensory layer via feedforward and feedback synapses. At the unisensory areas, the model computes the spatiotemporal position of the external stimuli. In addition, at the multisensory area the model computes causal inference.

This model maintains the neural connectivity (lateral, crossmodal, feedforward) and inputs described in the network presented in the previous section (Cuppini et al., 2017). Our model now includes feedback connectivity ($B_0^{am} = B_0^{vm}$ and $\sigma^{am} = \sigma^{vm}$), defined by the following equation:

$$B_{jk}^{cm} = B_0^{cm} \cdot exp\left(-\frac{(D_{jk})^2}{2(\sigma^{cm})^2}\right), cm = am \; or \; vm \quad \text{(C21)}$$

B_0^{cm} denotes the highest level of synaptic efficacy and D_{jk} is the distance between neuron at position j in the post-synaptic unisensory region and the neuron at position k in the pre-synaptic multisensory region. σ^{cd} defines the width of the feedback synapses.

Overall, the feedback input is defined as:

$$b_j^c(t) = \sum_{k=1}^N B_{jk}^{cm} \cdot y_k^c\left(t - \Delta t_{feed}\right) \quad \text{(C22)}$$

Here Δt_{feed} represents the latency of feedback inputs between the multisensory and unisensory regions. The feedback synaptic weights are also symmetrically ($B_0^{am} = B_0^{vm}$ and $\sigma^{am} = \sigma^{vm}$) defined:

All these external sources are filtered by a second order differential equation to mimic the temporal dynamics of the

stimuli in a cortex:

$$\begin{cases} \frac{d}{dt} o_j^c(t) = \delta_j^c(t) \\ \frac{d}{dt} \delta_j^c(t) = \frac{G^c}{\tau^c} \cdot \left[e_j^c(t) + c_j^c(t) + b_j^c(t) + n_j^c \right] - \frac{2 \cdot \delta_j^c(t)}{\tau^c} - \frac{o_j^c(t)}{(\tau^c)^2} \end{cases}, c = a, v \tag{C23}$$

Here, G^c represents gain and τ^c the time constants of the dynamics.

These external sources are also filtered by a second order differential equation:

$$\begin{cases} \frac{d}{dt} o_j^m(t) = \delta_j^m(t) \\ \frac{d}{dt} \delta_j^m(t) = \frac{G^m}{\tau^m} \cdot \left[i_j^m(t) \right] - \frac{2 \cdot \delta_j^m(t)}{\tau^m} - \frac{o_j^m(t)}{(\tau^m)^2} \end{cases} \tag{C24}$$

Here, G^m represents gain and τ^m the time constants of the dynamics in the multisensory neurons.

Furthermore, the cross-modal input is defined as:

$$\begin{aligned} c_j^a(t) &= \sum_{k=1}^{N} W_{jk}^{av} \cdot y_k^v(t - \Delta t_{cross}) \\ c_j^v(t) &= \sum_{k=1}^{N} W_{jk}^{va} \cdot y_k^a(t - \Delta t_{cross}) \end{aligned} \tag{C25}$$

Here, Δt_{cross} represents the latency of cross-modal inputs between two unisensory regions.

The Multisensory Spatiotemporal Causal Inference Network model can be called using Code 10.

Code 10 Example of the Multisensory Spatiotemporal Causal Inference Network model instantiation and default execution parameterisation within in the Scikit-NeuroMSI package.

```
# import the model
>>> from skneuromsi.neural import Paredes2025

# create the model object
>>> model = Paredes2025(neurons=90,
                        position_range=(0, 90),
                        position_res=1,
                        time_range=(0, 150),
                        time_res = 1)

# explore the model run default parameterisation
>>> model.run
<function skneuromsi.neural._paredes2025.Paredes2025.run(*,
↪  auditory_soa=50, visual_soa=None, auditory_onset=16,
↪  visual_onset=16, auditory_duration=7, visual_duration=12,
↪  auditory_position=None, visual_position=None,
↪  auditory_intensity=2.4, visual_intensity=1.4, auditory_sigma=32,
↪  visual_sigma=4, noise=False, noise_level=0.4,
↪  temporal_noise=False, temporal_noise_scale=5,
↪  lateral_excitation=2, lateral_inhibition=1.8,
↪  cross_modal_weight=0.075, cross_modal_latency=16, feed_latency=95,
↪  feedback_weight=0.1, feedforward_weight=1.4, auditory_gain=None,
↪  visual_gain=None, multisensory_gain=None, auditory_stim_n=2,
↪  visual_stim_n=1, feedforward_pruning_threshold=0,
↪  cross_modal_pruning_threshold=0, causes_kind='count',
↪  causes_dim='space', causes_peak_threshold=0.8,
↪  causes_peak_distance=None)>
```

Appendix D Model Fitting

D.1 Model Readouts

D.1.1 Implicit Causal Inference Task

The audio-visual spatial localization task (Noel et al., 2022b) involves participants determining whether an auditory stimulus is perceived to the right or left of a central position, indicated by a button press. Subsequently, psychometric functions are constructed by plotting the proportion of responses directed to the right as a function of the stimulus position. These data are modeled using a cumulative Gaussian function. The psychometric function provides two parameters that delineate the localization performance of the participants: bias and threshold. Bias is defined as the stimulus value at which responses are equally divided between rightward and leftward. A bias approximating 0°denotes highly accurate localization. The threshold is represented by the standard deviation of the fitted cumulative Gaussian function. A lower threshold indicates a higher precision in spatial localization.

In order to streamline the process and acknowledging the absence of noise in our simulations, we chose not to directly simulate the participants' left-right responses. Instead, we concentrated on modeling the auditory position estimate for each model. The auditory position estimate for the network model was ascertained by pinpointing the neuron displaying the highest activity level during the simulation, with each neuron corresponding to a discrete spatial segment.

D.1.2 Explicit Causal Inference Tasks

The explicit causal inference tasks require participants to ascertain whether auditory and visual stimuli originate from the same source. In the context of neural network models, this proportion is derived based on the maximal neural activation manifested within the multisensory neurons, contingent upon the condition that such activation exceeds the threshold of 0.15. In the spatial task, the evaluation of multisensory activity is conducted within the spatial domain at the concluding time point, while in the temporal task, the assessment occurs within the temporal domain at the locus of maximal activity. In cases where multiple peak values are identified, the average product of all potential combinations of these peak values is calculated to estimate the proportion attributable to the presence of multiple sources. Subsequently, the complementary proportion, representing the perception of multiple stimuli, is calculated to ascertain the proportion attributable to common source reports.

D.2 Fitting Procedure

The proportion of auditory bias or common cause reports reported by each model was fitted to the experimental data (Noel et al., 2022b) with the cost function defined by Eq. D26.

$$Cost = \sum_{i=1}^{N} \left(\frac{P_i^{data} - P_i^{model}}{P_i^{data}} \right)^2 \quad \text{(D26)}$$

Here, P_i^{data} and P_i^{model} denote the proportion measured in the ith audio-visual stimuli disparity. N represents the number of disparities measured (i.e. 8 in the empirical study for the spatial tasks). This cost function was minimised by the implementation of the differential evolution algorithm (Storn and Price, 1997) available in the SciPy library for the Python programming language (Virtanen et al., 2020).

For the near-optimal bimodal integrator (Alais and Burr, 2004), parameters σ_a and σ_v were fitted under the constraint that parameter σ_a exceeds parameter σ_v, thereby enhancing precision within the visual modality. The algorithm was provided with boundaries set at (0.1, 48) for both parameters.

For the Bayesian causal inference model (Körding et al., 2007), parameters σ_a, σ_v, p_μ and p_σ were also fitted under the constraint that parameter σ_a exceeds parameter σ_v. For spatial tasks, the algorithm was provided with boundaries set at (0.1, 48) for all parameters, except p_μ which was set at (21, 69). For the temporal task, the algorithm was provided with boundaries set at (1, 500) for all parameters.

For spatial tasks, the audio-visual integrator and causal inference network (Cuppini et al., 2017), parameters σ^a, σ^v, E^a and E^v, representing stimulus uncertainty and intensity, respectively, were fitted under the constraint that auditory uncertainty is larger than visual. The algorithm was provided with boundaries set at (0.1, 48) for uncertainties and (0.1, 30) for intensities. For the temporal task, the parameters τ^a, τ^v, τ^m, W_0^{mc}, and $W_0^{av,va}$ were fitted with boundaries set at (0.1, 55), (0.1, 55), (0.1, 75), (0.01, 150), and (0.01, 24), respectively. In the course of these simulations, the parameters σ^a, σ^v, E^a, and E^v were consistently maintained at values of 32, 4, 50, and 49, respectively, with the duration of stimuli being fixed at 6 ms.

For the spatial tasks, the Multisensory Spatiotemporal Causal Inference network parameters σ^a, σ^v, E^a and E^v with the same boundaries and constraints as the previous model. For the temporal task, the parameters τ^a, τ^v, τ^m, W_0^{mc}, $W_0^{av,va}$ and W_0^{cm} were fitted with boundaries set at (15, 50), (15, 50), (0.005, 50), (0, 7.5), (0, 0.15), (0, 0.005) respectively. During these simulations, the parameters σ^a, σ^v, E^a, and E^v were consistently maintained at values of 32, 4, 2.55, and 2.5, respectively, with the duration of stimuli being fixed at 6 ms. Parameters Δt_{cross} and Δt_{feed} were assumed to be constant at 16 ms and 24 ms, respectively.

We acknowledge that employing a standardized fitting routine based on a genetic algorithm for noise-free model simulations may not constitute the optimal choice across all models. This approach may be particularly suboptimal for the Bayesian Causal Inference model, which was originally implemented utilizing a maximum-a-posteriori estimator with the assumption of random motor noise over 10,000 simulations (Körding et al., 2007).

Acknowledgements The authors sincerely thank Jean-Paul Noel for providing the behavioral data modeled in this study and for providing insightful feedback on the preparation of this manuscript. The authors also express gratitude to Francisco Morote for contributing the illustrations included in the manuscript.

Author Contributions RP: Conceptualization, Methodology, Software, Visualization, Formal Analysis, Writing - original draft. JBC: Conceptualization, Methodology, Software, Visualization, Writing - original draft. PS: Supervision, Writing - reviewing and editing.

Funding Open access funding provided by Pontificia Universidad Catolica del Peru. No funding was received for conducting this study.

Data Availability The code and data used to generate the simulations presented in this article is available at: https://github.com/renatoparedes/NeuroMSI-Network

Declarations

Ethical Approval Not applicable.

Competing Interests The authors declare no competing interests.

References

Acerbi, L., Dokka, K., Angelaki, D. E., & Ma, W. J. (2018). Bayesian comparison of explicit and implicit causal inference strategies in multisensory heading perception. *PLoS Computational Biology, 14*(7), Article e1006110. https://doi.org/10.1371/journal.pcbi.1006110

Alais, D., & Burr, D. (2004). The Ventriloquist Effect Results from Near-Optimal Bimodal Integration. *Current Biology, 14*(3), 257–262. https://doi.org/10.1016/j.cub.2004.01.029

Blohm, G., Kording, K. P., & Schrater, P. R. (2020). *A How-to-Model Guide for Neuroscience. eNeuro, 7*(1), 1–12. https://doi.org/10.1523/ENEURO.0352-19.2019

Bolognini, N., Rasi, F., Coccia, M., Làdavas, E. (2005). Visual search improvement in hemianopic patients after audio-visual stimulation. *Brain: A Journal of Neurology*, *128*(Pt 12), 2830–2842. https://doi.org/10.1093/brain/awh656

Booch, G. (1982). *Object-oriented design. ACM SIGAda Ada Letters, 1*(3), 64–76. https://doi.org/10.1145/989791.989795

Brooks, F., & Kugler, H. (1987). *No silver bullet*. April.

Cascio, C.J., Foss-Feig, J.H., Burnette, C.P., Heacock, J.L., Cosby, A.A. (2012). The rubber hand illusion in children with autism spectrum disorders: delayed influence of combined tactile and visual input on proprioception. *Autism: The International Journal of Research and Practice*, *16*(4), 406–419. https://doi.org/10.1177/1362361311430404

Chandrasekaran, C. (2017). Computational principles and models of multisensory integration. *Current Opinion in Neurobiology, 43*, 25–34. https://doi.org/10.1016/j.conb.2016.11.002

Chrysaitis, N. A., & Seriès, P. (2023). 10 years of bayesian theories of autism: a comprehensive review. *Neuroscience & Biobehavioral Reviews, 145*, Article 105022. https://doi.org/10.1016/j.neubiorev.2022.105022

Colonius, H., & Diederich, A. (2004). Multisensory Interaction in Saccadic Reaction Time: A Time-Window-of-Integration Model. *Journal of Cognitive Neuroscience, 16*(6), 1000–1009. https://doi.org/10.1162/0898929041502733

Colonius, H., & Diederich, A. (2020). Formal models and quantitative measures of multisensory integration: a selective overview. *European Journal of Neuroscience, 51*(5), 1161–1178. https://doi.org/10.1111/ejn.13813

Colonius, H., Wolff, F. H., & Diederich, A. (2017). Trimodal Race Model Inequalities in Multisensory Integration: I. *Basics. Frontiers in Psychology, 8*, 1141. https://doi.org/10.3389/fpsyg.2017.01141

Cuppini, C., Magosso, E., Bolognini, N., Vallar, G., & Ursino, M. (2014). A neurocomputational analysis of the sound-induced flash illusion. *NeuroImage, 92*, 248–266. https://doi.org/10.1016/j.neuroimage.2014.02.001

Cuppini, C., Shams, L., Magosso, E., & Ursino, M. (2017). A biologically inspired neurocomputational model for audiovisual integration and causal inference. *European Journal of Neuroscience, 46*(9), 2481–2498. https://doi.org/10.1111/ejn.13725

Diederich, A. (1992). *Intersensory facilitation: Race, superposition, and diffusion models for reaction time to multiple stimuli*(vol. 369). Frankfurt am Main ; New York: Peter Lang.

Diederich, A. (1995). Intersensory Facilitation of Reaction Time: Evaluation of Counter and Diffusion Coactivation Models. *Journal of Mathematical Psychology, 39*(2), 197–215. https://doi.org/10.1006/jmps.1995.1020

Ernst, M. O., & Banks, M. S. (2002). Humans integrate visual and haptic information in a statistically optimal fashion. *Nature, 415*(6870), 429–433. https://doi.org/10.1038/415429a

Fang, Y., Yu, Z., Liu, J. K., & Chen, F. (2019). A unified neural circuit of causal inference and multisensory integration. *Neurocomputing, 358*, 355–368. https://doi.org/10.1016/j.neucom.2019.05.067

Ferri, F., Ambrosini, E., & Costantini, M. (2016). Spatiotemporal processing of somatosensory stimuli in schizotypy. *Scientific Reports, 6*(1), 38735. https://doi.org/10.1038/srep38735

Festa, E. K., Katz, A. P., Ott, B. R., Tremont, G., & Heindel, W. C. (2017). Dissociable Effects of Aging and Mild Cognitive Impairment on Bottom-Up Audiovisual Integration. *Journal of Alzheimer's disease: JAD, 59*(1), 155–167. https://doi.org/10.3233/JAD-161062

Fetsch, C. R., DeAngelis, G. C., & Angelaki, D. E. (2013). Bridging the gap between theories of sensory cue integration and the physiology of multisensory neurons. *Nature Reviews Neuroscience, 14*(6), 429–442. https://doi.org/10.1038/nrn3503

French, R. L., & DeAngelis, G. C. (2020). Multisensory neural processing: from cue integration to causal inference. *Current Opinion in Physiology, 16*, 8–13. https://doi.org/10.1016/j.cophys.2020.04.004

Frässle, S., Aponte, E. A., Bollmann, S., Brodersen, K. H., Do, C. T., Harrison, O. K., & Stephan, K. E. (2021). TAPAS: An Open-Source Software Package for Translational Neuromodeling and Computational Psychiatry. *Frontiers in Psychiatry, 12*, Article 680811. https://doi.org/10.3389/fpsyt.2021.680811

Gamma, E., Helm, R., Johnson, R., Vlissides, J., Patterns, D. (1995). *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley.

Gast, R., Rose, D., Salomon, C., Möller, H. E., Weiskopf, N., & Knösche, T. R. (2019). PyRates—A Python framework for rate-based neural simulations. *PLOS ONE, 14*(12), Article e0225900. https://doi.org/10.1371/journal.pone.0225900

Gieseler, A., Tahden, M. A. S., Thiel, C. M., & Colonius, H. (2018). Does hearing aid use affect audiovisual integration in mild hearing impairment? *Experimental Brain Research, 236*(4), 1161–1179. https://doi.org/10.1007/s00221-018-5206-6

Guest, O., & Martin, A. E. (2021). How Computational Modeling Can Force Theory Building in Psychological Science. *Perspectives on Psychological Science: A Journal of the Association for Psychological Science, 16*(4), 789–802. https://doi.org/10.1177/1745691620970585

Haß, K., Sinke, C., Reese, T., Roy, M., Wiswede, D., Dillo, W., & Szycik, G. R. (2017). Enlarged temporal integration window in schizophrenia indicated by the double-flash illusion. *Cognitive Neuropsychiatry, 22*(2), 145–158. https://doi.org/10.1080/13546805.2017.1287693

Hahn, N., Foxe, J. J., & Molholm, S. (2014). Impairments of multisensory integration and cross-sensory learning as pathways to dyslexia. *Neuroscience and Biobehavioral Reviews, 47*, 384–392. https://doi.org/10.1016/j.neubiorev.2014.09.007

Hoyer, S., & Hamman, J. (2017). xarray: Nd labeled arrays and datasets in python. *Journal of Open Research Software, 5*(1), 10–10. https://doi.org/10.5334/jors.148

Jacobson, I., Booch, G., Rumbaugh, J. (2000). *Uml: El proceso unificado de desarrollo de software*. Addison-Wesley.

Karvelis, P., Seitz, A. R., Lawrie, S. M., & Seriès, P. (2018). Autistic traits, but not schizotypy, predict increased weighting of sensory information in bayesian visual integration. *ELife, 7*, Article e34115. https://doi.org/10.7554/eLife.34115

Körding, K. P., Beierholm, U., Ma, W. J., Quartz, S., Tenenbaum, J. B., & Shams, L. (2007). Causal Inference in Multisensory Perception. *PLoS ONE, 2*(9), Article e943. https://doi.org/10.1371/journal.pone.0000943

Ma, W. J., & Pouget, A. (2008). Linking neurons to behavior in multisensory perception: A computational review. *Brain Research, 1242*, 4–12. https://doi.org/10.1016/j.brainres.2008.04.082

Ma, W. J., & Rahmati, M. (2013). Towards a neural implementation of causal inference in cue combination. *Multisensory Research, 26*(1–2), 159–176. https://doi.org/10.1163/22134808-00002407

Marr, D. (2010). *Vision: A Computational Investigation into the Human Pepresentation and Processing of Visual Information*. Cambridge, Mass: MIT Press.

Martin, B., Giersch, A., Huron, C., & van Wassenhove, V. (2013). Temporal event structure and timing in schizophrenia: Preserved binding in a longer "now". *Neuropsychologia, 51*(2), 358–371. https://doi.org/10.1016/j.neuropsychologia.2012.07.002

Martin, R. C. (2000). Design principles and design patterns. *Object Mentor, 1*(34), 597.

Meijer, D., & Noppeney, U. (2020). Computational models of multisensory integration. *Multisensory Perception* (pp. 113–133). Elsevier.

Miller, R. L., Stein, B. E., & Rowland, B. A. (2017). Multisensory Integration Uses a Real-Time Unisensory-Multisensory Transform.

The Journal of Neuroscience: The Official Journal of the Society for Neuroscience, 37(20), 5183–5194. https://doi.org/10.1523/JNEUROSCI.2767-16.2017

Noel, J. P., & Angelaki, D. E. (2023). A theory of autism bridging across levels of description. *Trends in Cognitive Sciences, 27*(7), 631–641. https://doi.org/10.1016/j.tics.2023.04.010

Noel, J. P., Paredes, R., Terrebonne, E., Feldman, J. I., Woynaroski, T., Cascio, C. J., & Wallace, M. T. (2022a). Inflexible Updating of the Self-Other Divide During a Social Context in Autism: Psychophysical, Electrophysiological, and Neural Network Modeling Evidence. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging, 7*(8), 756–764. https://doi.org/10.1016/j.bpsc.2021.03.013

Noel, J.P., Shivkumar, S., Dokka, K., Haefner, R.M., Angelaki, D.E. (2022b). Aberrant causal inference and presence of a compensatory mechanism in autism spectrum disorder. *eLife*, *11*, e71866. https://doi.org/10.7554/eLife.71866

Ohshiro, T., Angelaki, D. E., & DeAngelis, G. C. (2011). A normalization model of multisensory integration. *Nature Neuroscience, 14*(6), 775–782. https://doi.org/10.1038/nn.2815

Paredes, R., Ferri, F., Romei, V., & Seriès, P. (2025). Increased excitation enhances the sound-induced flash illusion by impairing multisensory causal inference in the schizophrenia spectrum. *Schizophrenia Research, 283*, 1–10. https://doi.org/10.1016/j.schres.2025.06.007

Paredes, R., Ferri, F., & Seriès, P. (2022). Influence of E/I balance and pruning in peri-personal space differences in schizophrenia: A computational approach. *Schizophrenia Research, 248*, 368–377. https://doi.org/10.1016/j.schres.2021.06.026

Paredes, R., Seriès, P., & Cabral, J. (2023). Scikit-NeuroMSI: a Python framework for multisensory integration modelling. *IX Congreso de Matemática Aplicada, Computacional e Industrial, 9*, 545–548.

Parise, C. V., & Ernst, M. O. (2016). Correlation detection as a general mechanism for multisensory integration. *Nature Communications, 7*, 11543. https://doi.org/10.1038/ncomms11543

Perez, F., Granger, B. E., & Hunter, J. D. (2010). Python: an ecosystem for scientific computing. *Computing in Science & Engineering, 13*(2), 13–21. https://doi.org/10.1109/MCSE.2010.119

Pesnot Lerousseau, J., Parise, C. V., Ernst, M. O., & Van Wassenhove, V. (2022). Multisensory correlation computations in the human brain identified by a time-resolved encoding model. *Nature Communications, 13*(1), 2489. https://doi.org/10.1038/s41467-022-29687-6

PKWARE Inc. (2022). *.ZIP file format specification* (Technical Specification No. APPNOTE.TXT). 201 E. Pittsburgh Avenue, Suite 400, Milwaukee, WI 53204:PKWARE Inc. http://www.pkware.com/appnote. (Status: FINAL - replaces version 6.3.9)

Ramkhalawansingh, R., Keshavarz, B., Haycock, B., Shahab, S., & Campos, J. L. (2017). Examining the Effect of Age on Visual-Vestibular Self-Motion Perception Using a Driving Paradigm. *Perception, 46*(5), 566–585. https://doi.org/10.1177/0301006616675883

Rew, R., & Davis, G. (1990). NetCDF: an interface for scientific data access. *IEEE Computer Graphics and Applications, 10*(4), 76–82. https://doi.org/10.1109/38.56302

Rideaux, R., Storrs, K. R., Maiello, G., & Welchman, A. E. (2021). How multisensory neurons solve causal inference. *Proceedings of the National Academy of Sciences, 118*(32), Article e2106235118. https://doi.org/10.1073/pnas.2106235118

Rohe, T., Ehlis, A. C., & Noppeney, U. (2019). The neural dynamics of hierarchical Bayesian causal inference in multisensory perception. *Nature Communications, 10*(1), 1907. https://doi.org/10.1038/s41467-019-09664-2

Rohe, T., & Noppeney, U. (2015). Cortical Hierarchies Perform Bayesian Causal Inference in Multisensory Perception. *PLoS Biology, 13*(2), Article e1002073. https://doi.org/10.1371/journal.pbio.1002073

Rossum, G.v., & Drake, F.L. (2010). *The Python language reference* (Release 3.0.1 [Repr.] ed.) (No. Pt. 2). Hampton, NH: Python Software Foundation.

Shams, L., & Beierholm, U. (2022). Bayesian causal inference: A unifying neuroscience theory. *Neuroscience & Biobehavioral Reviews, 137*, Article 104619. https://doi.org/10.1016/j.neubiorev.2022.104619

Shams, L., & Beierholm, U. R. (2010). Causal inference in perception. *Trends in Cognitive Sciences, 14*(9), 425–432. https://doi.org/10.1016/j.tics.2010.07.001

Shams, L., Kamitani, Y., & Shimojo, S. (2002). Visual illusion induced by sound. *Cognitive Brain Research, 14*(1), 147–152. https://doi.org/10.1016/S0926-6410(02)00069-1

Sánchez, A., Millán-Calenti, J. C., Lorenzo-López, L., & Maseda, A. (2013). Multisensory stimulation for people with dementia: A review of the literature. *American Journal of Alzheimer's Disease and Other Dementias, 28*(1), 7–14. https://doi.org/10.1177/1533317512466693

Stein, B. E., Burr, D., Constantinidis, C., Laurienti, P. J., Meredith, M. A., Perrault, T. J., & Lewkowicz, D. J. (2010). Semantic confusion regarding the development of multisensory integration: a practical solution. *The European journal of neuroscience, 31*(10), 1713–1720. https://doi.org/10.1111/j.1460-9568.2010.07206.x

Stein, B. E., & Stanford, T. R. (2008). Multisensory integration: current issues from the perspective of the single neuron. *Nature Reviews Neuroscience, 9*(4), 255–266. https://doi.org/10.1038/nrn2331

Stein, B. E., Stanford, T. R., & Rowland, B. A. (2020). Multisensory integration and the society for neuroscience: Then and now. *Journal of Neuroscience, 40*(1), 3–11. https://doi.org/10.1523/JNEUROSCI.0737-19.2019

Stevenson, R. A., Siemann, J. K., Woynaroski, T. G., Schneider, B. C., Eberly, H. E., Camarata, S. M., & Wallace, M. T. (2014). Evidence for diminished multisensory integration in autism spectrum disorders. *Journal of Autism and Developmental Disorders, 44*(12), 3161–3167. https://doi.org/10.1007/s10803-014-2179-6

Stimberg, M., Brette, R., Goodman, D.F. (2019). Brian 2, an intuitive and efficient neural simulator. *eLife*, *8*, e47314. https://doi.org/10.7554/eLife.47314

Storn, R., & Price, K. (1997). Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization, 11*, 341–359. https://doi.org/10.1023/A:1008202821328

Thurlow, W. R., & Jack, C. E. (1973). Certain Determinants of the "Ventriloquism Effect". *Perceptual and Motor Skills, 36*(3), 1171–1184. https://doi.org/10.2466/pms.1973.36.3c.1171

Ursino, M., Crisafulli, A., di Pellegrino, G., Magosso, E., & Cuppini, C. (2017). Development of a Bayesian Estimator for Audio-Visual Integration: A Neurocomputational Study. *Frontiers in Computational Neuroscience, 11*, 89. https://doi.org/10.3389/fncom.2017.00089

Ursino, M., Cuppini, C., & Magosso, E. (2014). Neurocomputational approaches to modelling multisensory integration in the brain: A review. *Neural Networks, 60*, 141–165. https://doi.org/10.1016/j.neunet.2014.08.003

Ursino, M., Cuppini, C., Magosso, E., Beierholm, U., & Shams, L. (2019). Explaining the Effect of Likelihood Manipulation and Prior Through a Neural Network of the Audiovisual Perception of Space. *Multisensory Research, 32*(2), 111–144. https://doi.org/10.1163/22134808-20191324

Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D.. others (2020). Scipy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

Wallace, M. T., Woynaroski, T. G., & Stevenson, R. A. (2020). Multisensory Integration as a Window into Orderly and Disrupted Cognition and Communication. *Annual Review of Psychology, 71*(1), 193–219. https://doi.org/10.1146/annurev-psych-010419-051112

Wang, C., Zhang, T., Chen, X., He, S., Li, S., Wu, S. (2023). Brainpy, a flexible, integrative, efficient, and extensible framework for general-purpose brain dynamics programming. *eLife*, *12*, e86365. https://doi.org/10.7554/eLife.86365

Wiecki, T. V., Sofer, I., & Frank, M. J. (2013). HDDM: Hierarchical Bayesian estimation of the Drift-Diffusion Model in Python. *Frontiers in Neuroinformatics, 7*,. https://doi.org/10.3389/fninf.2013.00014

Wilson, R.C., & Collins, A.G. (2019). Ten simple rules for the computational modeling of behavioral data. *eLife*, *8*, e49547. https://doi.org/10.7554/eLife.49547

Wu, J., Yang, J., Yu, Y., Li, Q., Nakamura, N., Shen, Y., & Abe, K. (2012). Delayed audiovisual integration of patients with mild cognitive impairment and Alzheimer's disease compared with normal aged controls. *Journal of Alzheimer's disease: JAD, 32*(2), 317–328. https://doi.org/10.3233/JAD-2012-111070

Zhou, H.y., Cai, X.l., Weigl, M., Bang, P., Cheung, E.F., Chan, R.C. (2018). Multisensory temporal binding window in autism spectrum disorders and schizophrenia spectrum disorders: A systematic review and meta-analysis. *Neuroscience & Biobehavioral Reviews, 86*, 66–76. https://doi.org/10.1016/j.neubiorev.2017.12.013

Zhu, H., Beierholm, U., & Shams, L. (2024). BCI toolbox: An open-source Python package for the Bayesian Causal Inference model. *PLoS Computational Biology, 20*(7), Article e1011791. https://doi.org/10.1371/journal.pcbi.1011791

Zhu, H., Beierholm, U., & Shams, L. (2024). The overlooked role of unisensory precision in multisensory research. *Current Biology, 34*(6), R229–R231. https://doi.org/10.1016/j.cub.2024.01.057

Zvyagintsev, M., Parisi, C., & Mathiak, K. (2017). Temporal processing deficit leads to impaired multisensory binding in schizophrenia. *Cognitive Neuropsychiatry, 22*(5), 361–372. https://doi.org/10.1080/13546805.2017.1331160