Final Talk

Simulation of EEG Activity based on Sequential Sampling Models

M.Sc. Business Informatics
Timo Zaoral

Table of Content

Introduction

What is this thesis about?

- Human **decision-making**
- **Sequential Sampling Models (SSMs)**
- Simulating **EEG activity**

Why is this important?

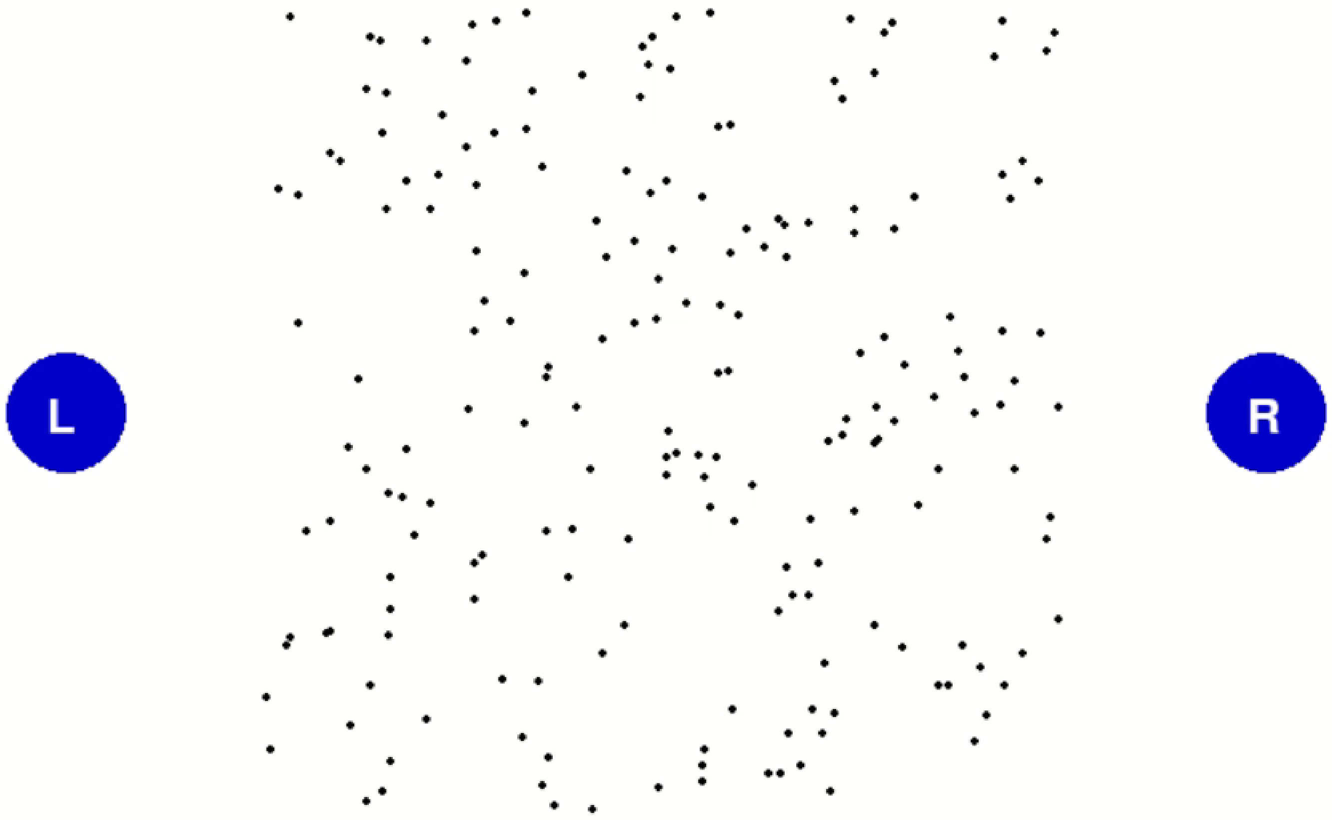
- Insights **interpreting** decision-related signals
- Open **debate in research**
- **Simulated EEG** data is **useful** for multiple tasks

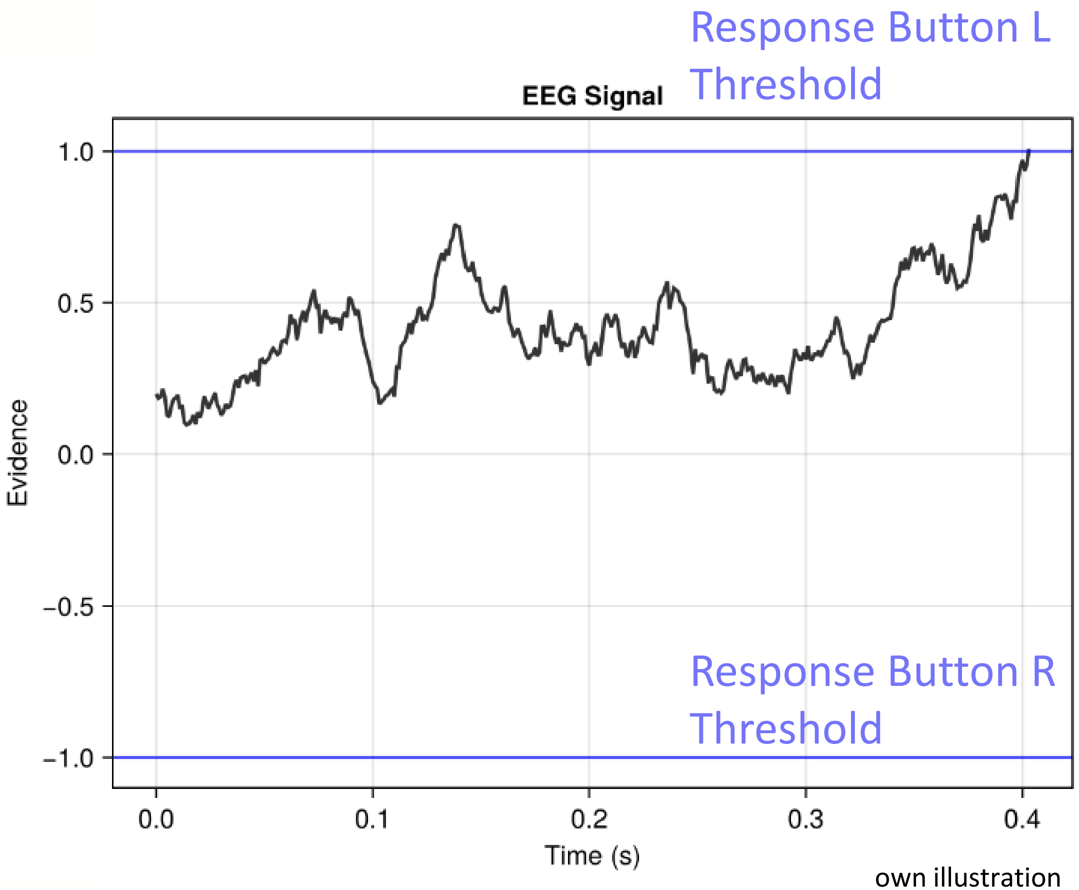
Main Research Goal:

- Extension **development**
- Integration into **UnfoldSim.jl**
- **Investigate** how SSMs can replicate **decision-making**

Motivation & related work

Changing coherence moving dots Experiment

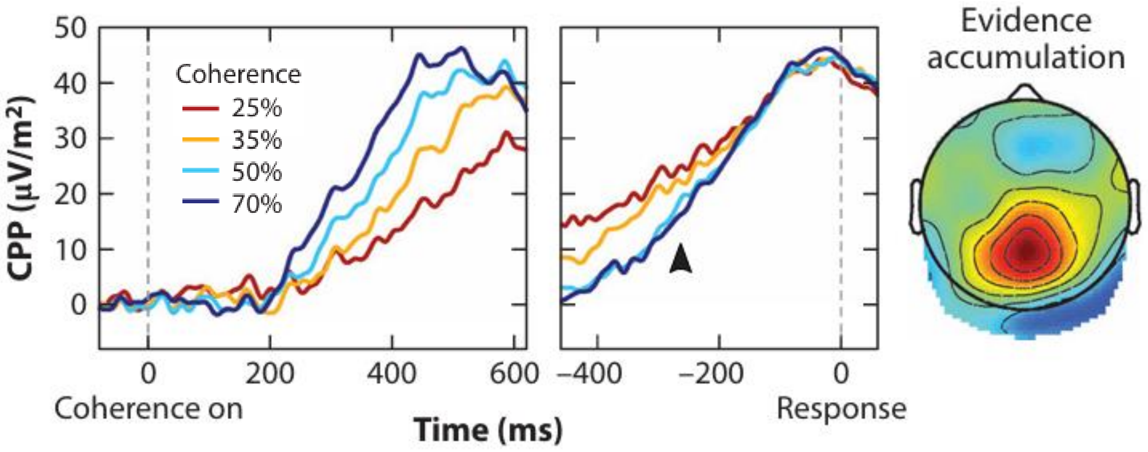




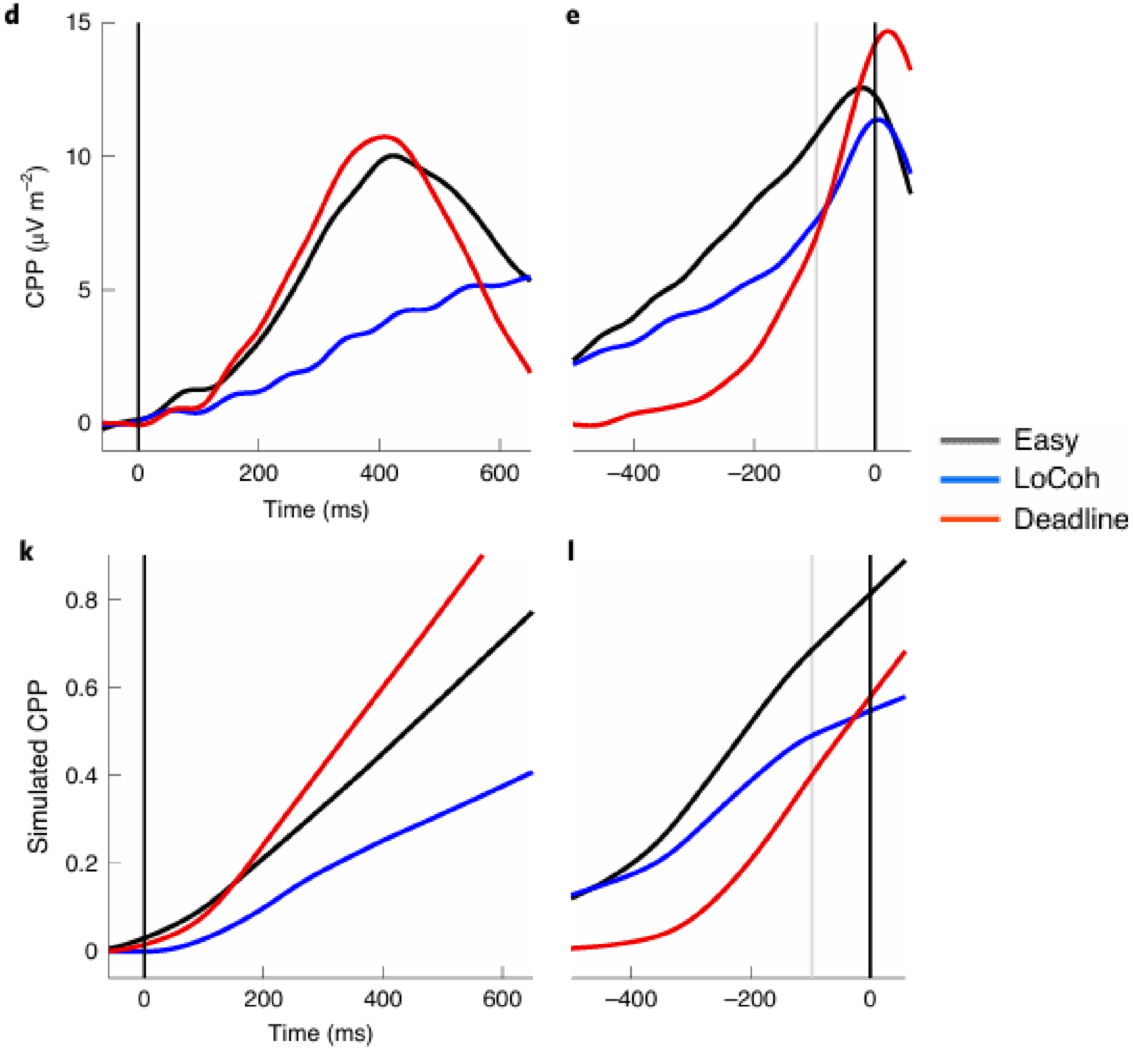
own illustration

Motivation & related work

Evidence-dependent structural dynamics of the component centroparietal positivity (CPP) in the brain

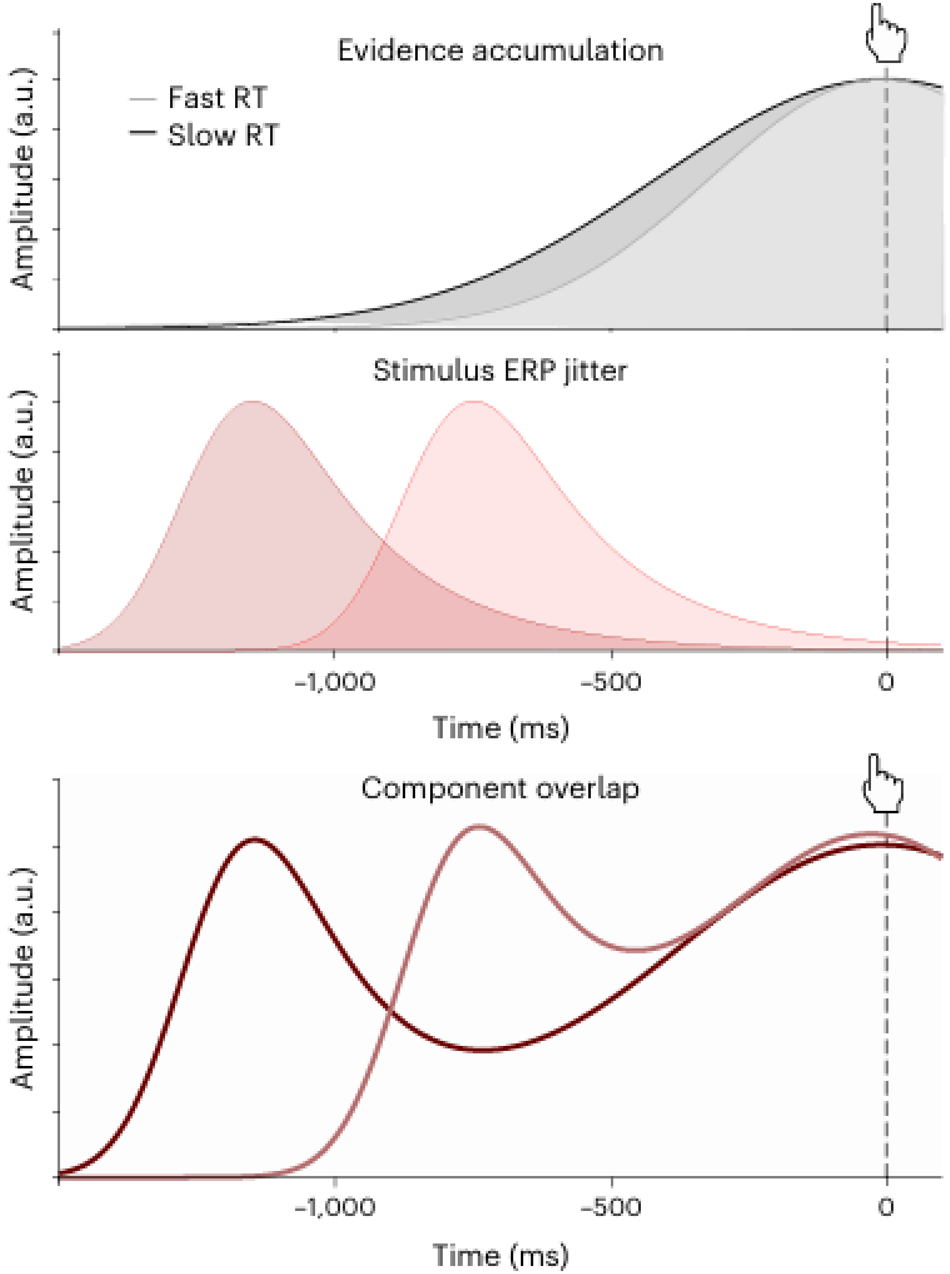


Kelly et al. Neurophysiology of Human Perceptual Decision-Making 2021 [1]



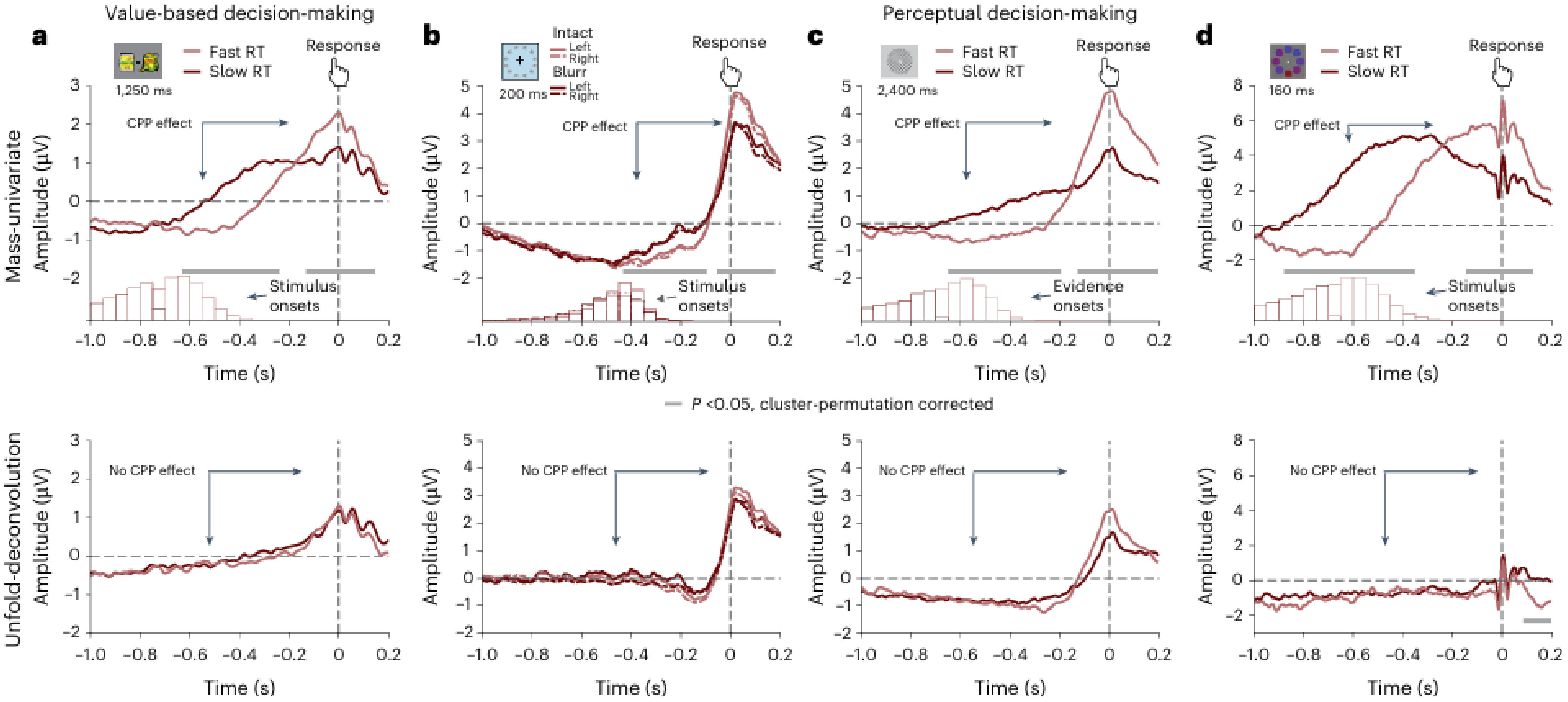
Kelly et al. Neurocomputational mechanisms of prior-informed perceptual decision-making in humans 2021 [2]

Motivation & related work



Frömer et al. Common neural choice signals can emerge artefactually amid multiple distinct value signals 2024 [3]

Motivation & related work



Frömer et al. Common neural choice signals can emerge artefactually amid multiple distinct value signals 2024 [3]

Motivation & related work

Challenges and Motivation

Understanding
Insights interpreting decision-related signals

Deconvolution
Techniques can separate overlapping components but need validation

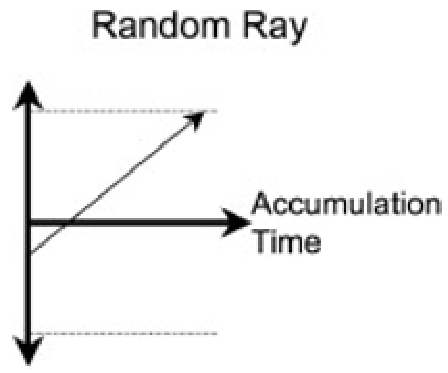
EEG Overlap
Signals overlap, making it hard to distinguish actual decision-making activity

This study provides a flexible EEG simulation extension to test for example such models systematically

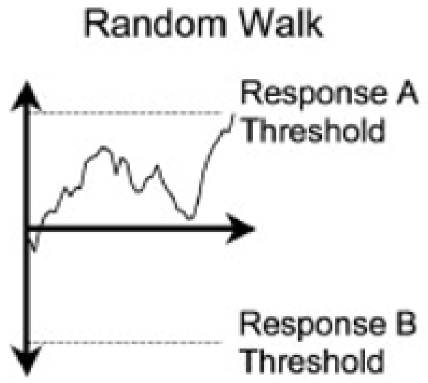
Approach

1. Theoretical Description of Sequential Sampling Models

linear ballistic accumulator (LBA)

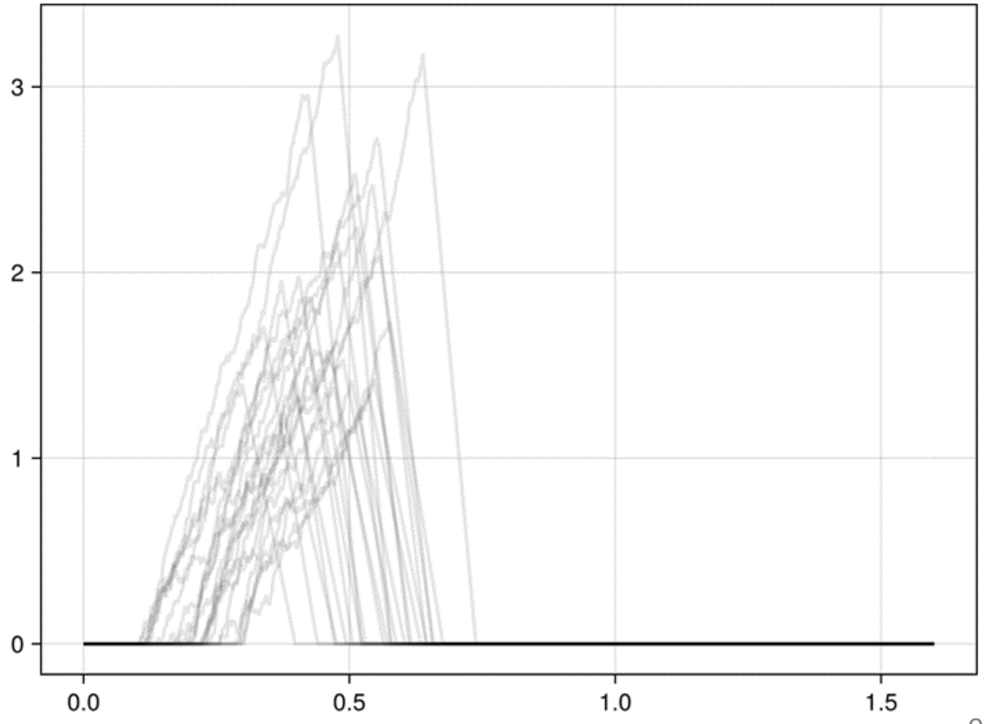


drift-diffusion model (DDM)



Brown et al. The simplest complete model of choice response time: Linear ballistic accumulation 2008 [4]

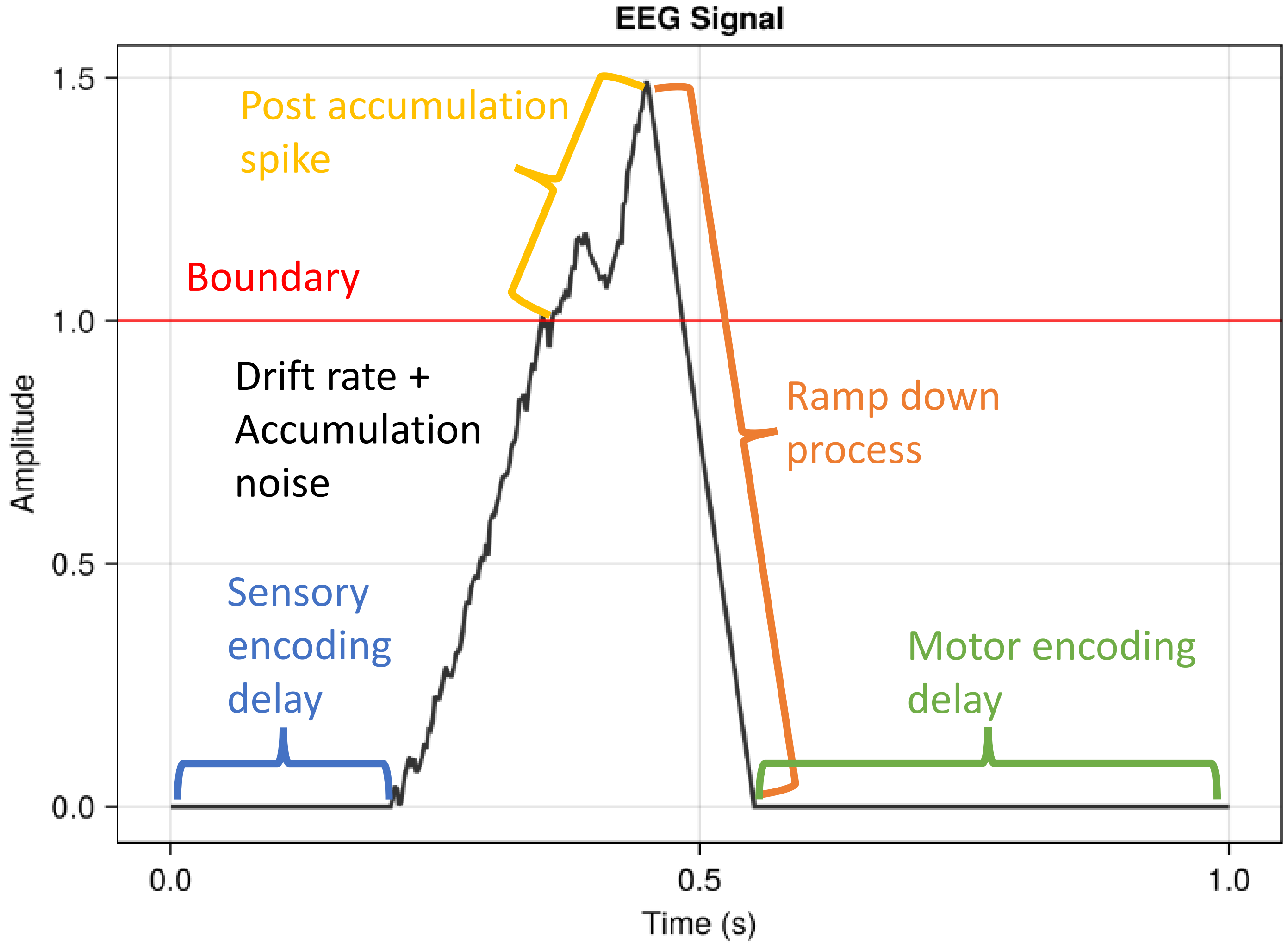
Kelly et. al Model of neural activity



Own illustration based on Kelly et al. Code converted to Julia

Approach

Kelly et. al Model of neural activity



Own illustration based on Kelly et al. Code converted to Julia

Approach

2. Reimplementation of the Kelly et. al Model

3. Integration into UnfoldSim

2.1 Understanding the code

3.1 Introduction to UnfoldSim.jl usage

2.2 Structured and methodical reimplementation

3.2 Adapting implementation to package paradigms

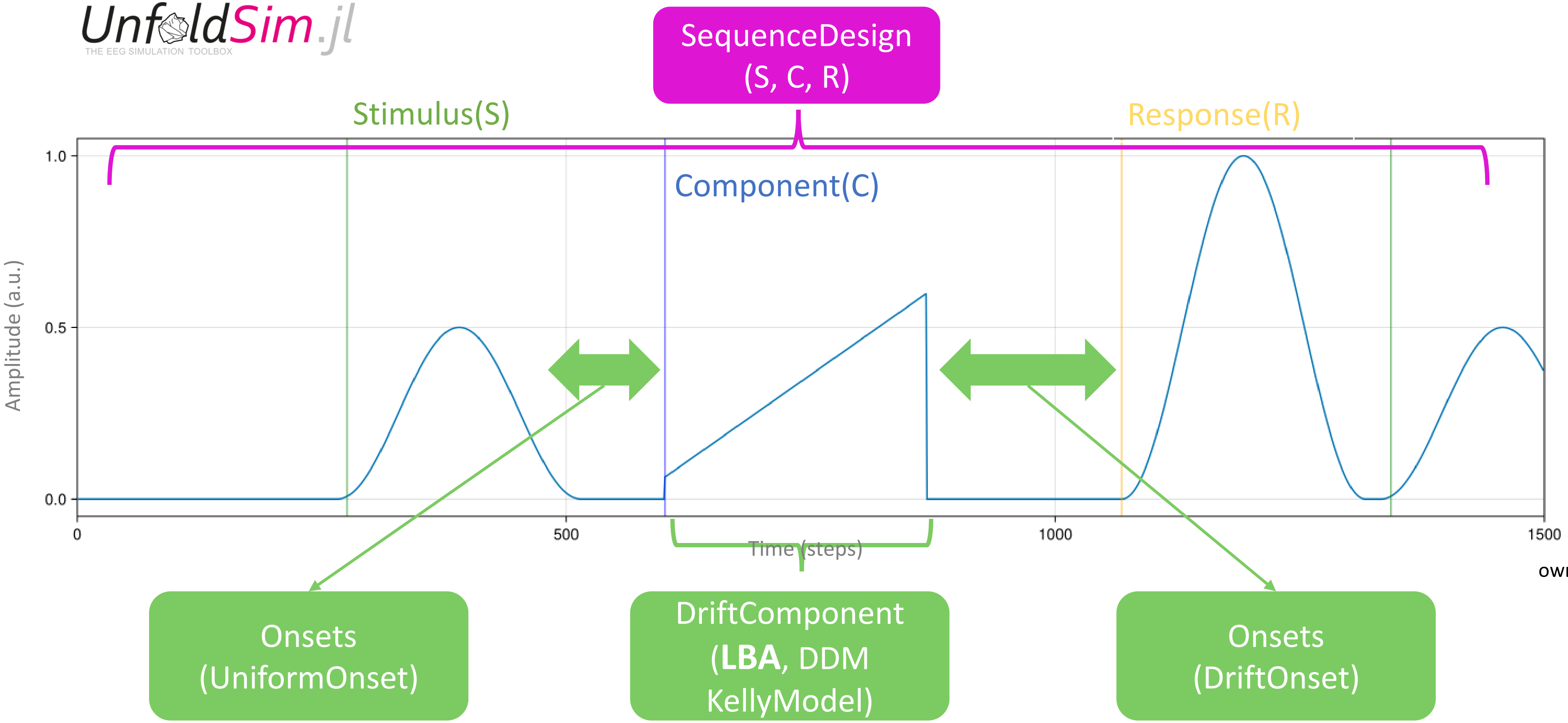
2.3 Parameters Docu.

```
u1 = Z .+ (U.+randn(rng).*Su) .* timevec;
u1 = u1 .+ (rand(rng)-.5)*Sz;
u2 = Z .+ (U+randn(rng).*Su) .* timevec;
u2 = u2 .+ (rand(rng)-.5)*Sz;
```

3.3 Splitting functionalities and documenting code

Implementation & Integration – Complete picture

UnfoldSim.jl
THE EEG SIMULATION TOOLBOX



own illustration

Implementation & Integration

UnfoldSim.jl
THE EEG SIMULATION TOOLBOX

Design

- SingleSubjectDesign
- SequenceDesign
- RepeatDesign

Component

Model simulation

Onset

Implementation & Integration

UnfoldSim.jl

THE EEG SIMULATION TOOLBOX

Design

- SingleSubjectDesign
- SequenceDesign
- RepeatDesign

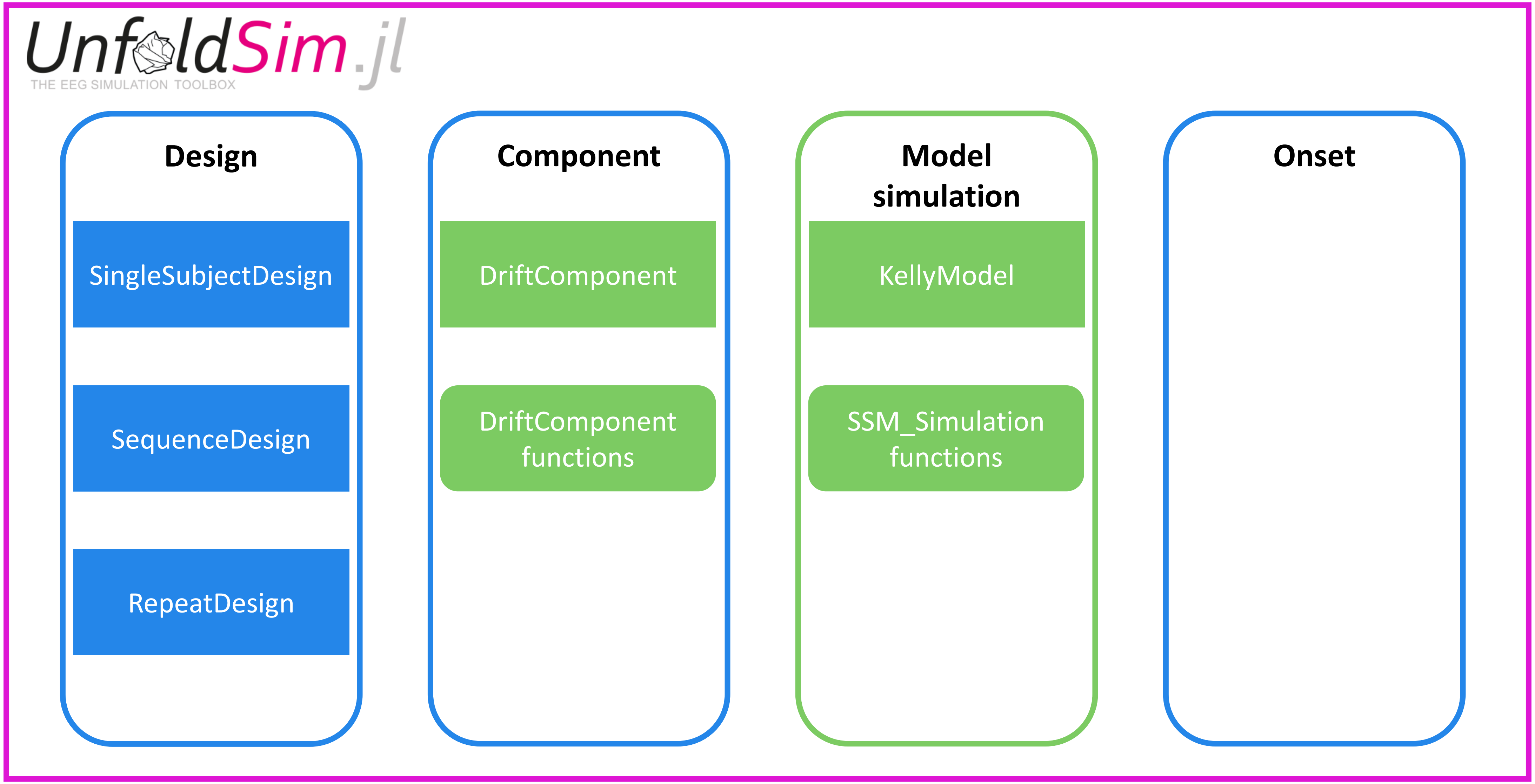
Component

- DriftComponent
- DriftComponent functions

Model simulation

Onset

Implementation & Integration



Implementation & Integration

UnfoldSim.jl
THE EEG SIMULATION TOOLBOX

Design
- SingleSubjectDesign
- SequenceDesign
- RepeatDesign

Component
- DriftComponent
- DriftComponent functions

Model simulation
- KellyModel
- SSM_Simulation functions

Onset
- SequenceOnset
- DriftOnset
- SequenceOnset/ DriftOnset functions

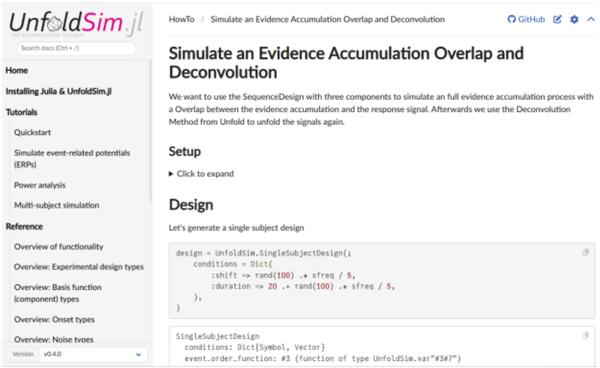
Implementation & Integration

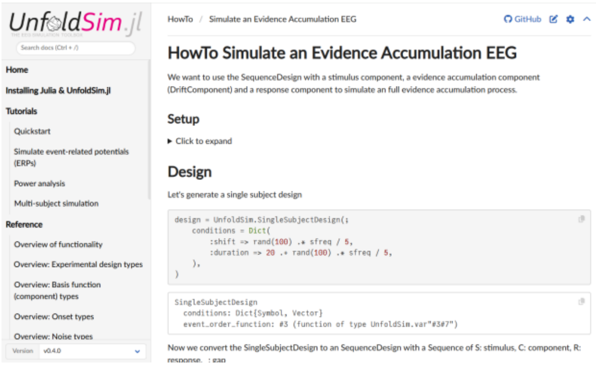
Test

```julia
@testset "sequentialSamplingModelSimulation" begin
    fs = 500
    Δt = 1 / fs # time step
    tEnd = 1.0 # trial Duration
    time_vec = 0:Δt:tEnd # time base
    max_length = tEnd / Δt
    rng = StableRNG(1)
    @testset "KellyModel" begin
        assert_event_onset = 0.663
        assert_drift_rate = "drift_rate"
        km = KellyModel(event_onset = assert_event_onset, drift_rate = assert_drift_rate)

        @test km.event_onset == assert_event_onset
        @test km.drift_rate == assert_drift_rate
    end

    @testset "KellyModel_simulate_cpp" begin
        boundary = 1.0
        result_rt, result_trace = UnfoldSim.KellyModel_simulate_cpp(
            rng,
            KellyModel(boundary = boundary),
            time_vec,
            Δt,
        )
        @test size(result_rt) == ()
        @test size(result_trace) == (501,)
        @test isapprox(result_rt, 399.6903067274333, atol = 1e-8)
        @test any(result_trace .== 0)
        @test any(result_trace .>= boundary)

        result_sim_rt, result_sim_trace = UnfoldSim.SSM_Simulate(rng, KellyModel(), fs, max_length)
        @test result_rt == result_sim_rt
        @test result_trace == result_sim_trace
    end
```

Documentation

````julia
"""
    SSM_Simulate(rng, model::KellyModel, sfreq, max_length)

Generate response time and evidence Vector of max_length by using the Kelly Model for the simulation.

# Arguments
- `rng::StableRNG`: Random seed to ensure the same traces are created for reconstruction.
- `model::KellyModel`: SequentialSamplingModel to simulate the evidence and response time.
- `sfreq::Real`: sample frequency used to simulate the signal.
- `max_length::Int`: maximum length of the simulated signal.

# Returns
- `Float64`: Simulated response time for the trial.
- `Vector{Float64}`: evidence values over time. The output dimension is `c.max_length`.

# Examples
```julia-repl
julia> model = KellyModel()

julia> SSM_Simulate(StableRNG(1), model, 500, 500)
Float64, Vector{Float64}:
(96.65745162948949, [0.0 0.0 … 0.0 0.0])
```
"""
function SSM_Simulate(rng, model::KellyModel, sfreq, max_length)
````





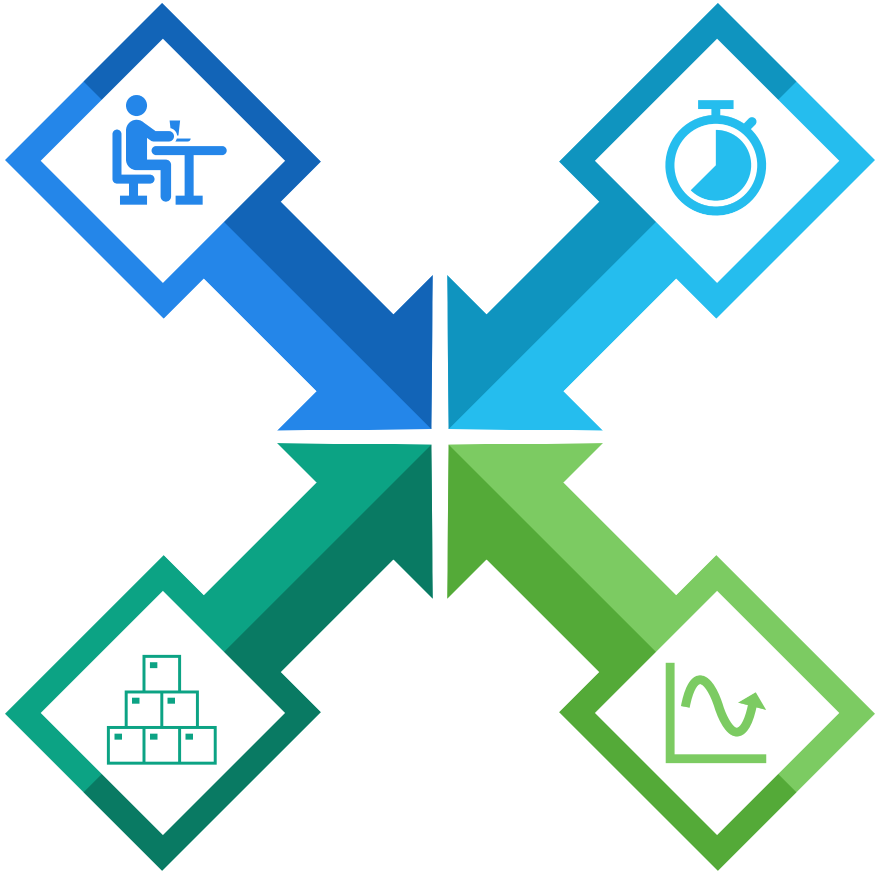
Results

1. Simulation Space
   1. Setup for all model simulations
   2. LBA
   3. DDM
   4. KellyModel
2. Use Case
   1. Setup for the use case
   2. Results of a basic overlap deconvolution
   3. Results of the simulated data

Results – Simulation Space - Setup

Design

```
design_single = SingleSubjectDesign(
    conditions = Dict(
        :drift_rate => [0.45, 0.8],
        :condition => [1])
    )
design_seq = SequenceDesign(design_single, "SCR")
design_rep = RepeatDesign(design_seq, 100)
```



Onsets

```
fs = 500
component_length = 1.0
seq_onset = SequenceOnset(
    Dict('S'=> UniformOnset(
                width=0,
                offset=(comp_length*fs*1.2)),
            'C'=> DriftOnset(),
            'R'=> UniformOnset(
                width=0,
                offset=(comp_length*fs*1.5)))
    )
```

Components

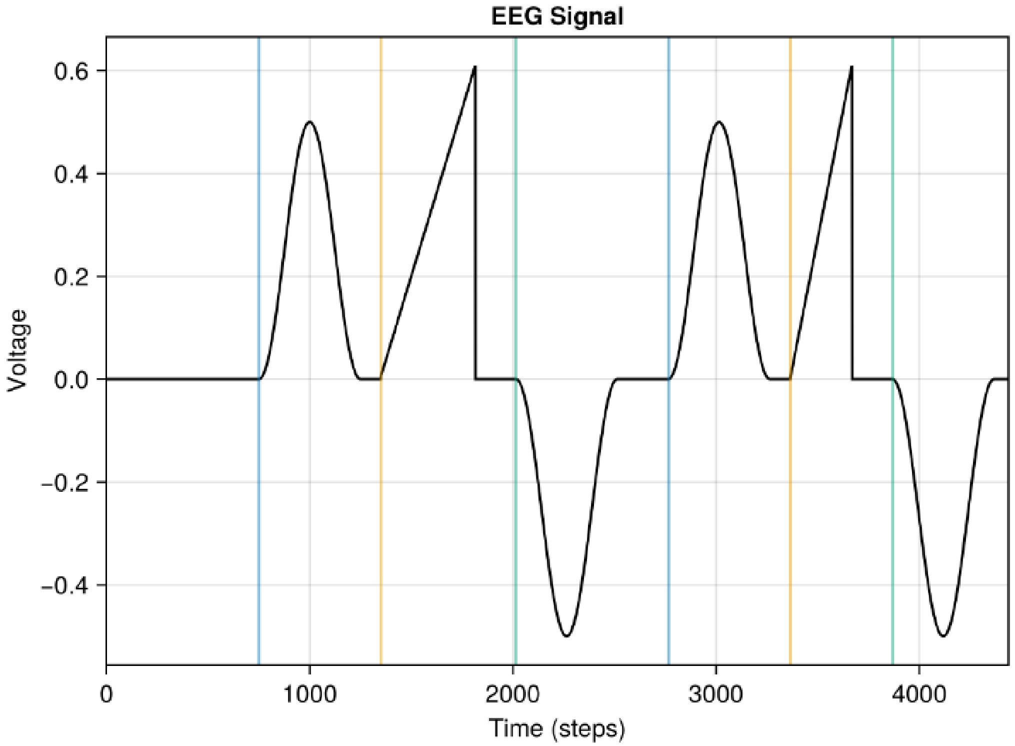
```
components = Dict(
    'S' => [stimulus],
    'C' => [drift_component],
    'R' => [response]
    )
```

Simulation

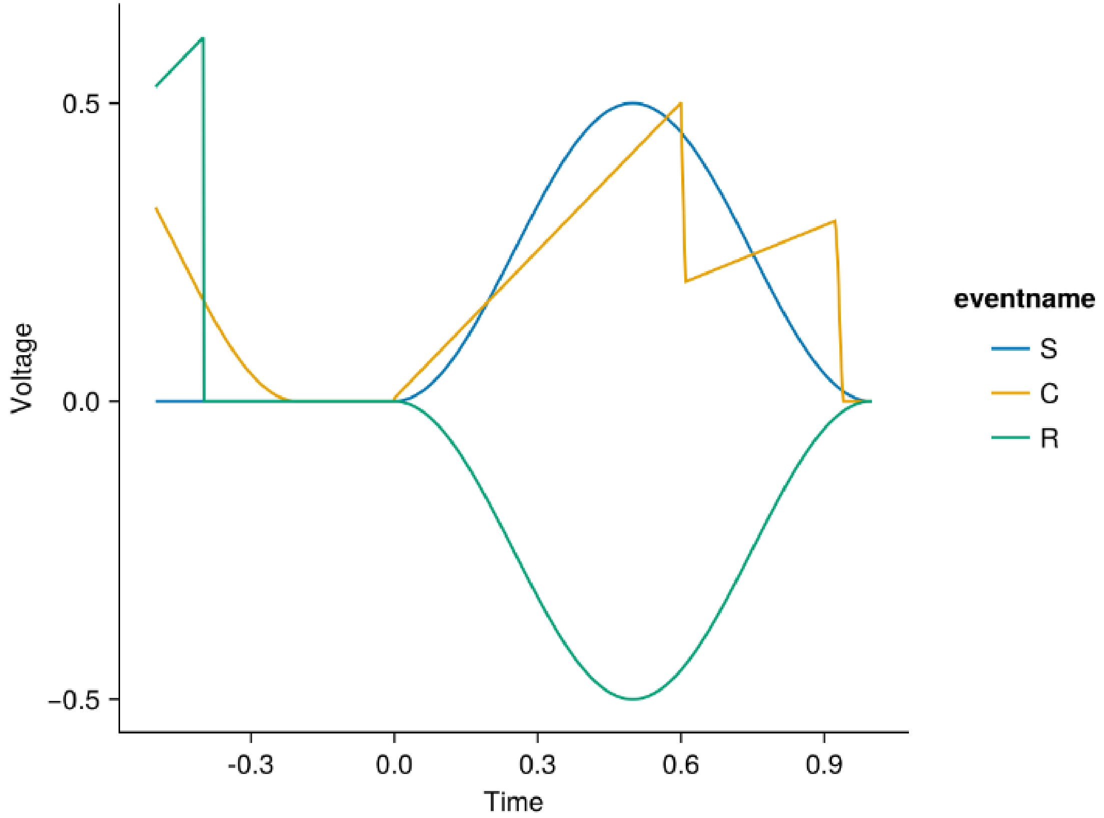
```
data, events = UnfoldSim.simulate(
    StableRNG(12),
    design_rep,
    components,
    seq_onset,
    NoNoise()
)
```

Results – Simulation Space - LBA

Legend
Stimulus(S) Component(C) Response(R)



own illustration

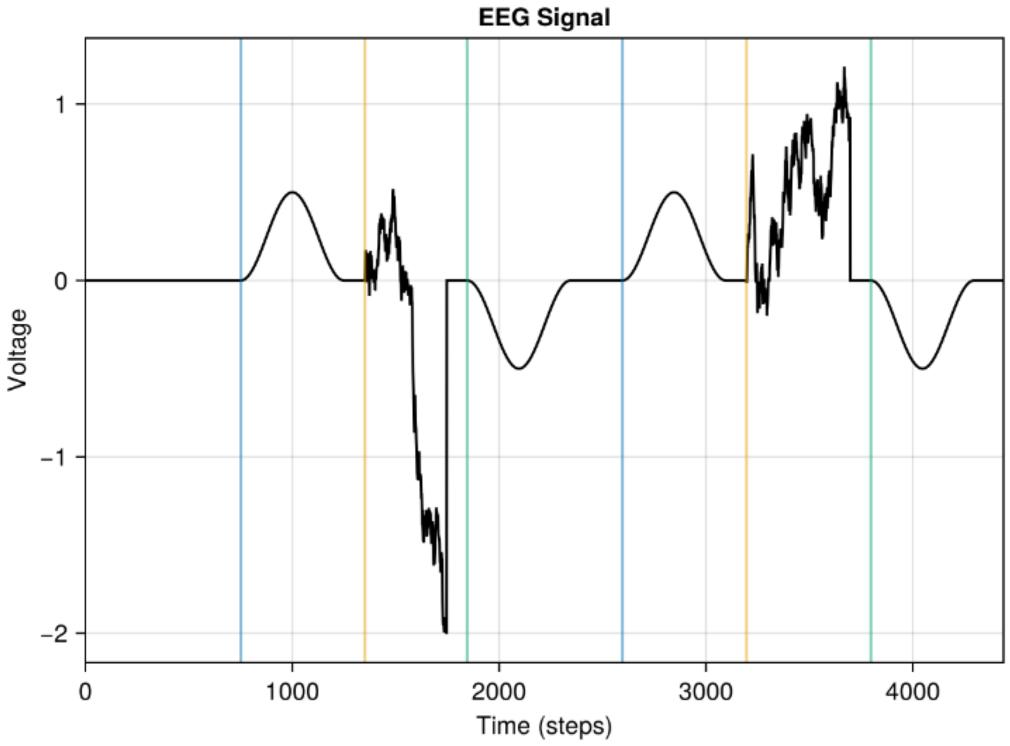


own illustration

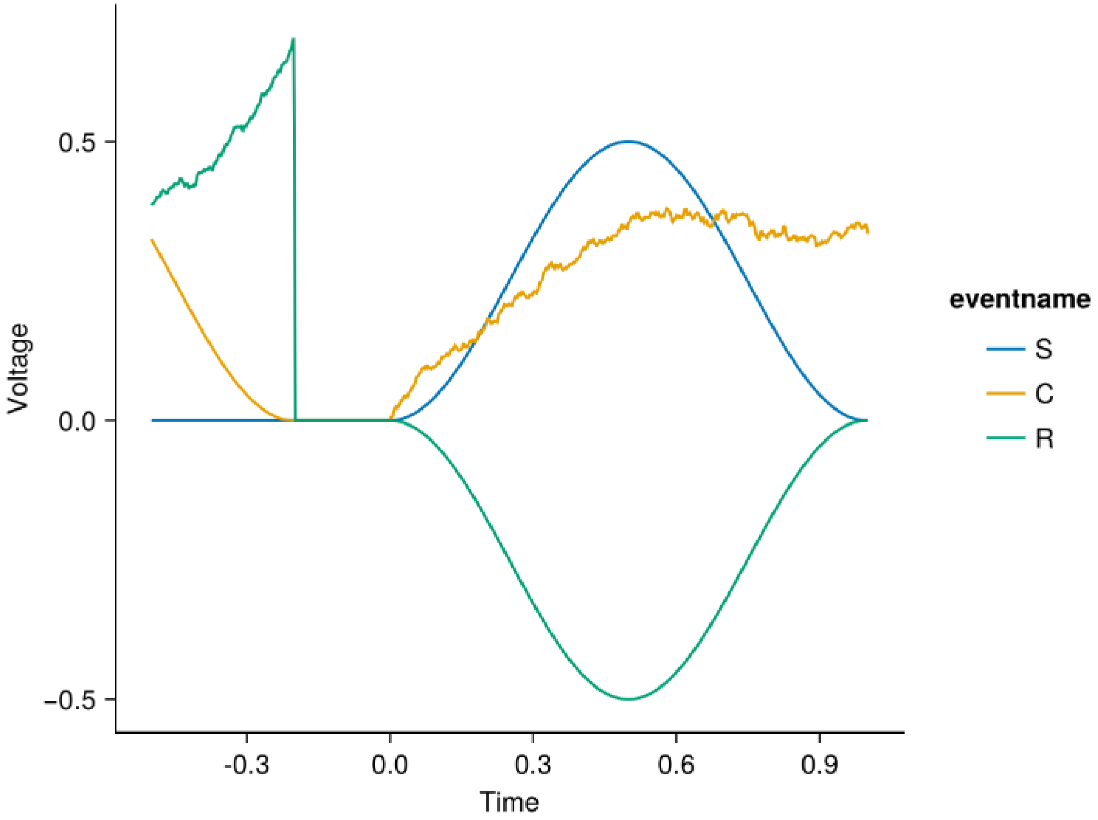
Results – Simulation Space - DDM

Legend

Stimulus(S) Component(C) Response(R)



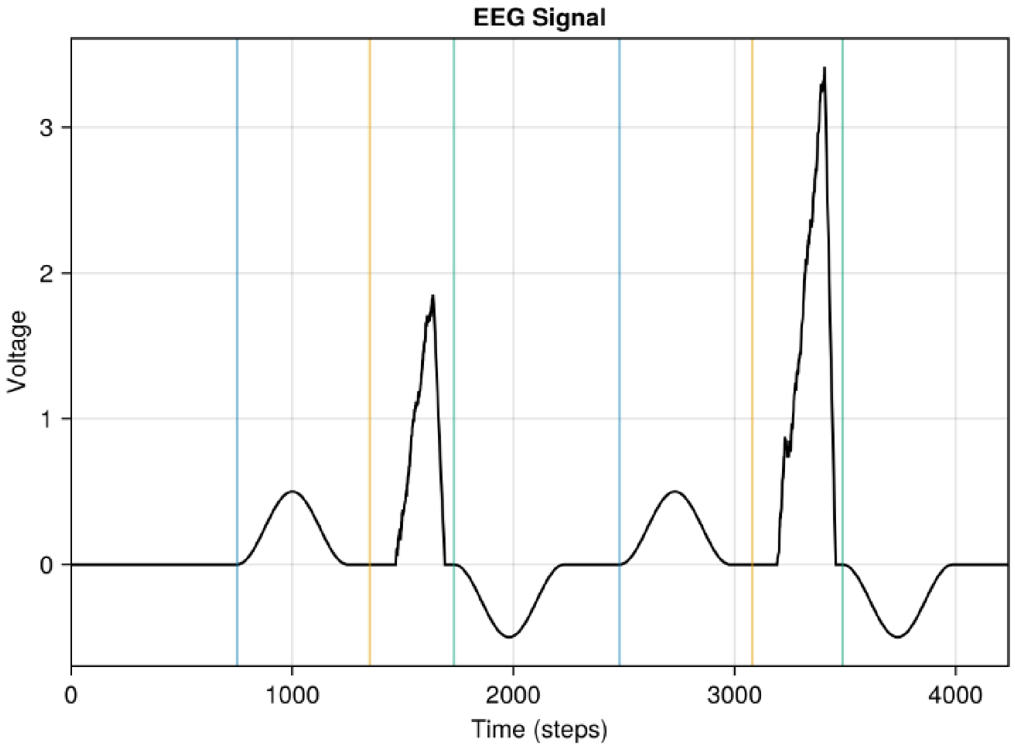
own illustration



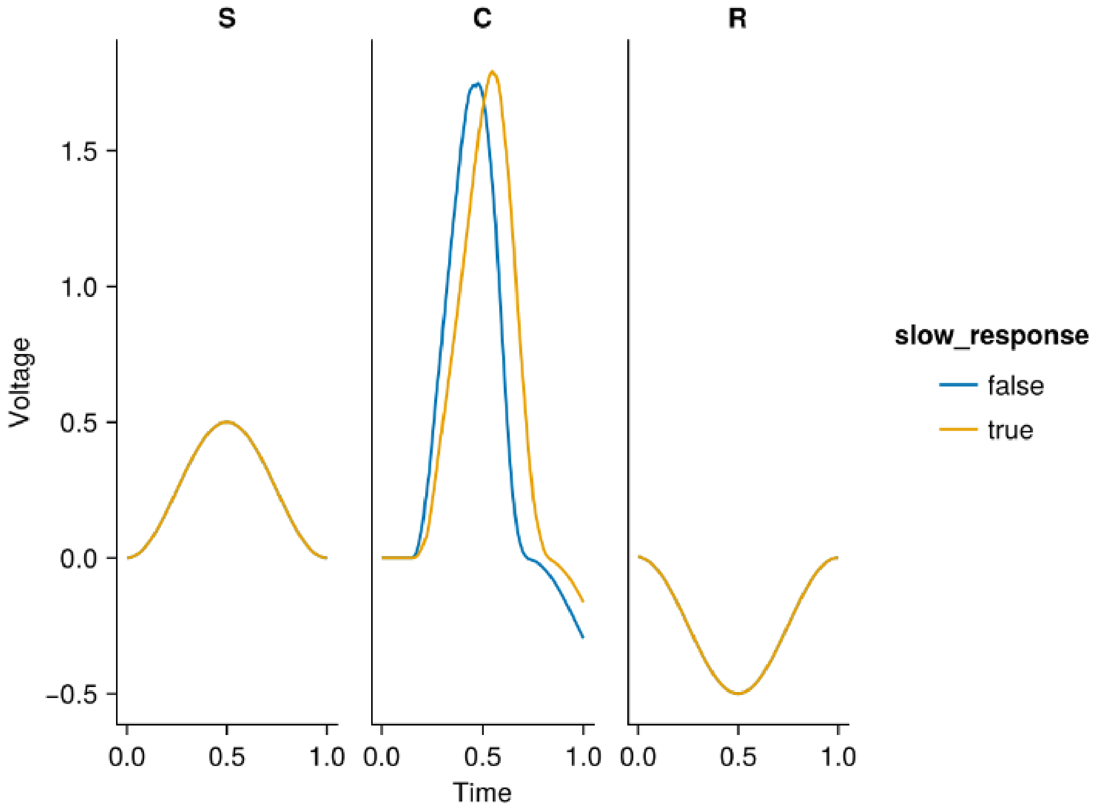
own illustration

Results – Simulation Space - KellyModel

Legend
Stimulus(S) Component(C) Response(R)

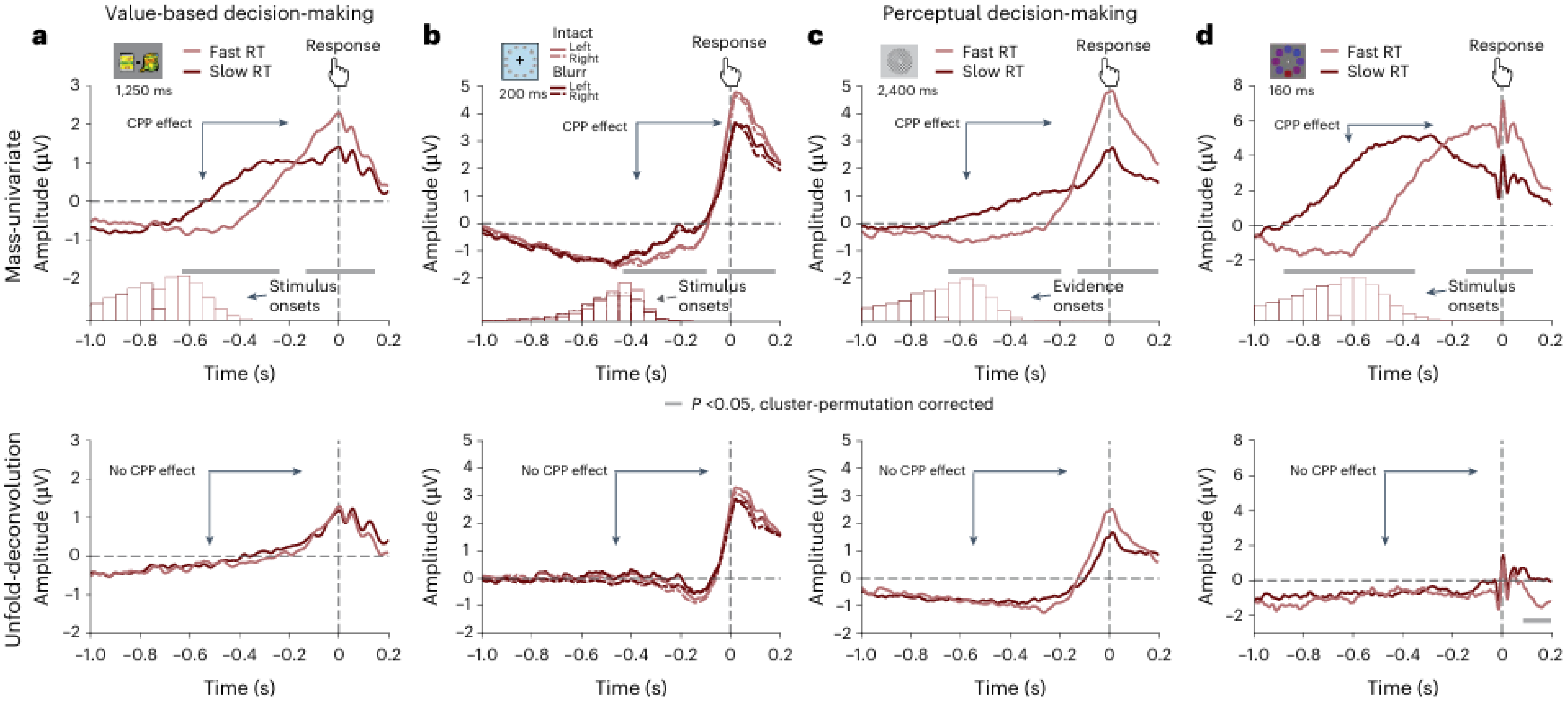


own illustration



own illustration

Results – No component activity after deconv

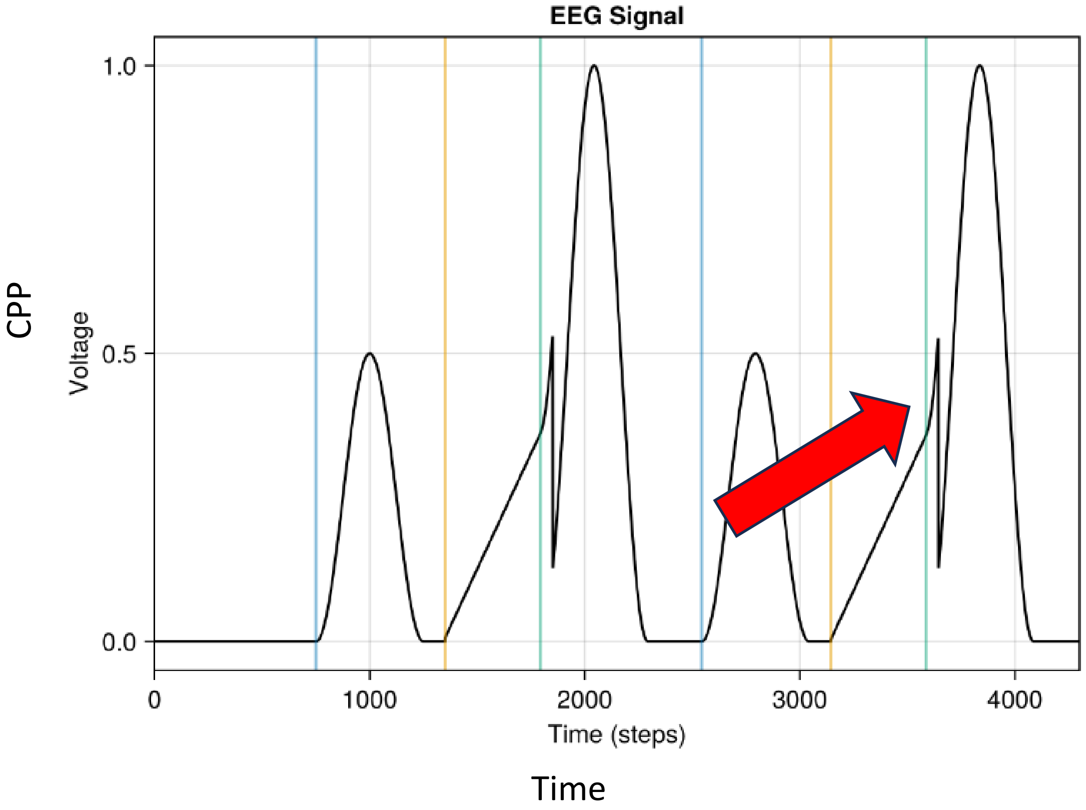


Frömer et al. Common neural choice signals can emerge artefactually amid multiple distinct value signals 2024 [3]

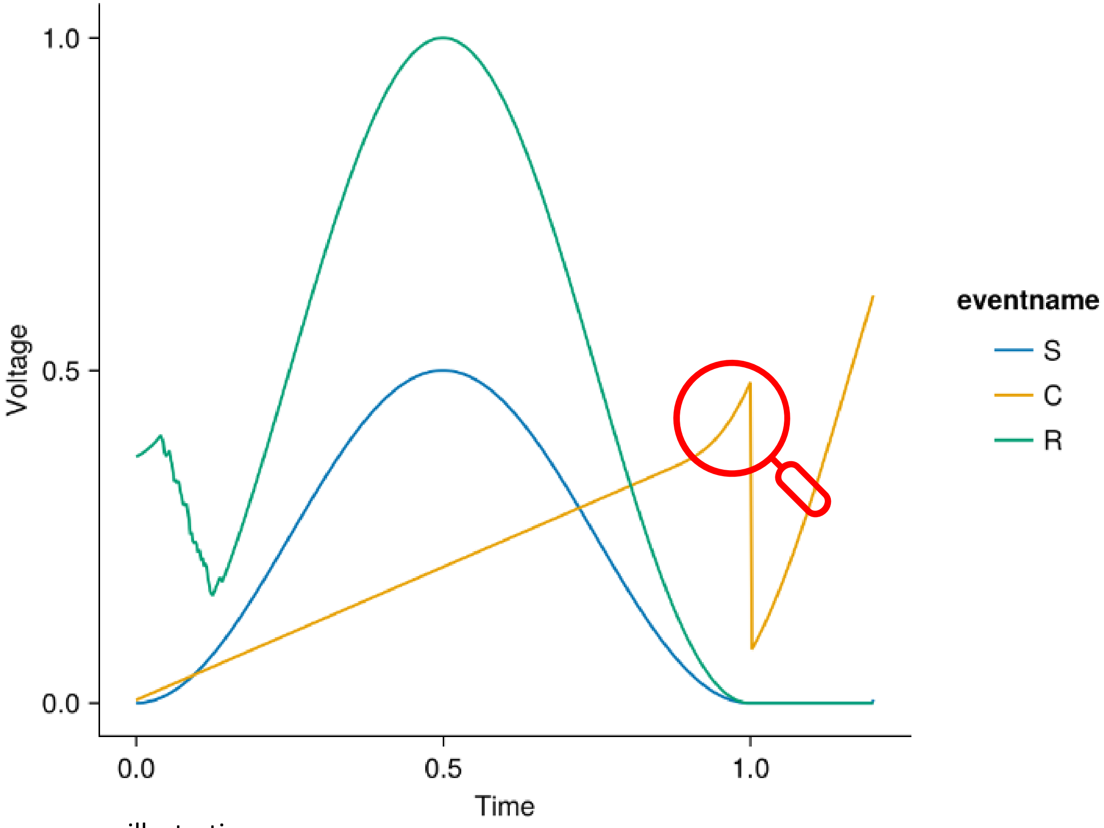
Results – Overlap Simulation - LBA

Legend

Stimulus(S) Component(C) Response(R)



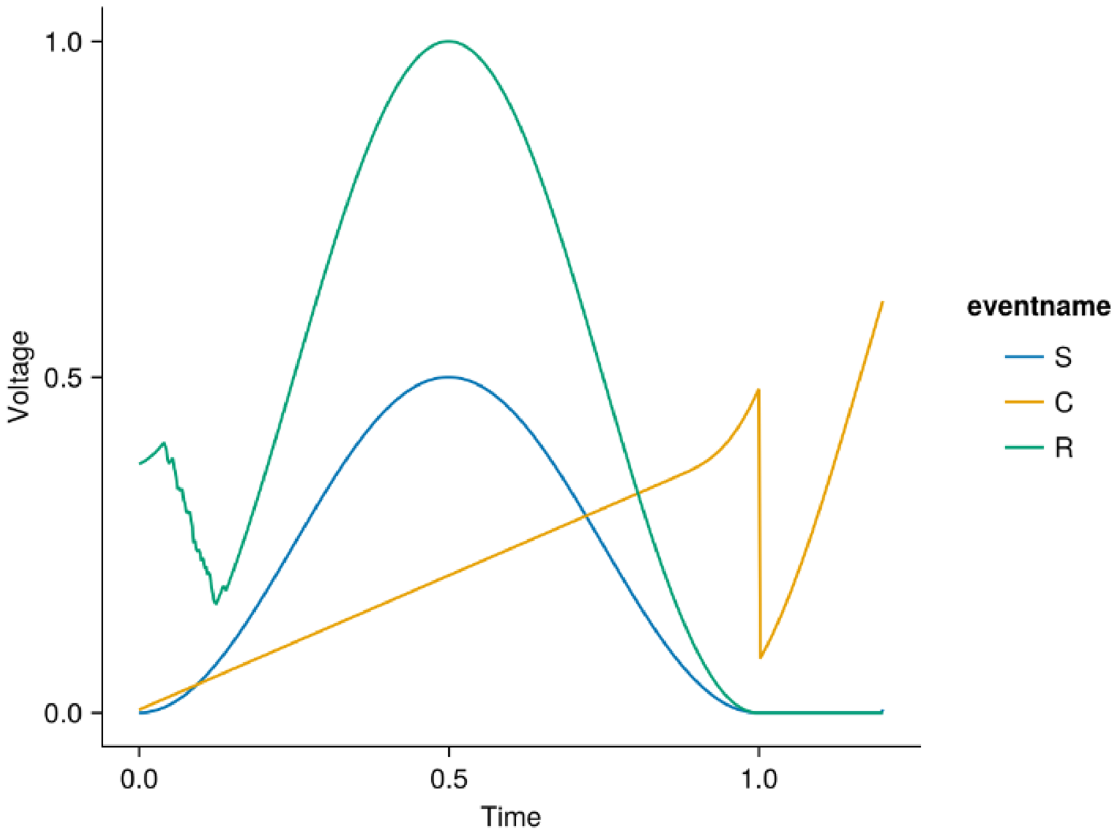
own illustration



own illustration

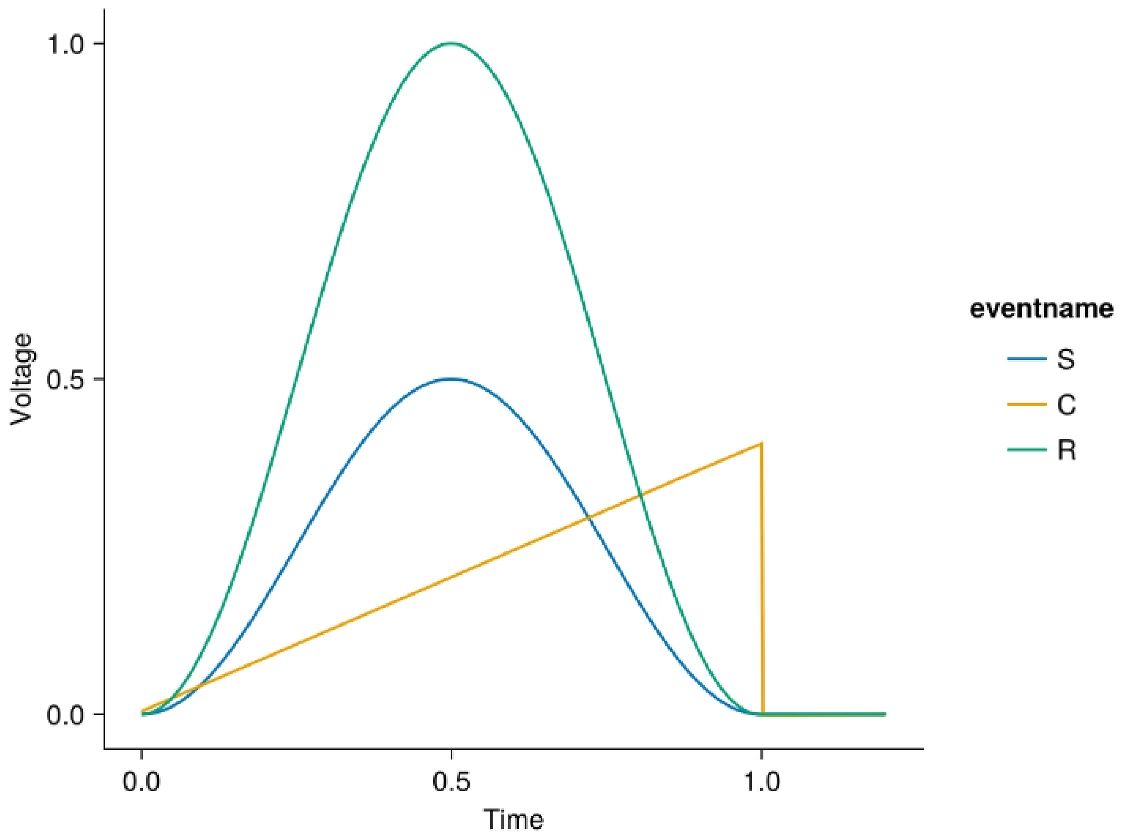
Results – Overlap Simulation - LBA

ERP



own illustration

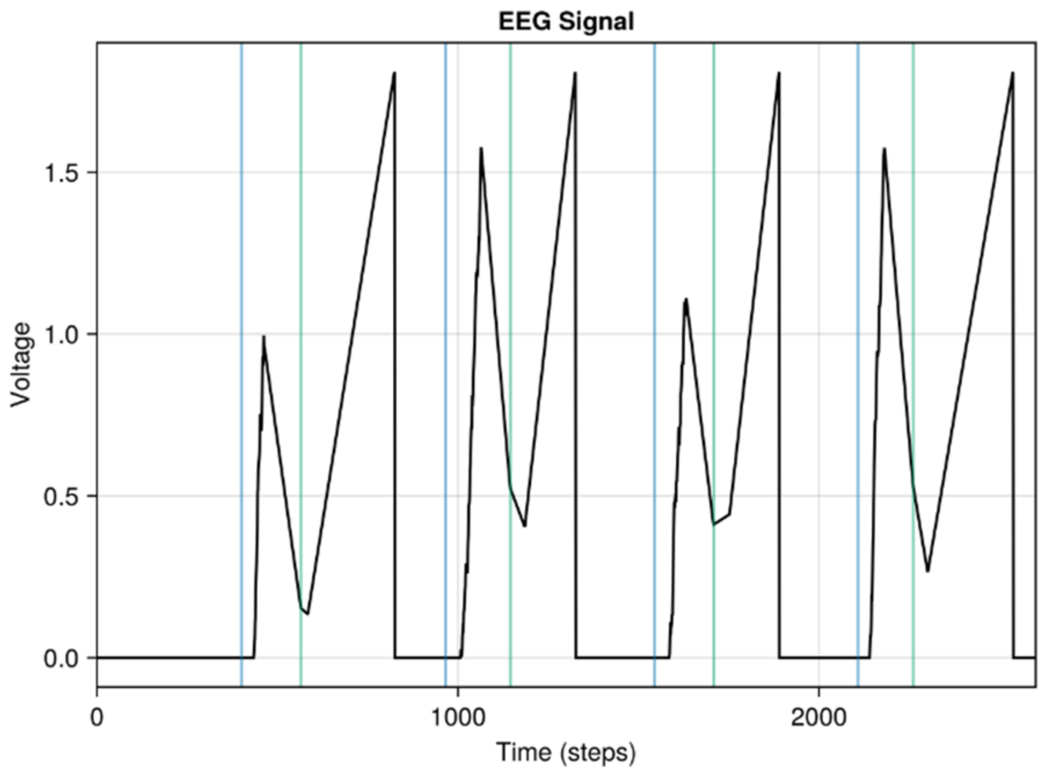
DECONV



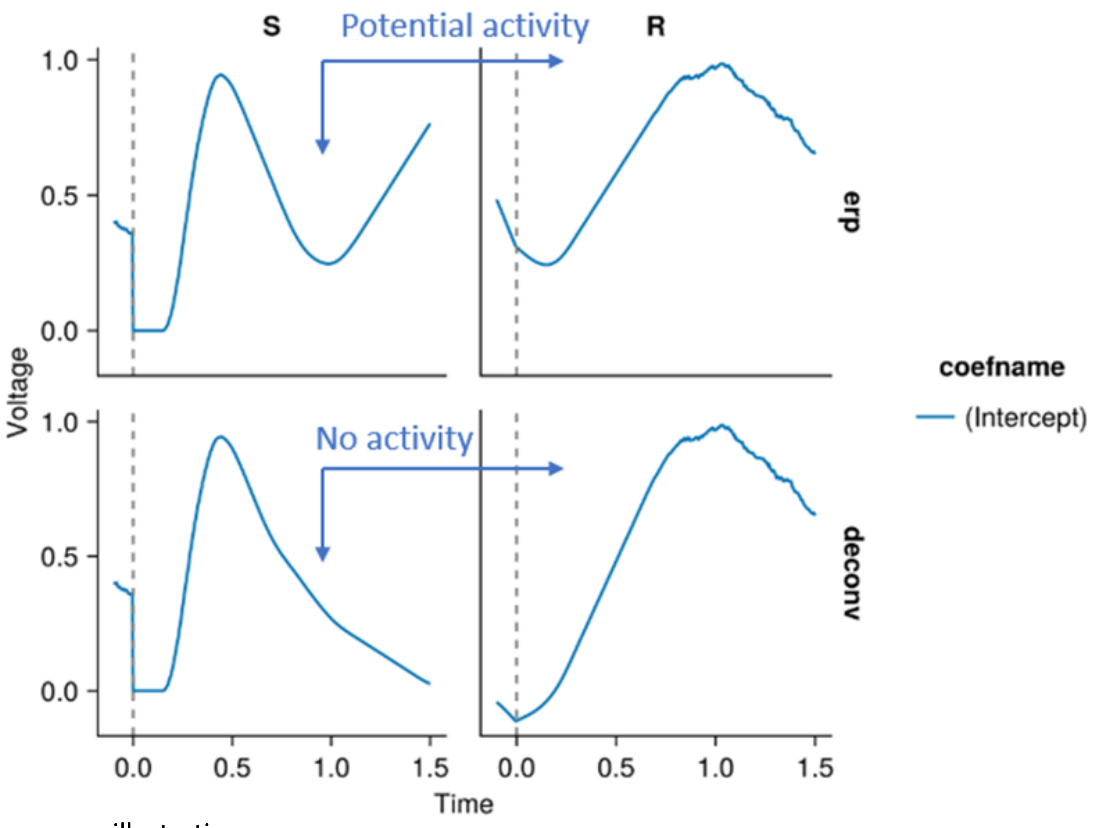
own illustration

Results – No component activity after deconv

Legend
Stimulus(S) Response(R)



own illustration



own illustration

Discussion

Main Contributions:

- **Extension** for **decision-making** research
- **Model-Based** EEG **Simulation**
- Overlap & Deconvolution **Use Cases**

Limitations:

- **Biological parameter validation**
- **Integration** of other **models**
- **Deconv** could be **further investigated (RIDE)**

Possible Next Steps

- **Refining** EEG simulations & **validating** with **real EEG data**
- **Parameter optimization** study & **integration** of new **models**

Questions?

Literature

[1]: Redmond G. Connell and Simon P. Kelly. Neurophysiology of human perceptual decision making. Annual Review of Neuroscience, 44(Volume 44, 2021):495–516, 2021.

[2]: Kelly, S. P., Corbett, E. A., & O'Connell, R. G. (2021). Neurocomputational mechanisms of prior informed perceptual decisionmaking in humans. Nature Human Behaviour, 5(4), 467–481. https:// doi.org/10.1038/s4156202000 9679

[3]: Frömer, R., Nassar, M. R., Ehinger, B. V., & Shenhav, A. (2024). Common neural choice signals can emerge artefactually amid multiple distinct value signals. Nature Human Behaviour, 8(11), 2194 2208. https://doi.org/10.1038/s415620240 1971z

[4]: S. D. Brown and A. Heathcote, “The simplest complete model of choice response time: Linear ballistic accumulation,” Cognitive Psychology, vol. 57, no. 3, pp. 153–178, 2008, doi: https://doi.org/ 10.1016/j.cogpsych.2007.12.002.