Double Responses in Drift Diffusion Models

Stien Mommen, David Matischek, Jeroen Timmerman, Lexi Li, & Emma Akrong

March 29th, 2025

Evans et al. (2020)

Cognitive Psychology 121 (2020) 101292

Contents lists available at ScienceDirect

Cognitive Psychology

journal homepage: www.elsevier.com/locate/cogpsych

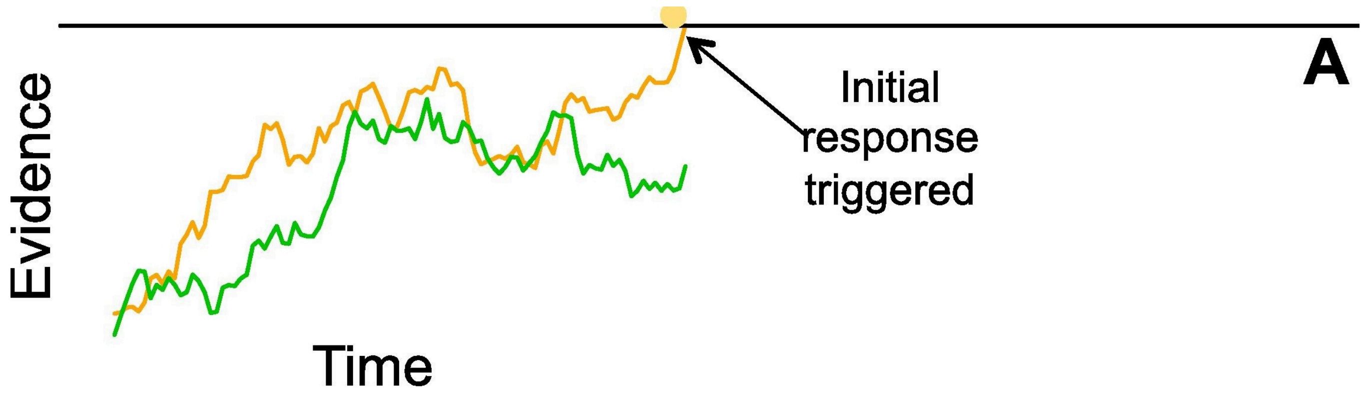
Double responding: A new constraint for models of speeded decision making

Nathan J. Evans[a,*], Gilles Dutilh[b], Eric-Jan Wagenmakers[a], Han L.J. van der Maas[a]

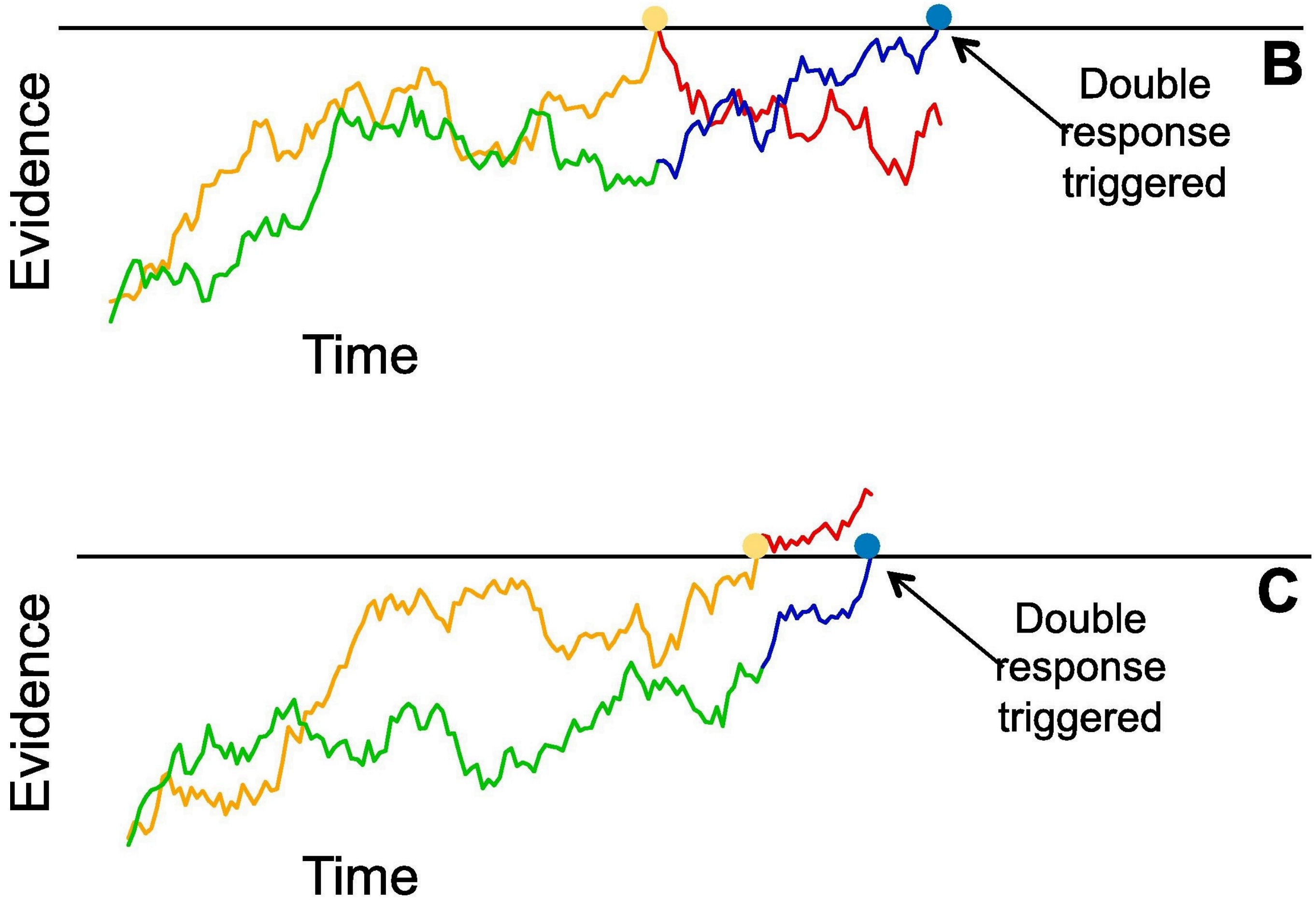
[a] *Department of Psychology, University of Amsterdam, the Netherlands*
[b] *Department of Clinical Research, University of Basel Hospital, Switzerland*

Racing Diffusion Model (RDM)

- Process of choosing between N alternatives ⟶ N racing evidence accumulators
- Assumptions:
  - One threshold level of evidence
  - Random uniform distribution of starting evidence
  - Drift rate for each alternative
  - Non-decision time



What is a double response (DR)?



Experimental Paradigm (Dutilh et al. (2009))

- Lexical decision task ⟶ word vs. non-word
- Participants: 4
- Trials: 10,000
- Conditions: Speed vs.Accuracy (between subjects design)
- DR implementation:
  - 250 ms to give second response
  - Participants were not instructed to give DRs (implicit)

Why include DR in model?

- Additional information beyond response choice and RT
- Better understanding of the decision-making process as a whole

BayesFlow

Aim

- Determine if including DR will make the model learn more
  - RQ: Does including DR improve posterior contraction

Why BayesFlow?

- MCMC more computationally costly
- BayesFlow: training is costly, but inference is fast

Observation Models

RDM (base model for all)

$$dx_i = [v_i]dt + [\sigma_i \epsilon]\sqrt{dt}$$

dx_i = change in evidence

v_i = drift rate of choice i

dt = difference in time

σ_i = scale of within-trial noise for choice i

ϵ = random variable

Observation Models

Feed-forward inhibition (FFI)

- Future evidence accumulation is reduced based on the accumulation rate of the competing alternative

$$dx_i = [v_i - \beta \sum_{j \neq i}^{n} v_j]dt + [\sigma_i \epsilon - \beta \sum_{j \neq i}^{n} \sigma_j \epsilon]\sqrt{dt}$$

β = amount of inhibition

dx_i = change in evidence

v_i = drift rate of choice i

dt = difference in time

σ_i = scale of within-trial noise for choice i, ϵ = random variable

Observation Models

Leaky-competing accumulator (LCA)

- Lateral inhibition with leakage + evidence cannot be negative
- Leakage: rate at which the alternative's accumulated evidence is reduced

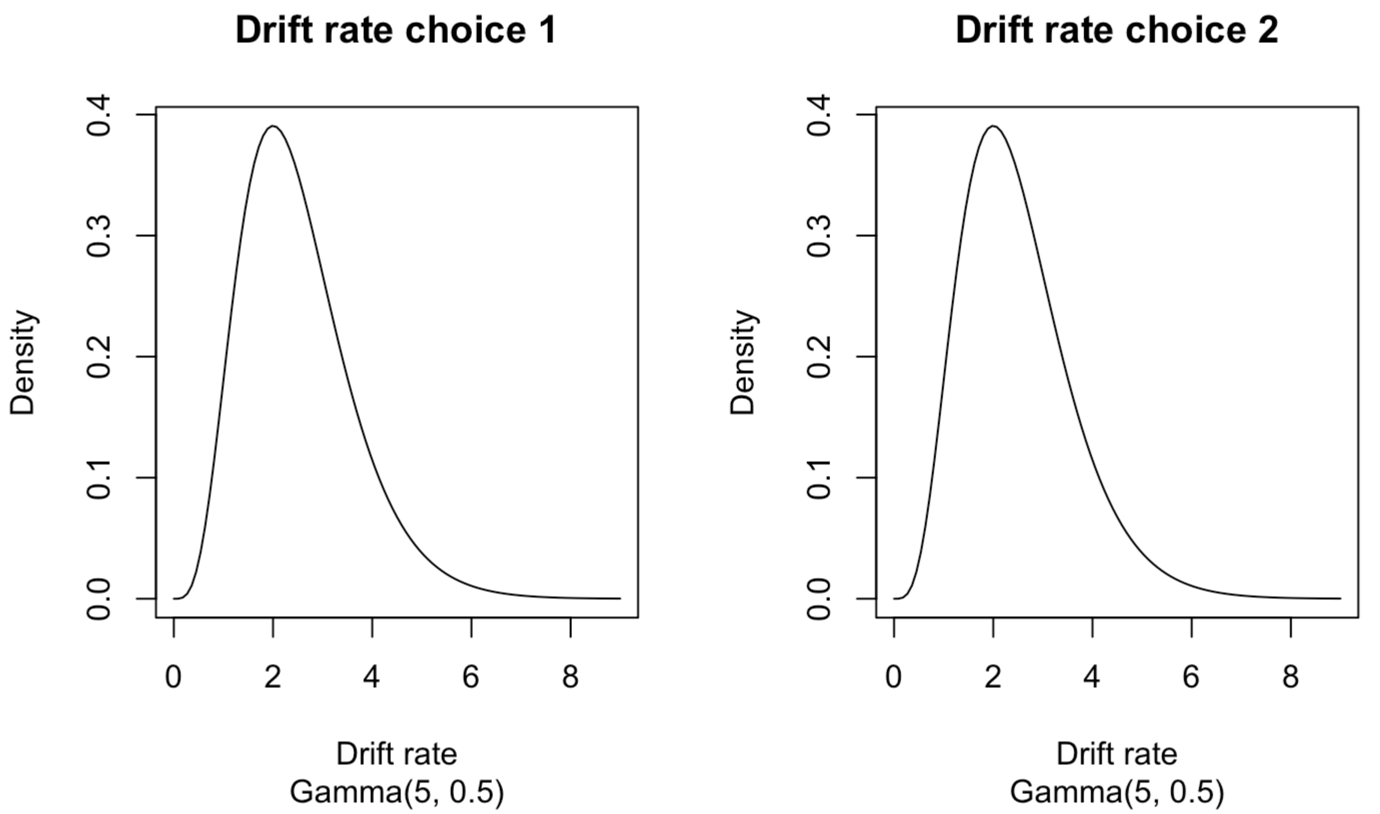
$$dx_i = [v_i - \lambda x_i - \beta \sum_{j \neq i}^{N} x_j]dt + [\sigma_i \epsilon]\sqrt{dt}$$

where $x > 0$
λ = leakage rate

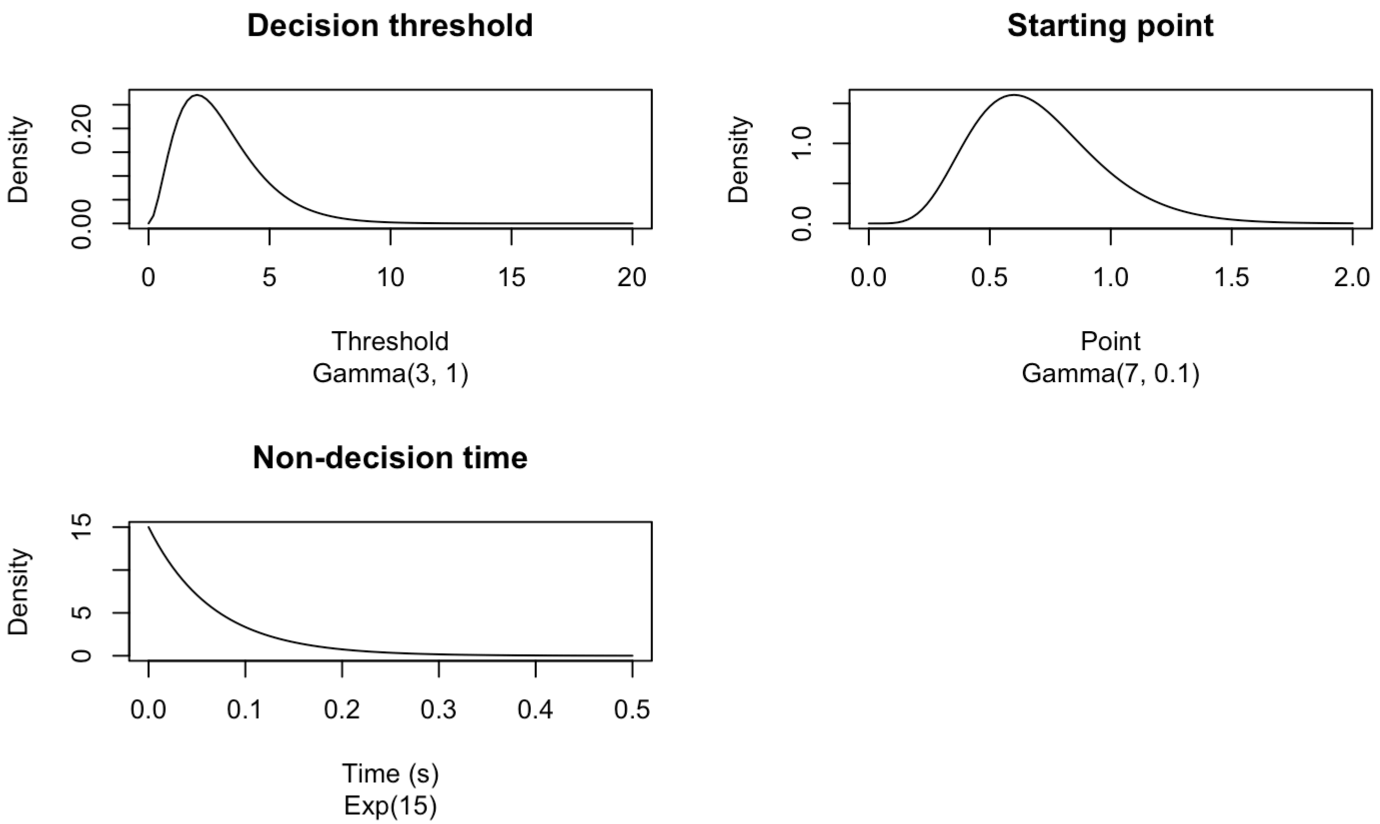
β = amount of inhibition

dx_i = change in evidence, v_i = drift rate of choice i, dt = difference in time, σ_i = scale of within-trial noise for choice i, ϵ = random variable

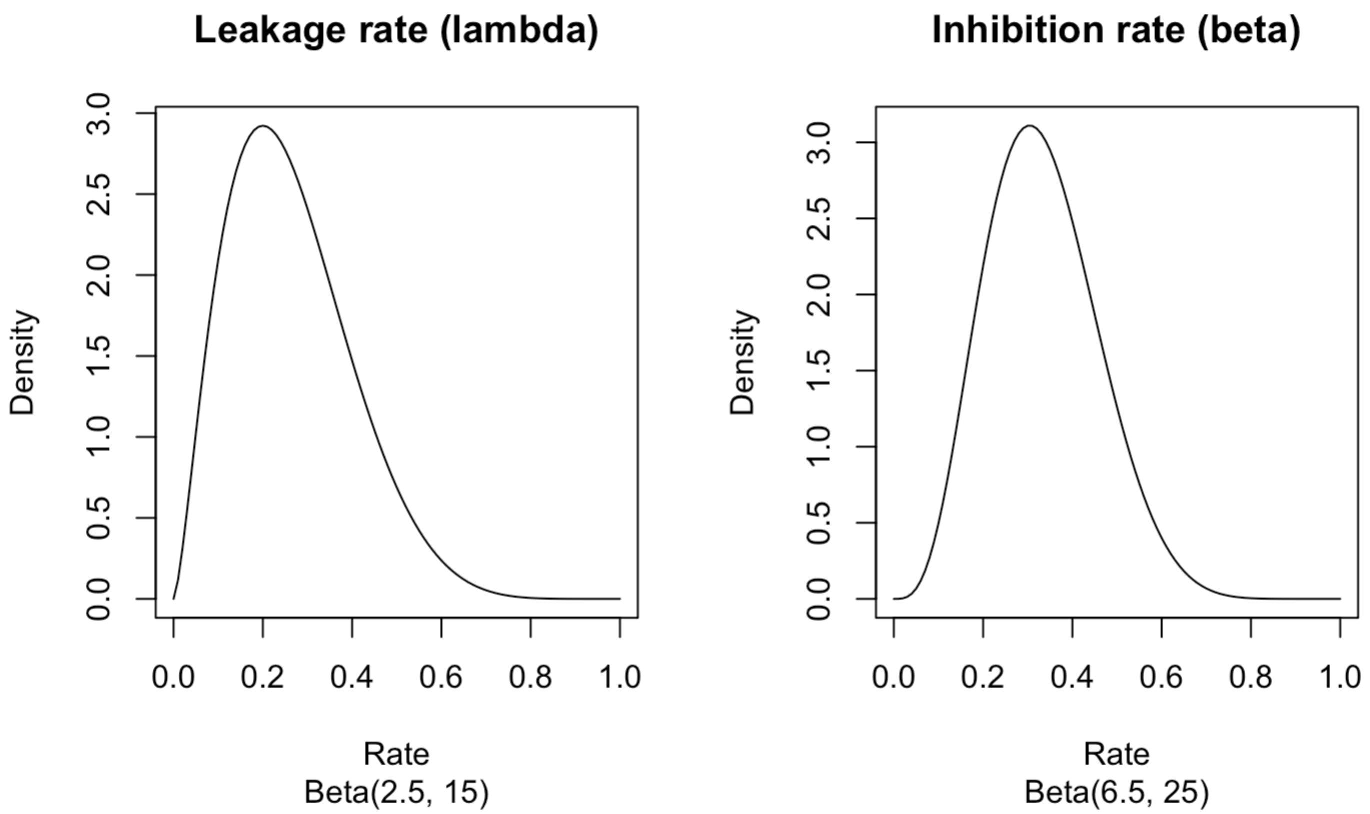
Priors


Drift rate choice 1
Density
0.0
0.1
0.2
0.3
0.4
0
2
4
6
8
Drift rate
Gamma(5, 0.5)
Drift rate choice 2
Density
0.0
0.1
0.2
0.3
0.4
0
2
4
6
8
Drift rate
Gamma(5, 0.5)

Priors


Decision threshold
Density
0.00
0.20
0
5
10
15
20
Threshold
Gamma(3, 1)
Starting point
Density
0.0
1.0
0.0
0.5
1.0
1.5
2.0
Point
Gamma(7, 0.1)
Non-decision time
Density
0
5
15
0.0
0.1
0.2
0.3
0.4
0.5
Time (s)
Exp(15)

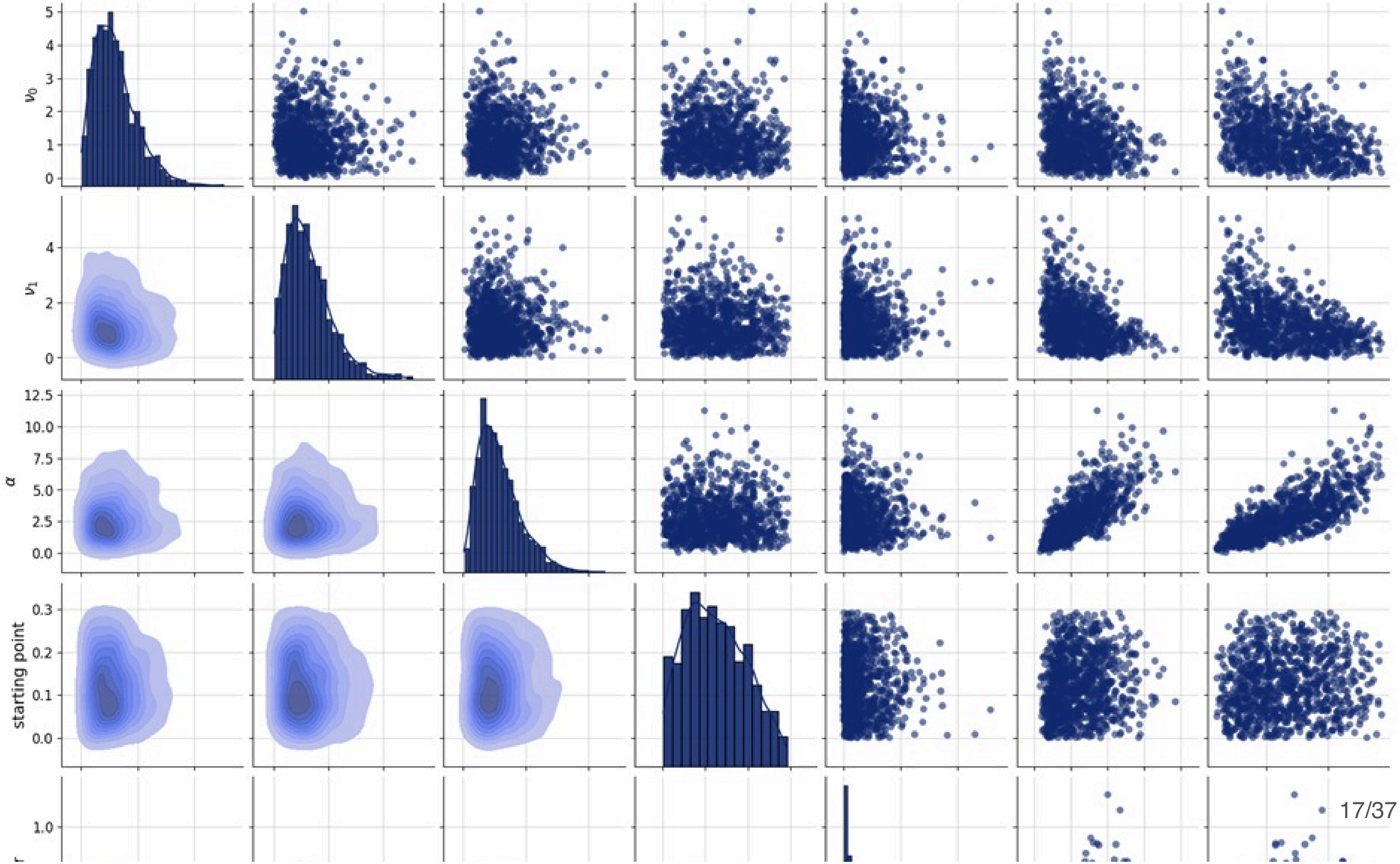
Priors


Leakage rate (lambda)
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.2
0.4
0.6
0.8
1.0
Rate
Beta(2.5, 15)
Inhibition rate (beta)
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.2
0.4
0.6
0.8
1.0
Rate
Beta(6.5, 25)

Implementation in python (RDM)

```
@nb.jit(nopython=True, cache=True)
def trial(drift, starting_point, boundary, ndt, max_t, max_drt=0.25, s=1, dt=None):
    drift = np.asarray(drift, dtype=np.float64)  # Convert before passing to JIT
    response = -1
    rt = -1

    if dt is None:
        dt = max_t / 10_000.0  # Ensure float division

    t = 0
    start = float(starting_point)  # Ensure float type
    evidence = np.random.uniform(0, start, size=len(drift))

    boundary += start  # Adjust boundary based on start
    dr = False  # Initialize double response

    # Initial evidence accumulation
    while np.all(evidence < boundary) and t < max_t:
        for resp in range(len(drift)):  # Normal loop (prange if parallel)
            evidence[resp] += dt * drift[resp] + np.random.normal(0, np.sqrt(s**2 * dt))
        t += dt
```

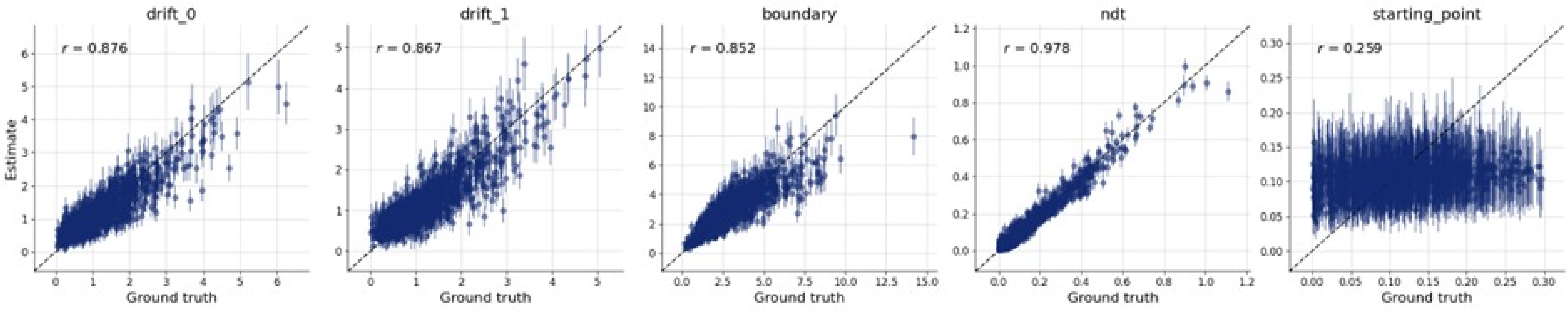
Diagnostics (RDM)



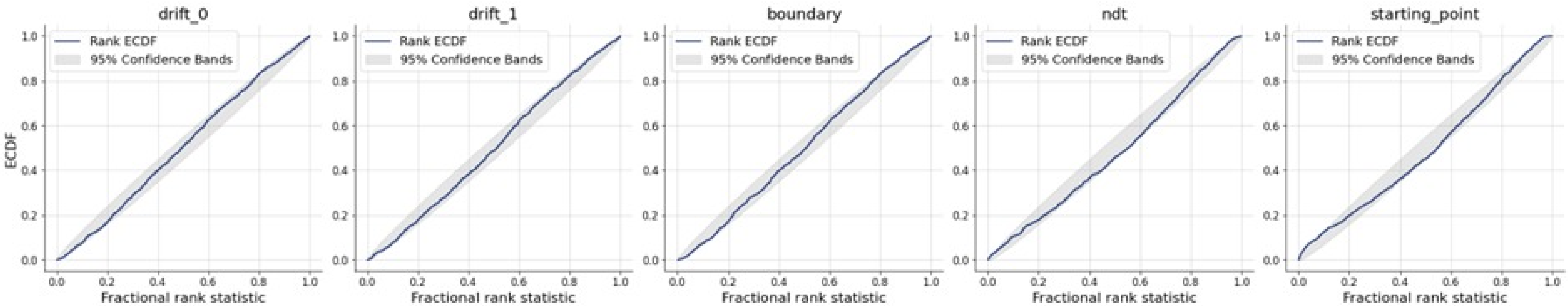
Results (RDM)

Loss Trajectory
Training
Validation
Value
Training epoch #
0
20
40
60
80
100
120
140
0
1
2
3
4
5
6
7

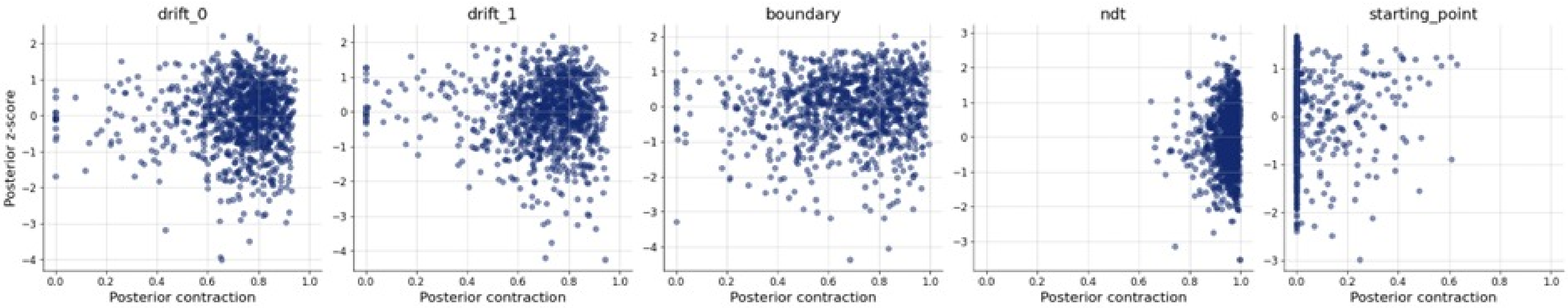
Results (RDM)


drift_0
r = 0.876
Estimate
Ground truth
drift_1
r = 0.867
Ground truth
boundary
r = 0.852
Ground truth
ndt
r = 0.978
Ground truth
starting_point
r = 0.259
Ground truth

Results (RDM)

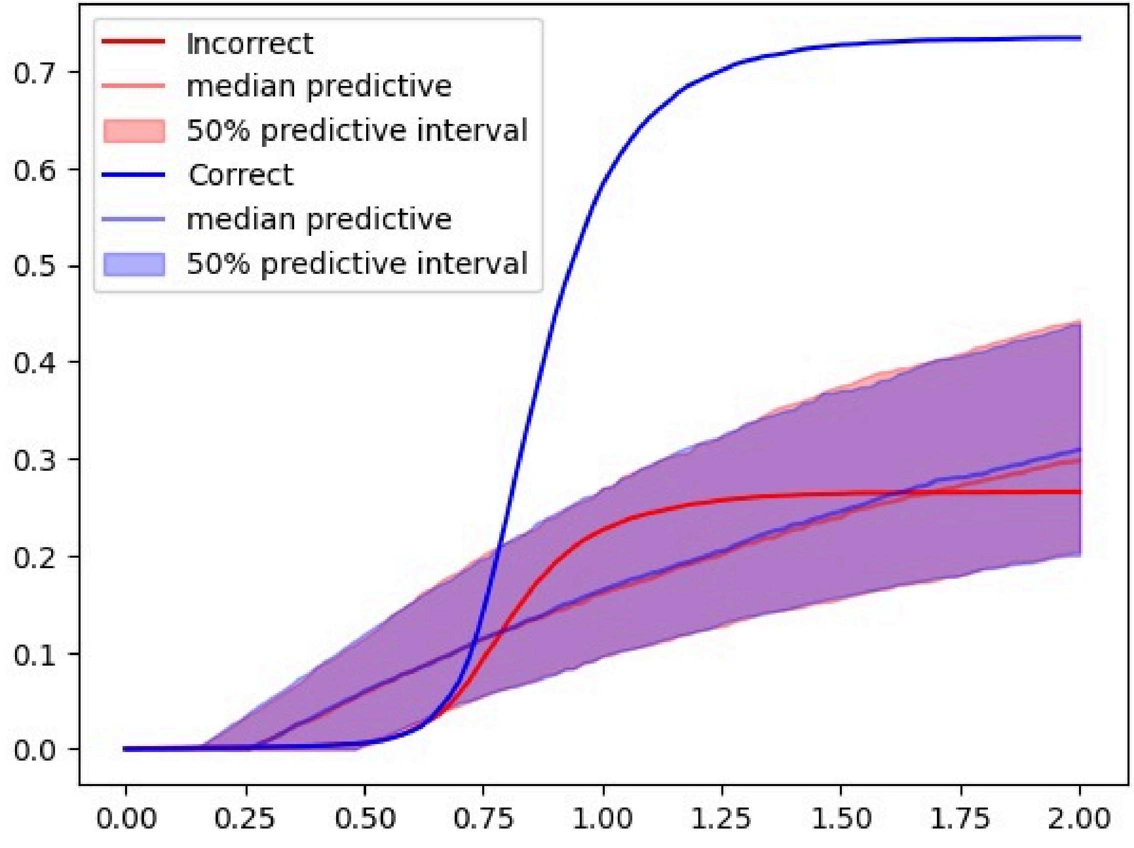

drift_0
drift_1
boundary
ndt
starting_point
Rank ECDF
95% Confidence Bands
ECDF
Fractional rank statistic
0.0
0.2
0.4
0.6
0.8
1.0

Results (RDM)



drift_0
drift_1
boundary
ndt
starting_point
Posterior z-score
Posterior contraction
Posterior contraction
Posterior contraction
Posterior contraction
Posterior contraction

Results (RDM)

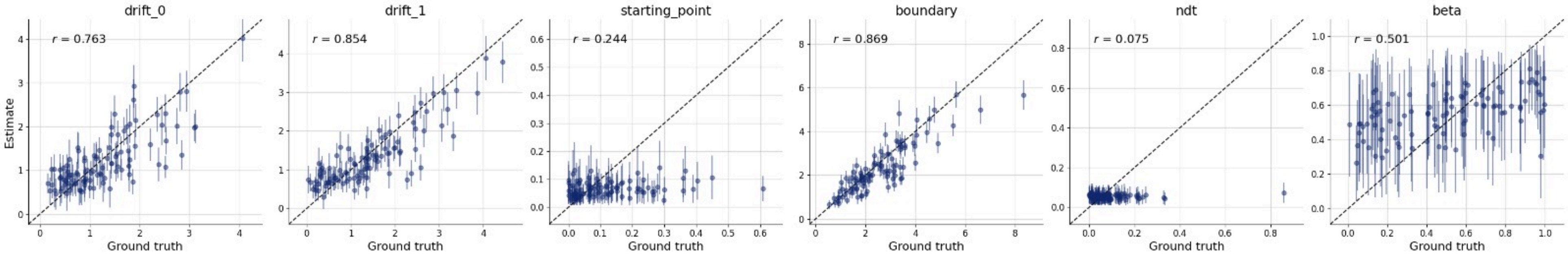


Incorrect
median predictive
50% predictive interval
Correct
median predictive
50% predictive interval
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

Results (FFI)

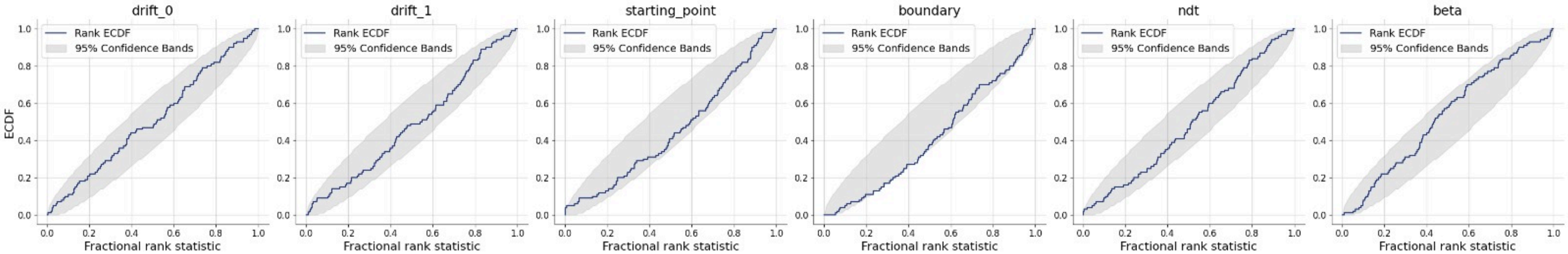
Loss Trajectory
Training
Validation
Value
Training epoch #
10
9
8
7
6
5
0
10
20
30
40
50

Results (FFI)



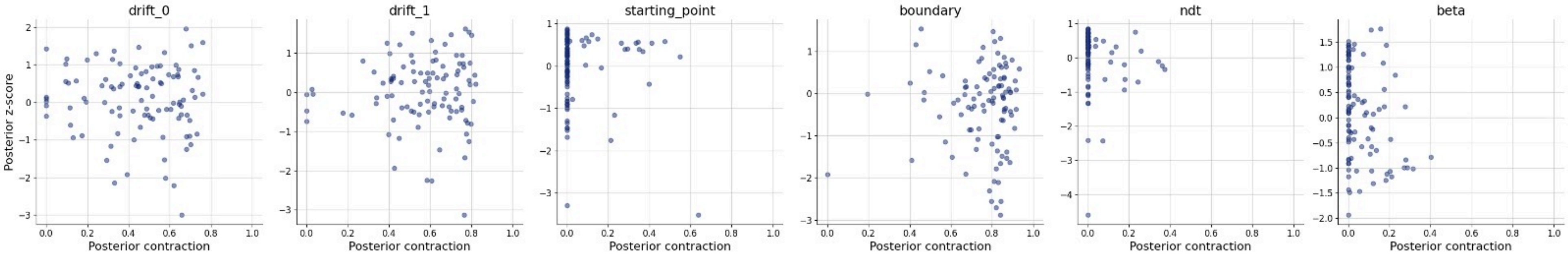
drift_0
r = 0.763
drift_1
r = 0.854
starting_point
r = 0.244
boundary
r = 0.869
ndt
r = 0.075
beta
r = 0.501
Estimate
Ground truth

Results (FFI)



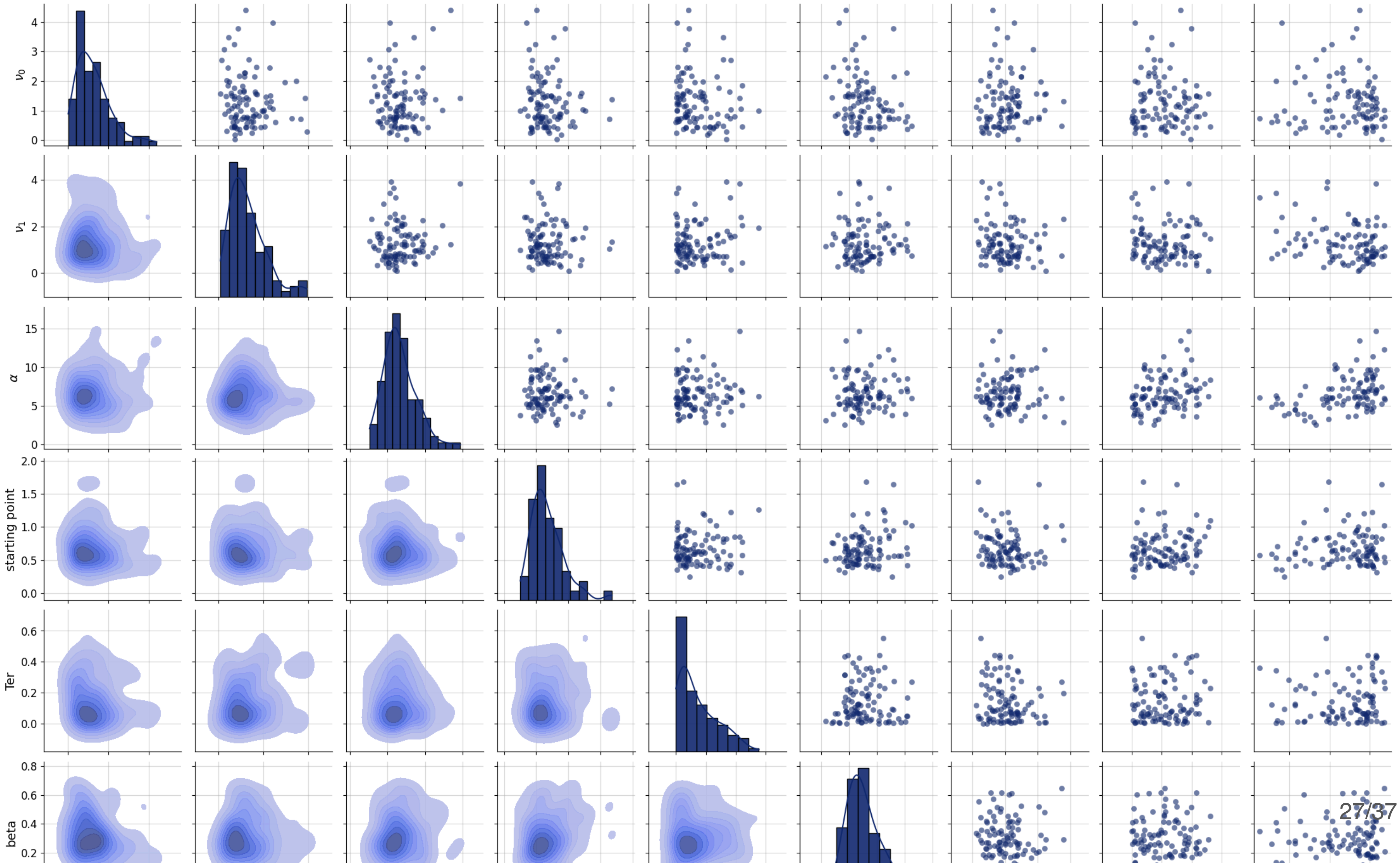
drift_0
drift_1
starting_point
boundary
ndt
beta
Rank ECDF
95% Confidence Bands
ECDF
Fractional rank statistic

Results (FFI)



drift_0
drift_1
starting_point
boundary
ndt
beta
Posterior z-score
Posterior contraction

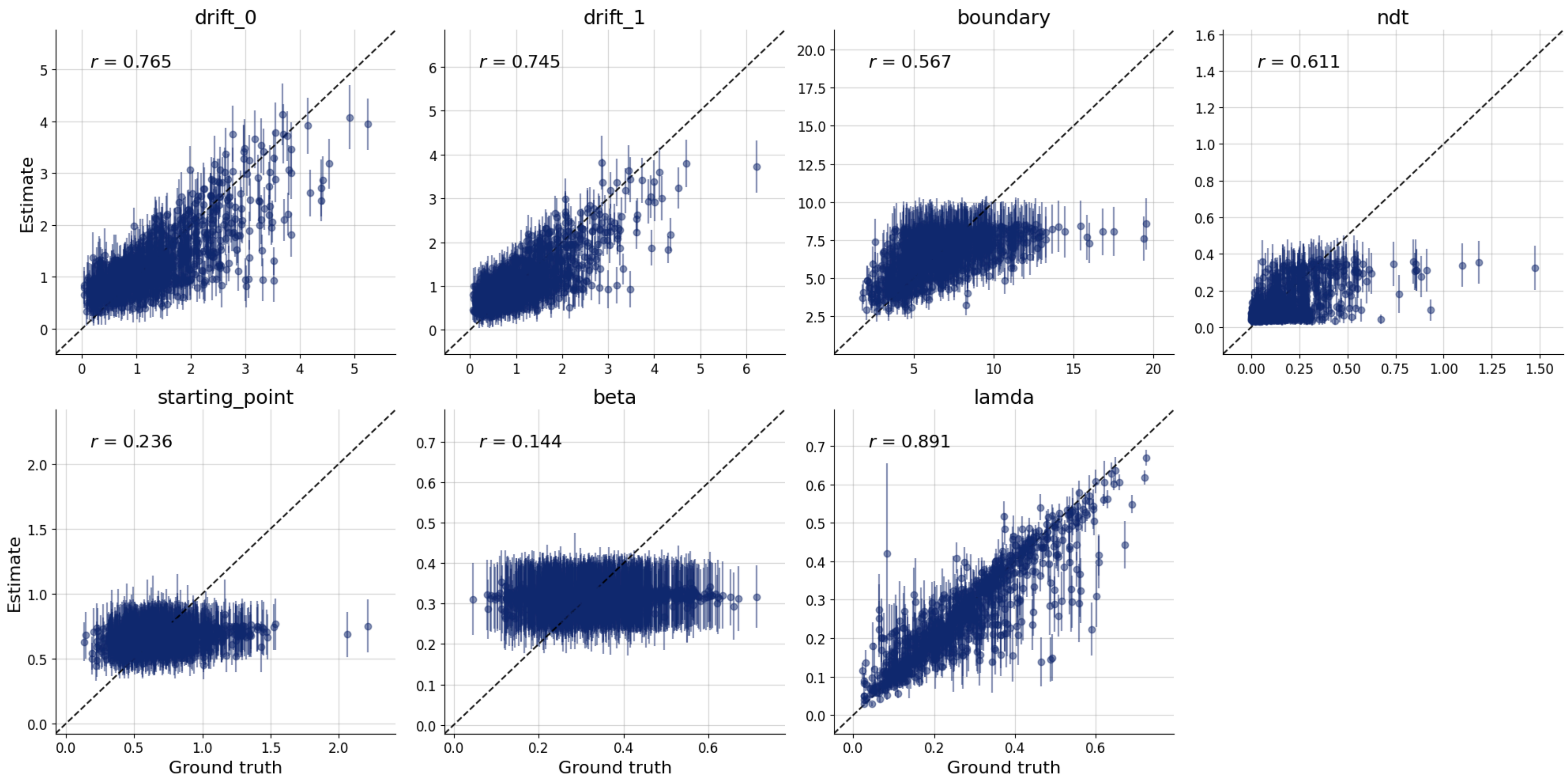
Diagnostics (LCA)



Results (LCA)

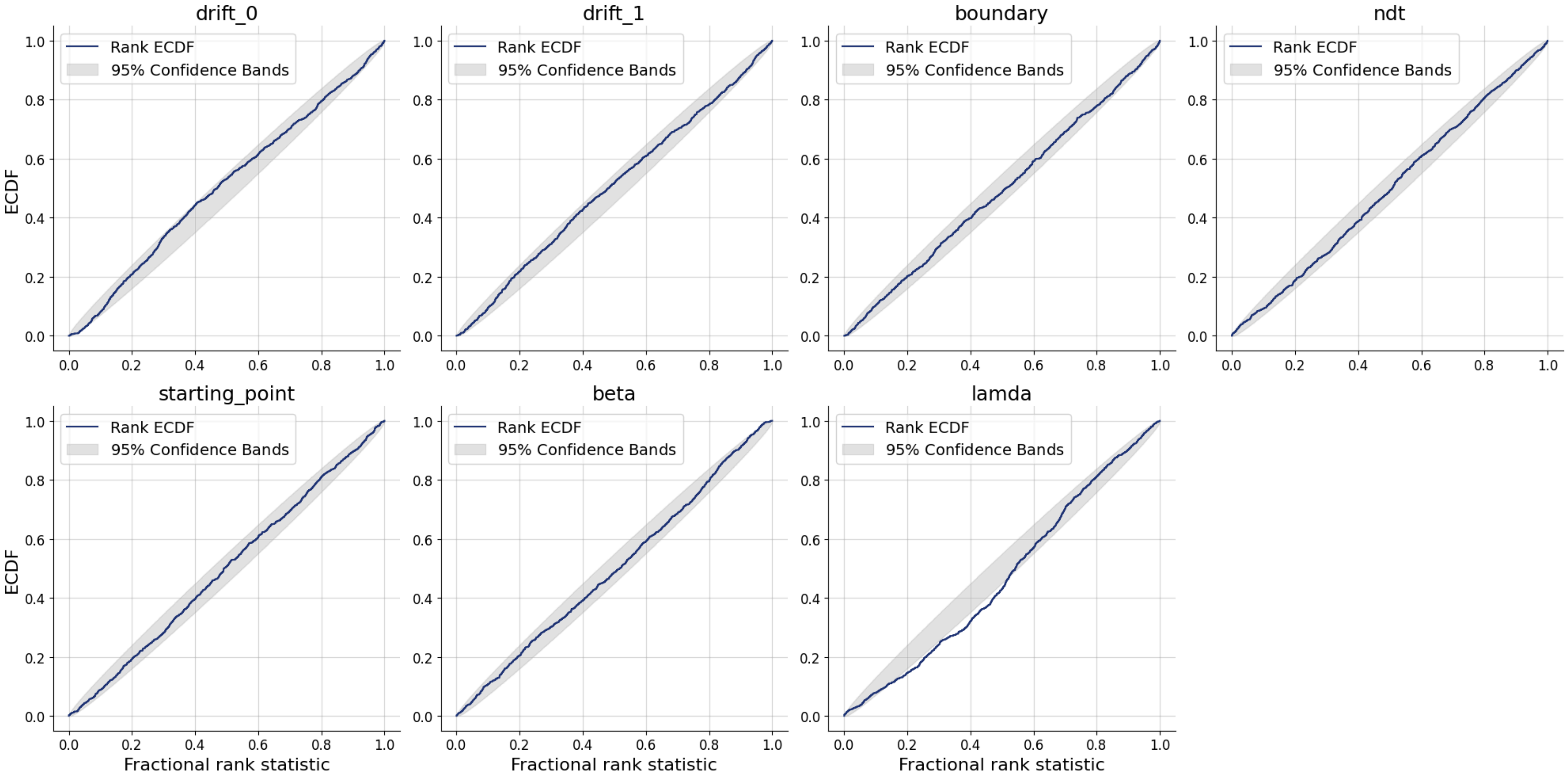
Loss Trajectory
Training
Validation
Value
Training epoch #
9
8
7
6
5
4
0
10
20
30
40
50

Results (LCA)

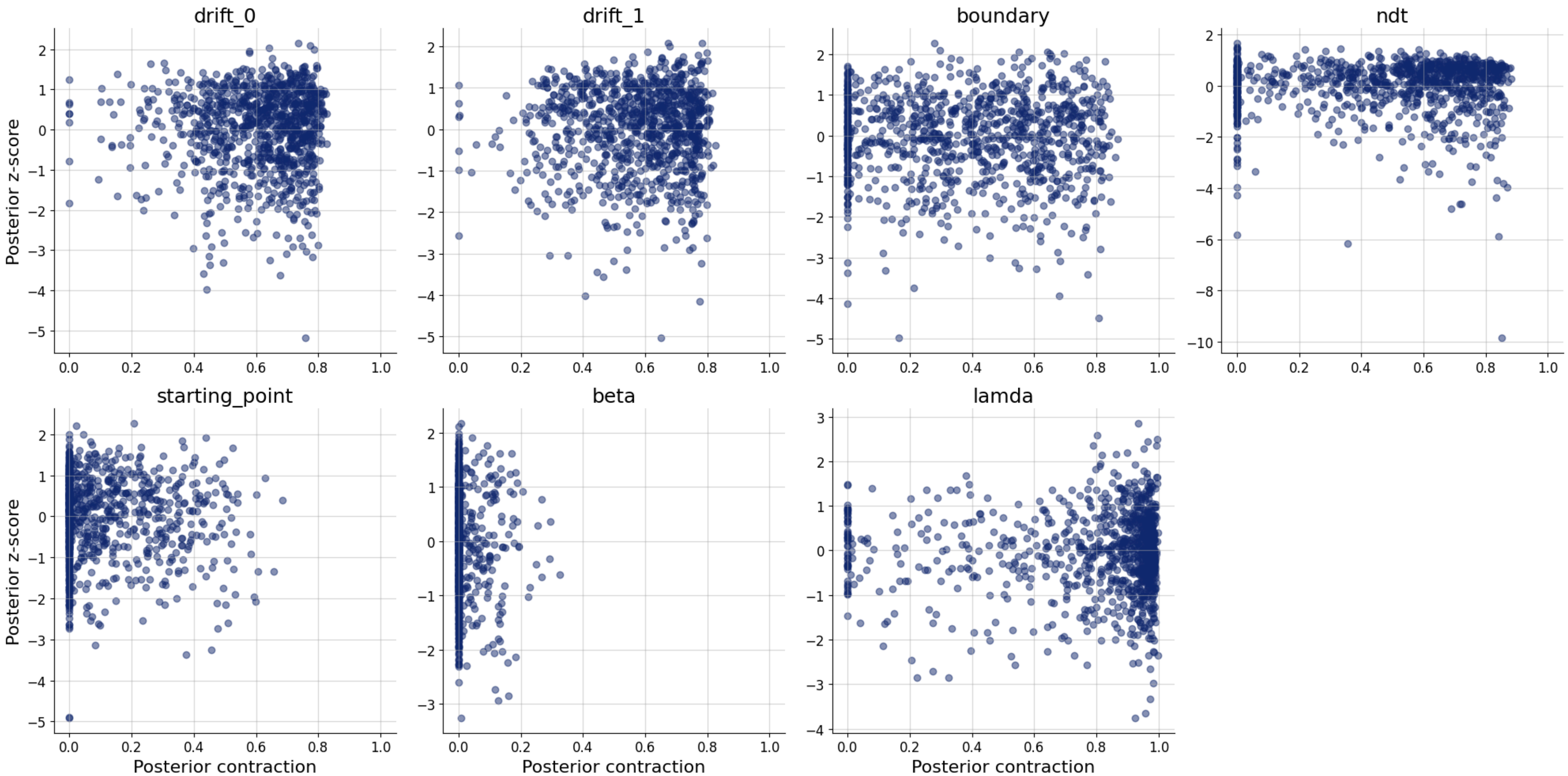


drift_0
r = 0.765
drift_1
r = 0.745
boundary
r = 0.567
ndt
r = 0.611
starting_point
r = 0.236
beta
r = 0.144
lamda
r = 0.891
Estimate
Ground truth

Results (LCA)

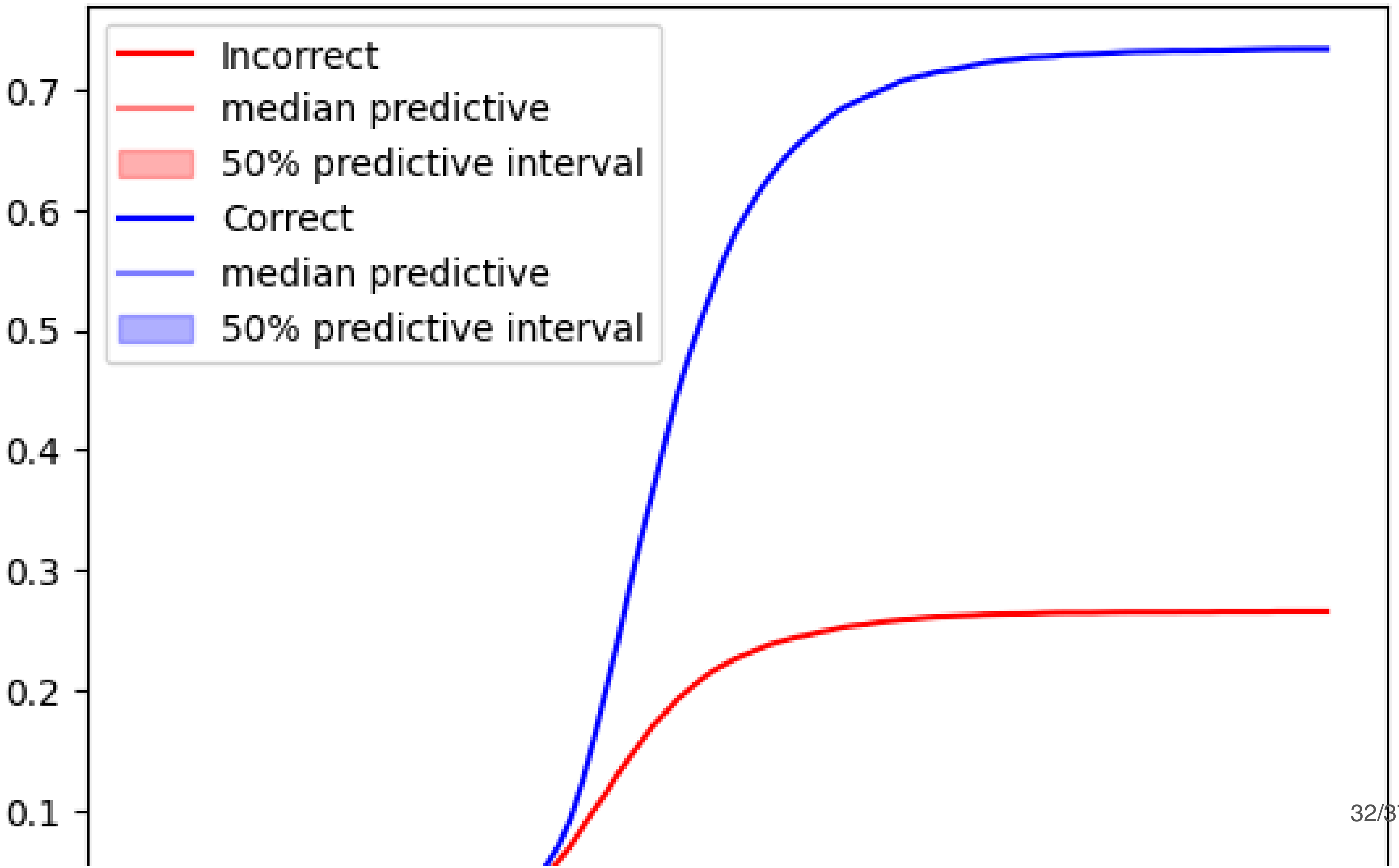

drift_0
drift_1
boundary
ndt
starting_point
beta
lamda
Rank ECDF
95% Confidence Bands
ECDF
Fractional rank statistic
0.0
0.2
0.4
0.6
0.8
1.0

Results (LCA)



drift_0
drift_1
boundary
ndt
starting_point
beta
lamda
Posterior z-score
Posterior contraction

Results (LCA)

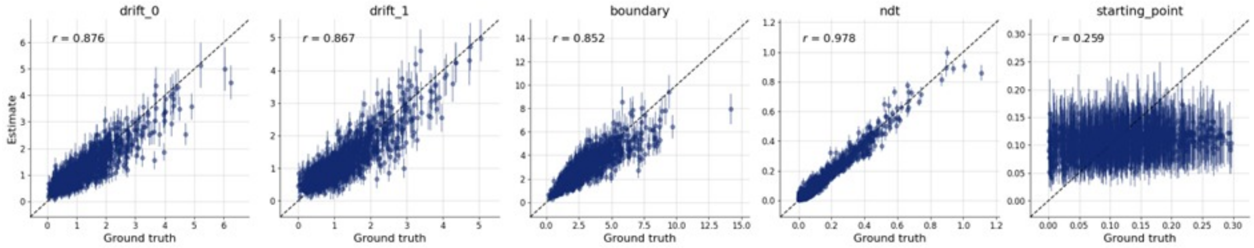


Incorrect
median predictive
50% predictive interval
Correct
median predictive
50% predictive interval
0.7
0.6
0.5
0.4
0.3
0.2
0.1

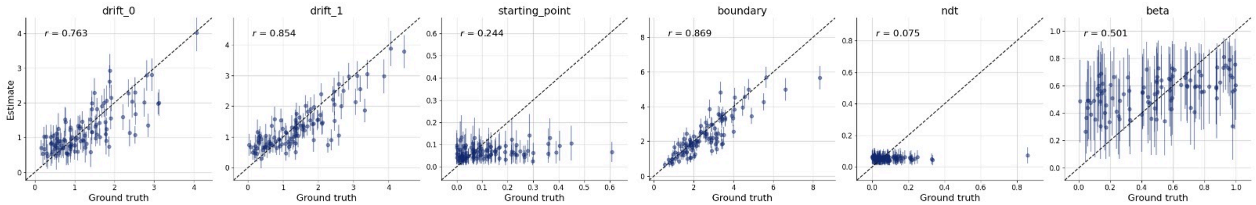
Key Findings

Adding double response did kind of lead to the model to learn more

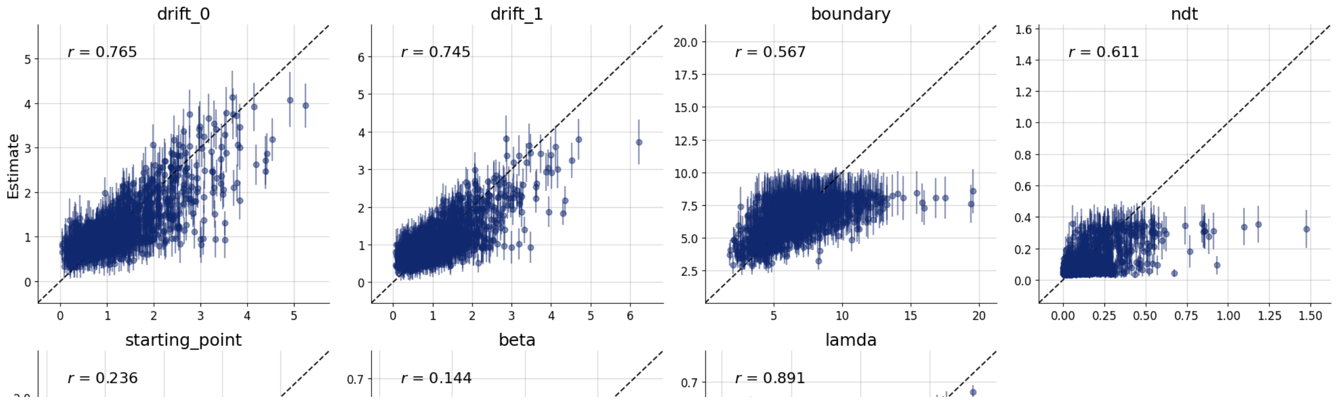
RDM



FFI



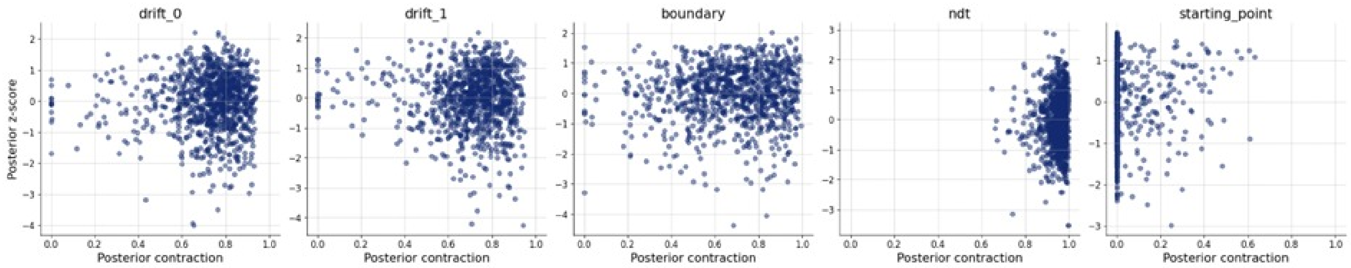
LCA



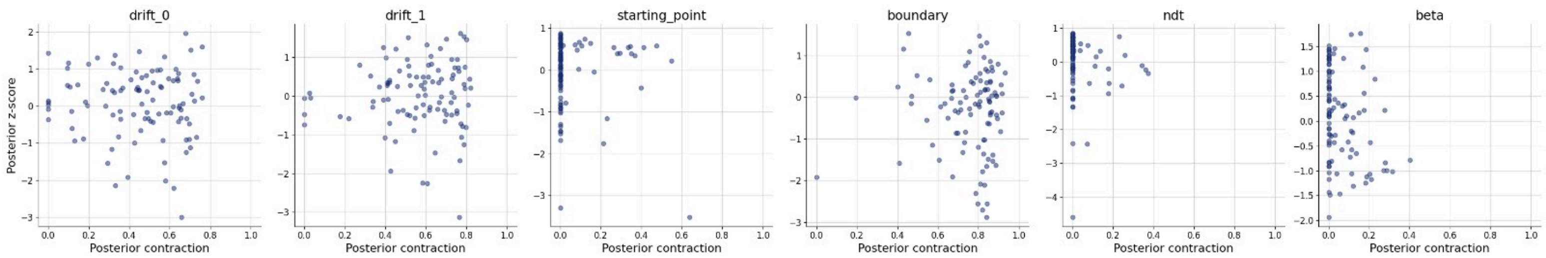
Key Findings

It also kind of improved the posterior contraction

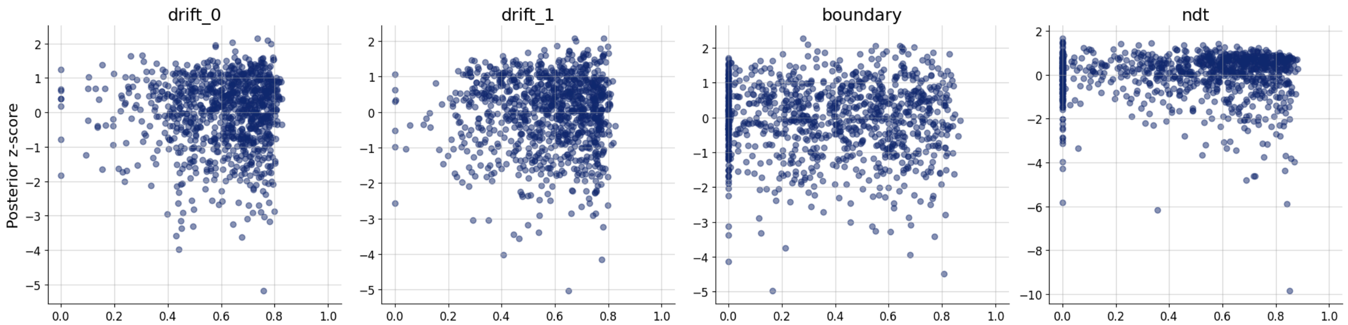
RDM



FFI



LCA



Limitations of our model

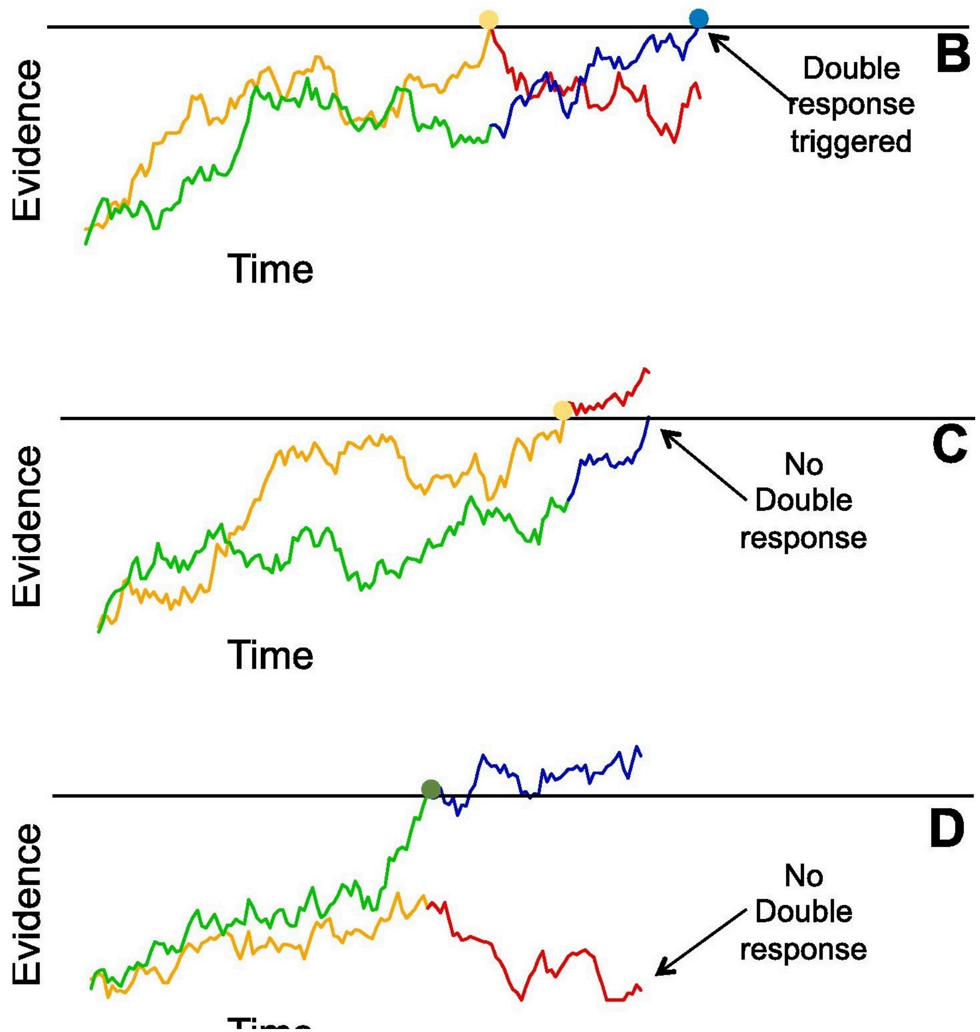
- Priors can be more grounded in theory
- More participants, more double response

Accuracy Participant 2

	Initial Response	First RT	Double Response	Double Response RT
6661	FALSE	0.4852	TRUE	0.0458
7372	FALSE	0.4194	TRUE	0.0540
9323	FALSE	0.4851	TRUE	0.0700

Directions for Future Research

- Study design ⟶ explicit double responses
- Alternative definition of double response ⟶ loser drift over takes winner drift



Thanks for listening <3