

## — Project information —

Please read the instructions below carefully. They will answer most of the questions you might have about the project.

## Schedule

- **From now on:** Start forming groups and choose topics from the list we provide (see below).
- **June 26:** During lecture: Opportunity for open questions about topics, groups, etc.
- **June 29 (midnight):** Deadline for final formation of groups and project registration via email.
- **In July:** During both lecture and exercise sessions: Opportunity to ask questions and discuss your project with the teaching staff.
- **July 14 (midnight):** Deadline for submission of the presentation slides (via email).
- **July 15 and 17:** Presentations, depending on number of presentation additional dates in this week.
- **August 17 (midnight):** Deadline for report submission (via email).

## Guidelines

**Groups** The project is meant to be done in groups of 2. It is possible to do it in groups of 1 or 3, however the scope for a 3 person group will be increased accordingly, while the scope for a 1 person group will be the same as for 2. While we allow groups of different sizes to accommodate your needs, we strongly encourage groups of 2.

**Previous Courses** Without previous experience in data analysis projects and reports, this project work may be very hard. Thus, having previously taken courses such as Introductory Case Studies is highly recommended (but not mandatory).

**Registration** Please form your groups and send an email to Aayush ([aayush.mishra@tu-dortmund.de](mailto:aayush.mishra@tu-dortmund.de)) containing the following information:

- Names, matriculation numbers, study programmes, and module (applied or methods; if applicable) of all group members.
- Preferred topic(s) for your project (see below).
- If you find your own project: The name of the teaching staff who has approved your project (see below).

You will receive a confirmation email with your assigned group number once your registration is processed.

Submission of presentation slides and the final report should also be directed to Aayush ([aayush.mishra@tu-dortmund.de](mailto:aayush.mishra@tu-dortmund.de)).

## Topics

To find a topic, you can do one of the two following options

- Select *up to three topics* of your preference from the list provided by us below. We will do our best to assign you a topic that you prefer, but if there is too much overlap with other groups, we may assign you to a project you did not select to avoid excessive repetition of the topics.
- Come up with your own topic. The topic needs to be selected such that using BayesFlow for ABI is reasonable. The topic must be discussed and **approved** by the teaching staff beforehand. To initiate the approval process, please talk to any member of the teaching staff. Note that if you chose to discuss this by email, it may take a couple of days to get your idea approved (depending on the amount of discussion needed).

**Topics** Here is the list of topics that you can choose from (unless you come up with a topic of your own). You can consult each project with its respective lecturer indicated below.

### 1. Double Responses in Decision Tasks

*Lecturer: Simon* ([simon.kucharsky@tu-dortmund.de](mailto:simon.kucharsky@tu-dortmund.de))

In many cognitive experiments, participants are asked to make quick decisions under time pressure, such as identifying a stimulus, choosing between alternatives, or categorizing an input. These are known as *speeded decision tasks*, typically provided data in a form of choices and response times. To analyze such data, researchers often turn to *evidence accumulation models* (EAMs), which offer a principled statistical framework for modeling how decisions unfold over time.

A persistent phenomenon observed during speeded decision tasks is *double responding*. For example, in a lexical decision task where participants judge whether a letter string is a word or non-word, a participant might first press a key indicating “word” and then quickly follow it with

another press indicating “non-word.” These double responses are often excluded during data processing or even not recorded at all. However, they may actually reflect moments of uncertainty or conflict in the decision process and could contain informative statistical signals.

In this project, you will explore how incorporating double responses into an EAM affects inference and model fit. Choose a model from Evans et al. (2020) that extends an EAM (such as the RDM) to account for double responses. Implement the model using **BayesFlow**. Estimate the parameters on data from Evans et al. (2020).

Evans, N. J., Dutilh, G., Wagenmakers, E.-J., & van der Maas, H. L. (2020). Double responding: A new constraint for models of speeded decision making. *Cognitive Psychology*, 121, 101292. <https://doi.org/10.1016/j.cogpsych.2020.101292>

## 2. Collapsing Decision Boundaries

Lecturer: Simon (simon.kucharsky@tu-dortmund.de)

In many cognitive experiments, participants are asked to make quick decisions, such as identifying a stimulus, choosing between alternatives, or categorizing an input. These are known as *speeded decision tasks*, and they provide data in the form of both choices and response times. A key question is how people adjust their decision-making strategies when pressured to respond quickly, potentially at the cost of accuracy.

To study these processes, researchers often use evidence accumulation models (EAMs) which simulate decisions as a process of accumulating noisy evidence over time until a threshold (“decision boundary”) is reached. One central modeling challenge is how to represent the decision boundary: should it remain fixed, collapse over time, or be modulated by a rising urgency signal?

A study by Katsimpokis et al. (2020) showed that the answer depends in part on *how time pressure is implemented* in the experiment. When participants were instructed to respond quickly, fixed boundaries seemed to fit best. However, when time pressure was enforced via an explicit deadline, *linearly collapsing boundaries* provided a better account of the data.

What remains unexplored is whether other mechanisms could improve the fit even further, such as non-linearly collapsing boundaries (e.g., exponential or sigmoid functions) or urgency gating mechanisms.

In this project, you will extend previous modeling work by implementing non-linear boundary functions or urgency gating mechanisms within an EAM of your choice. Implement an EAM model with non-linear collapsing boundaries/urgency gating mechanisms in **BayesFlow**. Fit it to empirical response time and choice data from the Katsimpokis et al. (2020) study and discuss results.

Katsimpokis, D., Hawkins, G. E., & van Maanen, L. (2020). Not all speed-accuracy trade-off manipulations have the same psychological effect. *Computational Brain & Behavior*, 3(3), 252-268. <https://doi.org/10.1007/s42113-020-00074-y>

## 3. The SWIFT Model of Eye Movements

Lecturer: Simon (simon.kucharsky@tu-dortmund.de)

In reading research, eye-tracking provides data relevant to how people process written language. One of the key questions is how readers decide when to move their eyes and where to look next. These decisions are influenced by both the cognitive state of the reader and properties of the texts such as word frequency or sentence structure. Modeling these dynamics statistically is challenging due to the complexity and variability of eye movement behavior.

One approach to modeling these processes is the *SWIFT model*, a dynamic generative model of eye movement control during reading. It simulates how a reader's gaze shifts across a sentence as they process its content. The model incorporates timing and control mechanisms for both *fixation duration* (how long the eye stays on a word) and *saccades* (eye movements to the next word). The full version of SWIFT is computationally intensive and has an intractable likelihood. However, you can work with a simplified version of the SWIFT model introduced by Engbert & Rabe (2024).

Implement the simplified SWIFT model in **BayesFlow**. Use eye-tracking data to estimate model parameters related to gaze control and reading dynamics. Investigate how well the model captures the observed fixation durations and movement patterns.

You will work with real eye-tracking data from a controlled reading experiment. One example dataset, available at <https://osf.io/teyd4>, contains fixation sequences for an individual participant. You can also link the fixation data to word properties using the corpus file from <https://osf.io/nj2mf>.

Engbert, R., & Rabe, M. M. (2024). A tutorial on Bayesian inference for dynamical modeling of eye-movement control during reading. *Journal of Mathematical Psychology*, 119, 102843. <https://doi.org/10.1016/j.jmp.2024.102843>

#### 4. Simple Model of Neuron Activation

Lecturer: Daniel (daniel.habermann@tu-dortmund.de)

Neurons are the fundamental signaling units of the nervous system. They communicate via electrical impulses, the so-called action potentials, which are triggered by the opening and closing of voltage-gated ion channels located across the neuron's membrane.

The fundamental understanding of how action potentials are generated as a function of the input current, ion conductances, as well as membrane potentials and capacitances, was laid in the 1930s and 1940s by scientist such as Alan Hodgkin and Andrew Huxley, who performed extensive experiments on giant squid axons.

In this project, the task is to implement a simple model of neuron activation, such as the leaky integrate-and-fire (LIF) model (see [1] and additional self-researched references), which is less biophysically detailed than the full Hodgkin-Huxley model, but still captures the essential elements neuron behaviour like membrane potential integration, leakage, and spiking.

Using BayesFlow, you will use simulated neuron activation trajectories to train a neural posterior density estimator and investigate how well it is possible to recover the original parameters for new (also simulated) trajectories.

References:

1. [https://compneuro.neuromatch.io/tutorials/W2D3\\_BiologicalNeuronModels/student/W2D3\\_Tutorial1.html](https://compneuro.neuromatch.io/tutorials/W2D3_BiologicalNeuronModels/student/W2D3_Tutorial1.html)

#### 5. Inference of protein secondary structure motifs

Lecturer: Daniel (daniel.habermann@tu-dortmund.de)

Proteins are long chains of amino acids that fold into specific shapes. One key level of organization is the secondary structure, where each amino acid is part of three local folding patterns (alpha-helix, beta-sheet or random coil [1]), which then further fold into three-dimensional structures, defining the function of the protein. In this project, we focus specifically on predicting alpha-helix patterns using a two-state Hidden Markov Model (HMM) [2]. The two states are "alpha-helix" and "other" (encompassing beta-sheets and coils). We assume fixed emission and transition probabilities derived from empirical data [3].

We define the following generative model for simulating amino acid sequences: The sequence always starts in the "other" state. We output an amino acid based on the following tables of emission probabilities:

#### alpha-helix

A	R	N	D	C	E	Q	G	H	I
12%	6%	3%	5%	1%	9%	5%	4%	2%	7%
L	K	M	F	P	S	T	W	Y	V
12%	6%	3%	4%	2%	5%	4%	1%	3%	6%

#### other

A	R	N	D	C	E	Q	G	H	I
6%	5%	5%	6%	2%	5%	3%	9%	3%	5%
L	K	M	F	P	S	T	W	Y	V
8%	6%	2%	4%	6%	7%	6%	1%	4%	7%

We are also using these transition probabilities: If the current state is "alpha-helix", the next state is "alpha-helix" with probability  $p = 90\%$  and "other" with probability  $1 - p = 10\%$ . If the current state is "other", the next state is "alpha-helix" with probability  $p = 0.05$  and "other" with probability  $1 - p = 95\%$ . Using this simulator, we can simulate amino acid chains of arbitrary length. Additionally, with the Viterbi algorithm [4], it is also possible to infer state probabilities for a given amino acid sequence, e.g. using the hmmlearn Python package [5]. Given pairs of amino acid sequences and state probabilities as training data, the goal is to use BayesFlow to train a neural posterior density estimator. Then, compare the posterior state probability estimates for a new protein sequence to the known ground truth, for example, the annotate secondary structure of human insulin [6].

References:

1. <https://old-ib.bioninja.com.au/higher-level/topic-7-nucleic-acids/73-translation/protein-structure.html>
2. [https://scholar.harvard.edu/files/adeqirmenci/files/hmm\\_adeqirmenci\\_2014.pdf](https://scholar.harvard.edu/files/adeqirmenci/files/hmm_adeqirmenci_2014.pdf)
3. <https://www.kaggle.com/datasets/alfrandom/protein-secondary-structure>
4. <https://web.stanford.edu/~jurafsky/slp3/A.pdf>
5. <https://pypi.org/project/hmmlearn/>
6. <https://www.rcsb.org/3d-sequence/1A7F>

#### 6. Estimating Cosmological Parameters from Galaxy Power Spectrum

Lecturer: Aayush (aayush.mishra@tu-dortmund.de)

The large-scale structure of the universe encodes information about the composition, expansion history, and initial conditions of the universe [4]. The matter power spectrum,  $P(k)$ , describes how matter density fluctuations vary with scale and is one of the most important tools in modern cosmology [1, 4]. The shape and amplitude of  $P(k)$  are sensitive to Hubble constant ( $H_0$ ), matter density ( $\Omega_m$ ), and spectral index ( $n_s$ ). Accurate inference of these parameters helps constrain fundamental models like  $\Lambda$ CDM.

The goal of this project is to use BayesFlow to infer the posterior distribution over cosmological parameters  $\theta = \{H_0, \Omega_m, n_s\}$  given noisy observations of the matter power spectrum  $P(k)$

where  $k$  is the spatial frequency (wavenumber). Prior should reflect physically plausible ranges for each parameter. The power spectrum  $P(k)$  can be generated using **CAMB** [2], with additional observation noise (e.g., Gaussian). Evaluate the trained neural posterior estimator by checking posterior coverage using simulated test data, calibration plots, etc. Optionally, compare the posterior estimates of the parameters with Planck 2018 [3] cosmology fits.

References:

1. <https://people.ast.cam.ac.uk/~pettini/IntroCosmology/Lecture14.pdf>
2. <https://camb.readthedocs.io/en/latest/>
3. <https://arxiv.org/abs/1807.06209>
4. [https://fma.if.usp.br/~mlima/teaching/PGF5292\\_2021/Weinberg\\_Cosmology.pdf](https://fma.if.usp.br/~mlima/teaching/PGF5292_2021/Weinberg_Cosmology.pdf)

## 7. Gravitational Wave Source Parameter Estimation

Lecturer: Aayush (aayush.mishra@tu-dortmund.de)

When two black holes orbit each other, they emit gravitational waves (GWs) that encode the physical properties of the system. As the system evolves, the GW waveform changes in frequency and amplitude in a way that is uniquely dependent on:

- $m_1, m_2$ : Component masses (affect frequency evolution, merger time)
- $\chi_1, \chi_2$ : Dimensionless spin magnitudes along orbital axis (affect precession and waveform shape)
- $D$ : Luminosity distance (affects overall amplitude)
- Inclination angle of orbital plane (affects waveform polarization)

The waveform shape is governed by general relativity and can be modeled using **pyCBC** which gives a strain  $h(t)$  which is a weak signal buried in noise. The goal of the project is to recover the posteriors of parameters ( $\theta = \{m_1, m_2, \chi_1, \chi_2, D, \text{inclination}\}$ ) from noisy strain data from the simulator. Prior should reflect physically plausible ranges for each parameter. Time-domain waveform can be generated using **pyCBC** along with addition of suitable noise.

References:

1. <https://pycbc.org/pycbc/latest/html/>
2. <https://arxiv.org/abs/1508.07253>
3. <https://arxiv.org/abs/1809.02293>

## 8. Inferring Initial Conditions in the 3-body Problem

Lecturer: Aayush (aayush.mishra@tu-dortmund.de)

The *3-body problem* refers to the task of predicting the motion of three mutually interacting particles under gravity. Each body exerts a gravitational force on every other body, and their positions and velocities evolve over time according to Newton's laws of motion and universal gravitation. This system is *non-integrable* and often exhibits *chaotic behavior*, making precise long-term predictions extremely sensitive to initial conditions.

The goal is to infer the *initial conditions* – positions  $(x_1, x_2, x_3)$ , and velocities  $(v_1, v_2, v_3)$  at  $t = 0$  of a gravitational N-body system, given noisy observations of the system's state  $(x_1(T), x_2(T), x_3(T), v_1(T), v_2(T), v_3(T))$  at a later time  $t = T$ . The problem can be made more challenging by varying the masses  $(m_1, m_2, m_3)$ . In a separate case, instead of just using the simulated final positions and velocities at  $t = T$ , the trajectory at various time-steps can be used to infer the initial conditions.

Sample priors over physically reasonable masses, positions, and velocities. Simulate the trajectories of the N-body system using a high-accuracy ODE solver (e.g., RK4). Generate observations by

adding Gaussian noise to simulate observational uncertainty (i.e., final positions of  $N$  bodies + observational noise). Use that data to infer the posterior distribution of the initial conditions. You can use synthetic data for inference as a demonstration.

**Note:** As this is a chaotic system, small fluctuations or noise in the initial conditions could lead to a completely different trajectory which makes getting good posterior estimates challenging.

References:

1. Valtonen, M., Karttunen, H. (2006) The Three-Body Problem (Cambridge University Press).
2. <https://webhomes.maths.ed.ac.uk/~douglas/cortina-v2.pdf>
3. <https://blbadger.github.io/3-body-problem.html>

**Models** You need to use **BayesFlow** for estimating parameters of the statistical model of interest. Try to keep to a minimal working model so that you do not become distracted by too many moving parts in your project. If you need to make some simplifications to the scientific model even if they are against the scientific literature, do so. The goal is to implement a **BayesFlow** model and train it on data generated on simulators inspired by relevant scientific models, but not necessarily about implementing the state of the art scientific models.

## Presentation

One presentation per group. The grade will usually be the same for all group members. However, in cases where the quality of presentation was very clearly different between group members, we may choose to give different grades for individual group members.

Submit your slides in PDF format via email to Aayush ([aayush.mishra@tu-dortmund.de](mailto:aayush.mishra@tu-dortmund.de)) before the deadline (see **Schedule**).

- 10 minutes (group of 1–2) or 15 minutes (group of 3)
- The times are sharp. You will be interrupted after the scheduled time is over.
- Audience and course staff will ask questions.
- First slide should contain title and group members' names.
- Introduce the topic.
- Explain the statistical model, describe how you fit it with BayesFlow.
- Discuss your results.
- Use captions and labels for all figures.
- Don't overload your slides. Think about what you consider good slides if you belong to the audience.
- Last slide should contain a TL;DR summary (or take-home-message) and contact info.

- No THANK YOU slide.

## Report

One report per group, one grade for the entire group.

Submit your report materials via email to Aayush ([aayush.mishra@tu-dortmund.de](mailto:aayush.mishra@tu-dortmund.de)) before the deadline (see **Schedule**).

- **Content:**
  - Introduction: motivation, problem, modeling idea, illustrative figure
  - Data (if applicable): data collection, source, previous usage and analyses
  - Statistical model (simulator)
    - \* Explicit proper priors
    - \* Describe how the data is generated
  - Approximator
    - \* Describe the network architectures
    - \* Explicit details (how many layers, activation functions, regularization, etc.)
  - Training
    - \* Training budget (number of epochs, number of batches, batch size)
    - \* Training regime (offline/online)
    - \* Optimizer details, learning rate schedule, etc.
  - Diagnostics
    - \* Convergence
    - \* Simulation-based calibration
    - \* Other relevant checks (posterior contraction, etc.)
  - Inference
    - \* Fit the model to real data
    - \* Show and discuss results
    - \* Posterior predictive checks (if applicable)
  - Limitations and potential improvements
  - Conclusion
  - Reflection on own learnings



- Length:  $\approx 10$  pages (groups of 1–2) or  $\approx 15$  pages (groups of 3) main text (excluding title page, table of contents, and references). This is a *rough reference* as a guideline for you. If you can convey the required content with less text, that's fine. On the other hand, if you need more text to describe your project or want to include more figures to communicate your findings, that's fine as well. The exact length of your project report is up to your individual judgment.
- You can add an appendix which contains less important information like diagnostic plots or model details which do not fit in the main text. Your software code shall be submitted as a repository or zip-archive, so you generally don't have to print software code in the appendix.
- Formatting: 12pt, 2.5cm margins, continuous text. We provide you a L<sup>A</sup>T<sub>E</sub>X template, but you are free to use any word processing software of your choice.

## Grading

- The final grade will be determined by the teaching staff
- Presentation: The grade will usually be the same for all group members. However, in cases where the quality of presentation was very clearly different between group members, we may choose to give individual grades.
- Report: One grade for the entire group.
- Weight: 50% presentation, 50% report
- After presentation week: We will write you an email whether your group has passed ( $>4.0$ ) or failed ( $\leq 4.0$ ) the presentation part. If you fail the presentation, you will need at least a 3.0 in the report to pass the course.