STbayes: An R package for creating, fitting and understanding Bayesian models of social transmission

Michael Chimento[1,2,3] and William Hoppitt[4]

[1] *Max Planck Institute of Animal Behavior; Konstanz, Germany*
[2] *Centre for the Advanced Study of Collective Behaviour, University of Konstanz; Konstanz, Germany*
[3] *Department of Evolutionary Biology and Environmental Studies, University of Zurich, Zurich, Switzerland*
[4] *Institute of Science and Environment, University of Cumbria, Carlisle, UK*

Abstract

1. A critical consequence of joining social groups is the possibility of social transmission of information from conspecifics related to novel behaviours or resources. Mathematical models of spreading have been developed leading to several established methods of inferring whether information spreads via social or individual learning. Network-based diffusion analysis (NBDA) has emerged as a leading frequentist framework for inferring and quantifying social transmission, particularly in non-human animal populations.
2. NBDA has been extended several times to account for multiple diffusions, multiple networks, individual-level variables, and complex transmission functions. Bayesian implementations of NBDA were introduced in two prior studies. However, there is not yet a user-friendly package to implement a Bayesian NBDA, and the approach has seen limited usage by other researchers.
3. Here, we present a unified framework for performing Bayesian analysis of social transmission using NBDA-type models, implemented in the widely used Stan programming language. We provide a user-friendly R package "STbayes" (ST: social transmission) for other researchers to easily use this framework. STbayes accepts user-formatted data, but can also import data directly from the existing NBDA R package. Based on the data users provide, STbayes automatically generates multi-network, multi-diffusion models that allow for individual covariates that may influence transmission, as well as individual-level varying (random) effects.
4. Using simulated data, we demonstrate that this model can accurately differentiate the relative contribution of individual and social learning in the spread of information through networked populations. We illustrate how incorporating upstream uncertainty about the relationships between individuals in the social group can improve model fit. We show that our framework can be used to infer complex transmission rules and describe a new parameterization of frequency-dependent transmission that is more numerically stable than prior parameterizations. Finally, we introduce support for dynamic transmission weights and a "high-resolution" data mode, which allows users to make use of fine-scale data collected by contemporary automated tracking methods. These extensions increase the set of contexts that this type of model may be used for.

1 Introduction

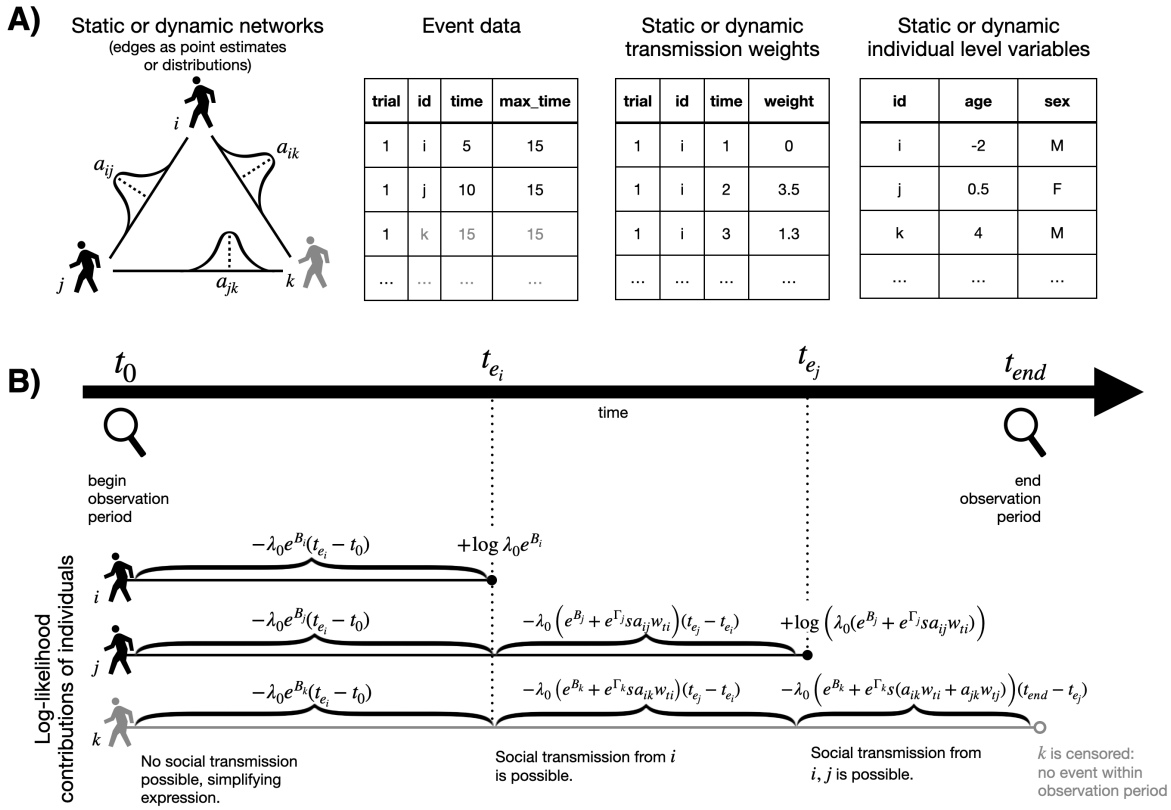
One key consequence of social interaction is the possibility of transmitting knowledge, behaviours, states, and diseases between individuals. Social transmission is a fundamental process that results in a variety of phenomena in collective behaviour and evolution [1, 2, 3]. Identifying and quantifying social transmission, relevant interaction networks, and transmission mechanisms are key questions for researchers interested in social behaviour. Network-based diffusion analysis (NBDA) has emerged as a leading statistical framework for detecting social transmission [4, 5, 1, 6]. The NBDA framework describes a family of time-to-event models. Time-to-event models, such as survival models, typically estimate hazard rates (and effects of covariates on hazard rates) from distributions of deaths over a given observation period. Rather than mortality events, NBDA has been used to study acquisition events where individuals acquire a novel, potentially socially learned behaviour. By combining data about the orders or times of events with the network structure of the population, NBDA can be used to infer the extent to which social transmission plays a role in the spread of a behaviour. However, these types of model could equally be applied in contexts other than novel behaviour, such as disease or information transmission, and for the remainder of the paper we use the term "event" as a catch-all for any of these contexts.

All NBDA models minimally require network data about the population under study, and event data where each observation is the occurrence of an event. The level of detail of either data type determines the specification of the log-likelihood of the model. If only the order of events is known, one can perform an order of acquisition diffusion analysis (OADA), where for each event, the likelihood specifies the probability of the next event occurring to individual i amongst the set of all other individuals for which the event has not occurred. If the times of events is known, one can perform a time of acquisition diffusion analysis (TADA). Different to OADA, the likelihood is specified by the joint probability of the event not happening to individual i before time t, and an event happening to individual i at time t. A detailed decision tree of what specific type of model to use, as well as recommended inferential steps for model selection to determine support for social transmission is provided by Hasenjager, Leadbeater, and Hoppitt [6].

Since it's introduction in 2009 [4], NBDA has been used to identify social transmission in a variety of taxa. Researchers have used it to determine whether or not novel behaviours were being socially learned in primates [7, 8, 9, 10], cetaceans [11, 12, 13], fish [14, 15], and birds [16, 17, 18, 19, 20]. NBDA has been adapted and extended to consider multiple networks [21], dynamic networks where nodes change their connections over time [8], and complex contagion, such as frequency dependent rules for acquisition [22, 23]. Bayesian implementations have been introduced before, in the context of spatial diffusions [24, 25], applying varying effects [26] and Bayesian model comparison [27]. However, these studies were published a decade ago, and the ecosystem of popular software and methods for fitting Bayesian models has evolved in the interim period. Bayesian NBDA has seen limited use or further development and to date, there is no unified framework for creating, fitting and comparing Bayesian models of social transmission.

There have been many software frameworks for Bayesian inference, but Stan [28] has emerged as a dominant framework across fields due to its active online forums and inclusion in popular contemporary textbooks [29, 30, 31, 32]. Instead of solving Bayes' Rule directly, which often involves intractable integrals, Stan approximates the posterior distribution using sampling algorithms. To sample from the posterior, Stan uses Hamiltonian Monte Carlo (HMC), an advanced Markov Chain Monte Carlo (MCMC) method. HMC uses gradient-based exploration to efficiently sample the probability space. It treats the posterior landscape like a physical system, where a particle moves according to the shape of the probability distribution, allowing it to reach high-probability regions faster and reduce sampling inefficiencies. Importantly, Stan readily accepts customized likelihood functions, making it possible to create flexible, complex models often required for studies of social transmission. Once a model is specified, it is compiled into optimized C++ code and the HMC sampler is run to generate thousands of posterior samples. This results in a full distribution of plausible parameter values, rather than just a point estimate, allowing users to compute credible intervals, visualize uncertainty, and make probabilistic predictions.

Here, we present a unified, user-friendly framework for performing Bayesian analysis of social transmission using NBDA-type models implemented in Stan. We create an R package "STbayes" that serves as a familiar interface between users and Stan, and greatly simplifies the workflow for organizing data, generating models, fitting those models, interpreting outputs and inference via model comparison. Additionally, we present several extensions of the NBDA framework, including the ability to account for uncertainty in network measurements, dynamic transmission weights, tools for cross-validation model comparison, and a novel parameterization for complex transmission. In the sections below we provide a primer for understanding Bayesian inference in the context of social transmission, a walk-through of the analysis pipeline using STbayes, and introduce the extensions to the framework.



trial	id	time	max_time
1	i	5	15
1	j	10	15
1	k	15	15
...	...	...	...

trial	id	time	weight
1	i	1	0
1	i	2	3.5
1	i	3	1.3
...	...	...	...

id	age	sex
i	-2	M
j	0.5	F
k	4	M
...	...	...

Figure 1: STbayes overview. A) Summary of data types. STbayes requires network data that defines edge weights between individuals, and event data, either the order in which or times that events occurred. Optionally, users can supply static transmission weights, representing the overall rate, or time-varying rates of behavioural usage or cues production that are relevant to the target events. Users may also provide static or time-varying ILVs. B) Toy example of log-likelihood function for a TADA-type model. In this example, three individuals are under observation, and individuals i and j acquire the target behaviour within the observation period. Shown are individuals' contributions to the log-likelihood during each inter-event period, as well as the contribution of when an event occurs for i and j.

2 Bayesian inference in a social transmission context

Bayesian inference can be summarized using Bayes' Rule:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}, \tag{1}$$

where θ are parameters from the data generating process and D is the observed data. $P(D|\theta)$ is the likelihood, or how likely the observed data is given some parameter values, $P(D)$ is the evidence (or marginal likelihood), which is the total probability of data across all possible parameter values, and $P(\theta|D)$ is the posterior, or an updated estimate of parameter values after seeing the data.

When studying social transmission, we are primarily interested in estimating two parameters: an intrinsic event rate (λ_0) that operates independently of social information, and the strength of social transmission (s), which quantifies the influence that interactions with others has on event rates. NBDA has been extended to include additive individual-level variables (ILVs) that might affect either the intrinsic rate or strength of social transmission, or multiplicative ILVs that equally affect both rates [5]. The coefficients of which may written as β, γ, and ϕ respectively. Thus, in a social transmission context:

$$\theta = \{\lambda_0, s, \beta, \gamma, \phi\}. \tag{2}$$

The data we usually collect in transmission studies are the states of individuals in a population over time ($z \in \{0, 1\}$), e.g. whether an event happened yet, such as the acquisition of novel behaviour or pathogen), network connections between individuals a, other relevant individual level variables x, and ideally transmission weights (w, e.g. behavioural production rates, in the context of the spread of a novel behaviour). Thus,

$$D = \{z, a, x, w\}. \tag{3}$$

The likelihood function for given parameter values is composed of two parts: the probability that each individual remained $z_i = 0$ until time t_e, and the probability that an individual became $z_i = 1$ at time t_e. The likelihood contribution of an individual who experienced an event during the observation period ($i \in N_e$) at time t_e is:

$$\log \mathcal{L}_{\text{uncensored}i} = -\sum_{t_0}^{t_e} \lambda_{it} + \log \lambda_{it_e}. \tag{4}$$

For individuals who do not learn ($i \in N_c$), the likelihood contribution is:

$$\log \mathcal{L}_{\text{censored},i} = -\sum_{t_0}^{t_{\text{end}}} \lambda_{ti}. \tag{5}$$

Here, λ_{ti} is the hazard function for individual at time t is given by:

$$\lambda_{ti} = \lambda_0 (1 - z_{ti}) e^{\Psi_i} \left(e^{B_i} + e^{\Gamma_i} s T(a, z, w, t) \right). \tag{6}$$

In a standard model, the transmission function T defines an simple linear relationship between connections to informed individuals and the hazard rate:

$$T(a, z, w, t) = \sum_j a_{ijt} z_{jt} w_{jt}, \tag{7}$$

although other complex transmission functions may be defined [22, 23], and we have included support for frequency-dependent transmission, detailed in Section 6. Additive ILVs for the intrinsic rate and social transmission are defined as $B_i = \sum_v \beta_v x_{vi}$ and $\Gamma_i = \sum_v \gamma_v x_{vi}$, respectively. These variable act independently on either rate. Multiplicative ILVs are defined as $\Psi_i = \sum_v \psi_v x_{vi}$. We note that we do not fit s directly, but rather we reparameterize equation 6 to:

$$\lambda_{ti} = (1 - z_{ti}) e^{\Psi_i} \left(\lambda_0 e^{B_i} + e^{\Gamma_i} s' T(a, z, w) \right), \tag{8}$$

s' represents $\lambda_0 s$, and the original interpretation of s is recovered by s'/λ_0 [24]. Both parameters are fit on the log scale for sampling performance, and the original exponentiated interpretations are output by STbayes for easy interpretation. The overall likelihood function can be summarized as:

$$\log \mathcal{L}(\theta|D) = \sum_{k=1}^{K} \left(\sum_i^{N_e} \left(-\sum_{t_0}^{t_e} \lambda_{kit} + \log \lambda_{kit_e} \right) + \sum_i^{N_{c_k}} \left(-\sum_{t_0}^{t_e} \lambda_{kti} \right) \right). \tag{9}$$

Summation over K diffusions, or trials, allows for multiple diffusions to be included. If precise times of events are not known, the user may also specify a discrete TADA (dTADA) likelihood, which accounts for this. If order, rather than timing of events is known, users can specify an OADA model, whose likelihood is described in supplementary text S1.

STbayes also supports non-constant hazards. The user can fit a Weibull hazard by specifying `generate_model(intrinsic_rate="weibull")`. This fits an additional parameter γ, where $\gamma < 1$ indicates a decreasing hazard over time, and $\gamma > 1$ indicates an increasing hazard over time. $\gamma = 1$ indicates a constant hazard.

Priors must also be defined. Our default, weakly informative priors are described in supplementary text S2. STbayes departs from the previous Bayesian NBDA in that priors are not entirely uninformative ($U(-10, 10)$ for $log(\lambda_0)$ and $\log(s')$). This extreme range of values generates highly unlikely predictions in a prior predictive check (Figure S1A), and is generally not recommended for parameters fit on a log scale [31]. For example, large values of λ_0 would result in diffusions finishing within the first timestep. Our weakly informative priors produce a more plausible range of predictions for most data, although these can be easily customized when generating a model.

3 The pipeline from data to inference in STbayes

STbayes is a user-friendly R package that allows researchers to create customized models of social transmission, and perform inferential tasks. The core functionality of STbayes is to 1) facilitate the intake and organization of data, 2) generate specifications of models in Stan based on that data and a users needs, 3) compile and

Notation	Name	Description
λ_i	hazard rate	the rate at which individual i experiences the target event (e.g. acquisition of behavior, pathogen, information).
λ_0	intrinsic rate (also called baseline rate [6])	the underlying rate of a target event in the absence of social transmission.
s'	social transmission rate	the rate of transmission per unit connection.
s	strength of social transmission	the rate of transmission per unit connection relative to the intrinsic rate.
z_i	event state	describes the state of individual i ($0 =$ event has not occurred, $1 =$ event has occurred).
a_{ij}	edge weight	describes the edge between individuals i and j within a network.
B_i	intrinsic rate ILV	linear combination of individual-level variables affecting the intrinsic rate.
Γ_i	transmission rate ILV	linear combination of individual-level variables affecting the social transmission rate.
Ψ_i	multiplicative ILV	linear combination of individual-level variables affecting both the intrinsic and social transmission rate.
w_j	transmission weight	describes the rate of a cue of interest (e.g. expression of target behaviour) is exhibited by individual j.

Table 1: Summary of notation.

fit those models, and 4) interpret the output of the models. STbayes implements existing extensions of the NBDA model, and the flexibility of Stan allows us to implement several extensions, including the propagation of uncertainty from network models to transmission models, the use of cross-validation methods for model comparison, the easy inclusion of random effects on any parameter, time-varying transmission weights, and posterior distributions of estimated event times. We first provide a step-by-step overview of how to use the full pipeline of STbayes using simulated data, and then describe each of the novel additions in turn. We emphasize that the code below is for illustrative purposes, and direct readers to the "Getting started" and "Advanced recipes" vignettes available in the Data Availability section for more details and examples.

3.1 Data intake

STbayes accepts data either in a user supplied format (illustrated in Figure 1A), or if users already use the NBDA package, STbayes will accept NBDA objects. Minimally, the user needs to provide an event dataframe and a networks dataframe. In the following examples we will manually create exemplar data-frames for illustrative purposes:

```
event_data <- data.frame(
    trial = rep(1:2, each = 3),
    id = LETTERS[1:6],
    time = c(2, 1, 3, 4, 2, 1),
    t_end = c(3, 3, 3, 4, 4, 4)
)
```

The user needs to provide an event dataframe that must contain columns:

- `id`: Character or numeric values of individual identities.
- `trial`: Character or numeric values indicating which trial the diffusion belongs to. If there is only one diffusion, set all values to 1.

- `time`: Integer or float values indicating the time (TADA) or order (OADA) in which the individual was recorded as first informed/knowledgable. If an individual had the event occur prior to the start of the observation period (e.g. pre-trained demonstrator), set as 0. These left censored individuals will not contribute to the likelihood calculation. If an individual never learned during the observation period, set its value $t_{\text{end}} + 1$. These will be treated as right-censored individuals in the likelihood calculation.

- `t_end`: Integer or float value representing the duration of the observation period for each trial. If the user observed a population for 30 days, `t_end=30`. If there are no censored individuals set `t_end=max(time)`.

The second dataframe gives information regarding the network connections between dyads in a long format (i.e. an edge list). The edge list can be weighted or binary, directed or undirected, symmetric (all combinations of individuals provided, useful for directed networks) or asymmetric:

```
networks <- data.frame(
    trial = rep(1:2, each = 3),
    from = c("A", "A", "B", "D", "D", "E"),
    to = c("B", "C", "C", "E", "F", "F"),
    kin = c(1, 0, 1, 0, 1, 1), # first network
    inverse_distance = c(0, 1, .5, .25, .1, 0) #second network
)
```

The first three columns must be:

- trial: Character or numeric column indicating which trial the networks belong to. If there is only one diffusion, set all values to 1.

- from: Character or numeric column of individual identities.

- to: Character or numeric column of individual identities.

Finally, the user must supply at least one descriptively named column of integer or float edge weights where all $w_{ij} \geq 0$. The column names will be carried forward in naming variables in the output of the model. In the example code above, we have defined a binary kin network and an inverse distance network.

Optionally, the user may supply dynamic networks that change over the observation period. The network should be summarized for each inter-event period. The networks dataframe should thus include the column:

- time: optional an integer column indicating which timestep the edges belong to.

For example, if the user's observation period included 10 events, they should supply edge weights from 11 networks in total, where `time=1` should contain the network representing the period from $[t_0, t_{e1})$, `time=2` represents $[t_{e1}, t_{e2})$, and `time=11` represents from $[t_{e10}, t_{\text{end}}]$. If the network is a static representation to be used for the entire observation period, this column is not required.

These two dataframes are sufficient to create a STbayes data object:

```
data_list <- import_user_STb(
   event_data = event_data,
   networks = networks)
)
```

3.1.1 Constant and time-varying ILVs

Users may supply constant or time-varying ILVs by also supplying dataframes summarizing each. At present, STbayes only accepts continuous or 2-level categorical variables, although we plan to expand this to $n > 2$ level categorical variables. For other variable types, the Stan model can be directly modified after generating it. For constant ILVs, `import_user_STb` requires a dataframe with `id` and variable values, where the column names are the desired variable names. Time-varying ILVs require an additional column `time`, and as in the networks dataframe above, the variables should be summarized for inter-event periods.

```
ILV_c <- data.frame(
    id = LETTERS[1:6],
    age = c(-1, -2, 0, 1, 2, 3), # Continuous variables should be normalized
    weight = c(0.5, .25, .3, 0, -.2, -.4),
```

```
    sex = c(0, 1, 1, 0, 1, 0), # Factor ILVs must be input as numeric
)

ILV_tv <- data.frame(
    trial = c(rep(1, each = 9),rep(2, each = 9)),
    id = c(rep(LETTERS[1:3], each=3), rep(LETTERS[4:6], each=3)),
    time = c(rep(1:3, times = 3), rep(1:3, times=3)),
    dist_from_resource = rnorm(18)
)
```

The user then supplies these dataframes to the import function, and specifies whether they should be treated additively or multiplicatively using arguments `ILVi`, `ILVs`, and `ILVm`:

```
data_list <- import_user_STb(
    event_data = event_data,
    networks = networks,
    ILV_c = ILV_c,
    ILV_tv = ILV_tv,
    ILVi = c("age", "dist_from_resource"),
    ILVs = c("age", "sex"),
    ILVm = c("weight")
)
```

The example configuration above indicates the user wishes to estimate effects of constant ILV age and time-varying ILV `dist_from_resource` only for the intrinsic rate. The effect of constant ILV `sex` will be estimated for the social transmission rate, as well as age. Note that this is an additive construction, so two separate effects of age will be estimated. The constant ILV `weight` is included as a multiplicative effect on both intrinsic and social transmission rates, meaning a single β_{weight} will be estimated.

3.2 Model generation

The next step is to dynamically generate an appropriately specified Stan model using a call to the `generate_STb_model()` based on the data supplied, as well as a few options the user must choose. Continuing our toy example above, we can illustratively generate Stan code for a model with varying effects for both parameters and a modified prior for $log(\lambda_0)$:

```
model = generate_STb_model(
    STb_data,
    data_type = "continuous_time",
    model_type = "full",
    intrinsic_rate = "constant",
    transmission_func = "standard",
    gq = T,
    est_acqTime = T,
    veff_ID = c("lambda_0", "s"),
    priors = list(log_lambda0="normal(-5,2)")

write(model, file="example_model.stan")
```

Argument `data_type=c("continuous_time", "discrete_time", "order")` distinguishes between cases where the times of events are known (TADA type models) or only the order of events is known (OADA type models). If exact times of acquisition are known the user can select `"continuous_time"`, creating a cTADA model. If a population was regularly (or irregularly) sampled, so the exact time of events are unknown but the rough interval they occurred in is known, the user should select `"discrete_time"`, which creates a dTADA model. Further discussion of the distinctions between these models, as well as a decision tree for choosing the appropriate model is provided in Hasenjager, Leadbeater, and Hoppitt [6].

Argument `model_type=c("full","asocial")` distinguishes creating a full model or an asocial model constrained to the intrinsic rate. Argument `intrinsic_rate=c("constant", "weibull")` specifies whether or not the user wishes to use a constant intrinsic rate, or a Weibull shaped rate that can increase or decrease over time. `transmission_func=c("standard","freqdep_f", "freqdep_k")` specifies whether

the model uses simple or complex transmission. Argument gq specifies whether or not a generated quantities block is created in the Stan model. This defaults to true, as the log-likelihood of observations is output using generated quantities and is needed for model comparison. `est_acqTime` specifies whether estimated acquisition times are also output by the generated quantities block. These are useful for posterior predictive checks, but this argument defaults to false for efficiency in fitting. Finally, `priors` allows users to specify priors for parameters in the format of a named list.

If the user is modelling repeated trials with the same individuals, it is possible fit varying effects (random effects in frequentist terminology) that model individuals' parameter estimates as normally distributed around some mean value. The NBDA package outputs JAGS code that can be used to fit varying effects, but does not offers a user friendly in-package solution. Argument `veff_ID=c("lambda_0", "s", ...)`, is used to specify which variables should receive varying effects by individual. For example, `veff_ID=c("lambda_0", "s")` will create a model that will attempt to fit varying effects for both intrinsic rate and strength of social transmission. It is possible to include other ILVs, if desired. For λ_0 and s, varying effects are added onto the main effect prior to transformation from log-scale back to linear. For example, if we apply a varying effect for λ_0, the model will calculate a value for each individual in the `transformed parameters` block of the Stan model:

$$\boldsymbol{\lambda}_{0,\text{ID}} = \exp(\mu_{\log \lambda_0} + \boldsymbol{v}_{\text{ID},\lambda_0}). \tag{10}$$

This ensures that λ_0 or s values are constrained to be positive. If models have convergence issues using varying effects, there is probably not enough data to meaningfully estimate these and we suggest fitting a simpler model. Finally, advanced users may optionally output the Stan model to text for further modification. However, this is not necessary for the model fitting process.

3.3 Model fitting

Users can then pass the formatted data and the model (either as the variable or a file path of an existing Stan model) to `fit_STb()`, which interfaces with CmdStanR [33].

```
fit = fit_STb(data_list_user,
    model_obj,
    chains = 5,
    cores = 5,
    parallel_chains=5,
    iter = 5000)
STb_save(fit, output_dir = "../data/saved_fits", name="my_first_fit")
```

The model will compile and run, and the function returns an object of class `CmdStanMCMC`. In addition to the data and model, the users can pass other arguments to CmdStanR, such as the number of independent chains, the number of iterations, and control parameters. Fitting the model can take anywhere between seconds to hours, depending on the user's machine, model complexity, amount of data, number of chains and iterations, and hyper-parameter values. After the model finishes, users can call function `STb_save()` that cleanly saves the fit and associated csv files containing samples to a named RDS file. By default, CmdStanR will save these csvs to a temporary folder, which is purged when R is restarted. If users are not familiar with CmdStanR, we strongly recommend using our function to save the output.

In order to validate that STbayes performed similarly to the frequentist NBDA package, we conducted 100 stimulated diffusions with known parameter values and fit models to the same data using both packages. For a description of simulations used for this manuscript, please see supplementary text S3. Stochastic simulation variation dominated the variance in estimated values, yet estimates were comparable between packages (Figure 2). Estimates were also strongly correlated between packages (Figure S2).

In order to validate that STbayes can recover varying effects, we simulated 100 diffusion trials from the same 50 individuals with known values for their intrinsic and social transmission rates. The model recovered main and varying effects (Figure 3). We advise that accurate recovery of varying effects requires many trials. We additionally ran the model on a subset of 10 and 50 trials to illustrate the differences in accuracy (Figure S3).

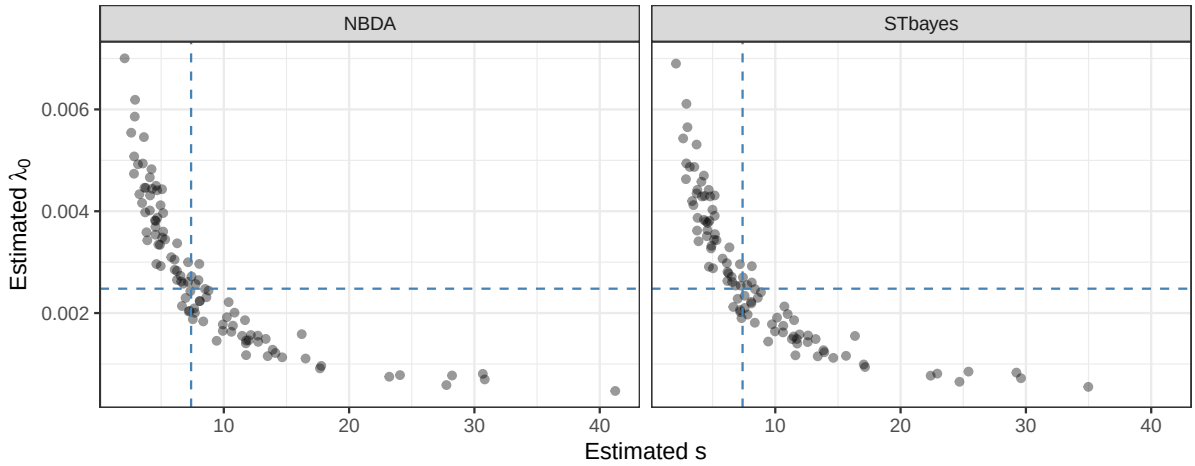


Figure 2: Comparison of estimated parameters between NBDA and STbayes. Shown are the point estimates for either s (x-axis) or λ_0 (y-axis) using either package (panel) from single simulations. Blue dashed lines indicate the true values of parameters used for all simulations. Both packages produce a similar distribution of estimates.

4 Interpreting data: model comparison using cross-validation methods

At this point, if the user is familiar with Bayesian analysis and CmdStanR, they may proceed to use their own pipeline for analysing the `CmdStanMCMC` object. For less familiar users, STbayes provides convenient functionality for understanding model outputs. Calling `STb_summary(fit)` will output a summary table of parameter estimates, confidence intervals, and diagnostics for parameters. Here we provide an example output from a single-network model in Table 2.

Parameter	Median	MAD	HPDI_Lower	HPDI_Upper	n_eff	Rhat
log_lambda_0_mean	-6.031	0.068	-6.172	-5.902	8102.993	1.000
log_s_prime_mean	-3.999	0.033	-4.065	-3.934	3704.247	1.001
lambda_0_mean	0.002	0.000	0.002	0.003	8103.006	1.000
sprime_mean	0.018	0.001	0.017	0.020	3704.233	1.001
s_mean	7.632	0.598	6.510	8.876	6840.528	1.001
sigma_ID[1]	0.258	0.097	0.025	0.430	2950.694	1.001
sigma_ID[2]	0.204	0.028	0.152	0.262	4131.197	1.001
percent_ST[1]	0.874	0.003	0.868	0.880	16510.609	1.000

Table 2: Typical summary statistics from STb_summary() function. Output taken from the varying effects model shown in Figure 3. Columns include parameter names, median value and median average divergence (MAD), upper and lower limits of the credible interval (defaults to 95% HPDI), number of effective samples (n_eff), and Rhat diagnostics. Log and linear values of λ_0 and s' are always given. The strength of social transmission ($s = s'/\lambda_0$) and %ST are calculated in the generated quantities block. Parameters sigma_ID[1] and sigma_ID[2] are the SD of varying effects for λ_0 and s'. For multi-network models, the summary function will show multiple s and %ST values associated with each network. The order will be identical to the columns in the networks dataframe.

Researchers may want to ask what is the percentage of events that occur through social transmission (%ST). In the case of a single network, we may calculate the probability of event e occurring through social transmission as the social component of the rate of acquisition divided by the full rate (intrinsic and social component), at t_e:

$$P(\text{e occurred through ST}) = \frac{e^{\Gamma_i} s' T(a, z, w)}{\lambda_{t_e}}. \tag{11}$$

%ST is then the mean probability over all events. This can be generalized to multi-network models by calculating the probability that each event occurred through social transmission on network n:

$$P(\text{e occurred through ST on network n}) = \frac{e^{\Gamma_i} s' T(a_n, z, w)}{\lambda_{t_e}}. \tag{12}$$

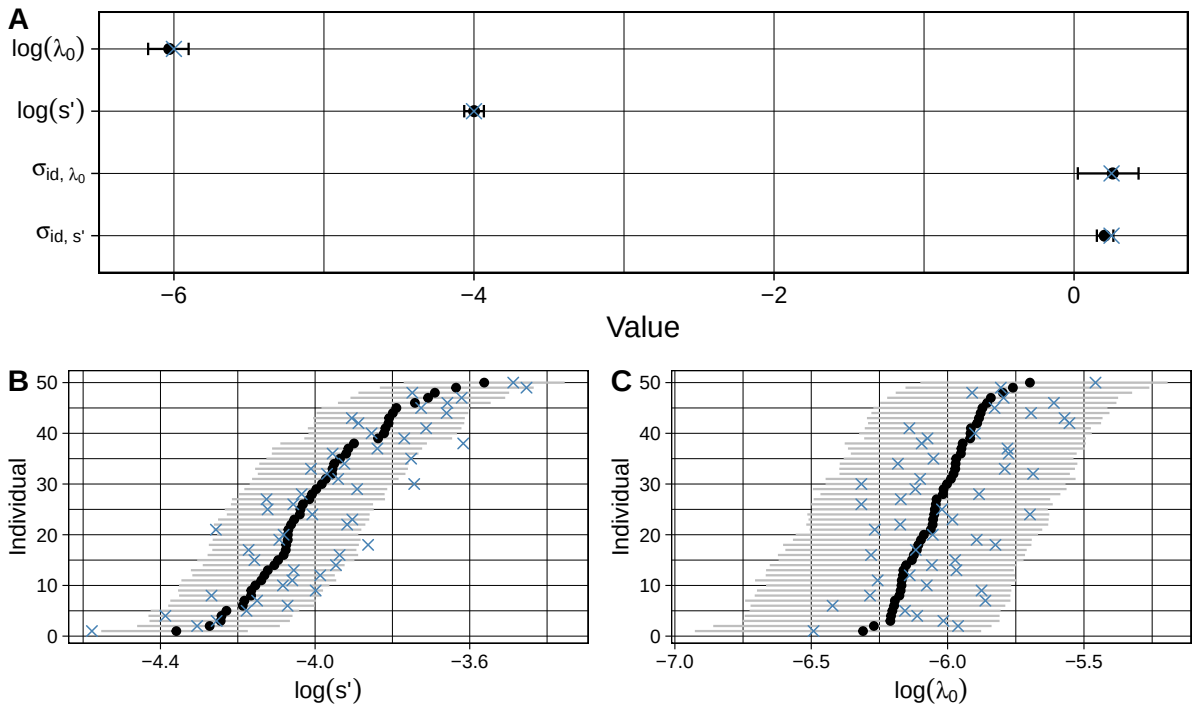


Figure 3: STbayes estimates of main and varying effects. We simulated 100 diffusion trials from the same 50 individuals with varying values of intrinsic rate and strength of social learning. A) Main effect estimates, black error bar and point are 95% HPDI and median, blue x is the true value. Varying effects for B) log s' and C) log λ_0, comparing the true value (x) with the estimated value and 95% CI.

The NBDA package produced confidence intervals of %ST by re-calculating this value using the upper and lower 95% CI for s [6]. However, STbayes refines this by computing the posterior distribution of %ST in the generated quantities block of the STbayes model, which includes full uncertainty over all parameters. When calling `STb_summary()`, %ST for network n is reported as `percent_ST[n]`.

Often researchers will want to perform model comparison, and `STb_compare()` gives a convenient function to evaluate multiple models against each other using functions from the *loo* package [34]. The NBDA package uses AIC (Akaike information criterion) for model comparison [35], using ΔAIC between constrained asocial models and social transmission models as evidence for social transmission [6]. However, there are several more contemporary options available for Bayesian model comparison using existing cross-validation tools, such as WAIC (Widely Applicable Information Criterion) [36] or LOO-PSIS (leave-one-out via pareto smoothed importance sampling) cross-validation [34], both of which use in-sample fits to estimate out-of-sample predictive accuracy. WAIC and LOO-PSIS are advantageous over AIC for several reasons. AIC assumes a single set of point estimates for parameter values that maximizes the likelihood function (MLE), and ignores any uncertainty around estimates. In comparison, WAIC and LOO-PSIS integrate over the entire posterior distribution, accounting for uncertainty. WAIC and LOO-PSIS are more appropriate for cases when a research has a small number of observations, or non-Gaussian likelihoods.

While it is beyond the scope of this paper to fully describe cross validation, we provide a basic description for unfamiliar readers. We fit a model to observations $y_i, ..., y_n$, representing the times at which each individual in a population experienced an event (e.g. acquiring a novel behaviour or disease). We define a set of parameters θ, which include the intrinsic rate, the social transmission rate, and other covariates, and the model outputs a posterior distribution of values for each parameter $p(\theta|y)$. Leave-one-out cross validation (LOO-CV) is a technique where the model is trained on $n-1$ observations and tested on the single left-out observation, repeating this process for each data point to estimate out-of-sample predictive performance, as measured by the expected log predictive density (ELPD). This is defined as:

$$\text{ELPD} = \sum_{i=1}^{n} \log p(y_i|y_{-i}, M) \tag{13}$$

where y_i represents the left-out observation, y_{-i} denotes the dataset excluding y_i, and M is the model under evaluation. The higher the ELPD, the better the model's predictive performance. However, computing

ELPD directly using LOO-CV is computationally expensive, requiring the model to be refitted multiple times. WAIC estimates ELPD by incorporating both the posterior mean log-likelihood and a variance-based penalty term. LOO-PSIS provides another approximation of ELPD, but instead of using log-likelihood variance for penalization, it uses importance sampling. We provide equations and descriptions of both measures in each in supplementary text S3.4. For a thorough description see Vehtari, Gelman, and Gabry [34]. Both can be converted to information criterion-style scores by multiplying ELDP by -2, matching the scale of other information criteria such as AIC, such that lower values of WAIC or LOOIC indicate better predictive performance.

In order to use cross-validation methods, the log-likelihood of each observation must also be calculated for each draw ($p(y_i|\theta_s)$) and exported from the model. This is efficiently done via the generated quantities block of a Stan model, such that log-likelihoods are computed once per iteration. The boolean gq argument of the `generate_STb_model()` function allows users to specify whether or not a generated quantities block is created in the STbayes model. Since all models are fit with the same set of n observations, we compare models by computing the standard error of the difference in ELPD estimates between models. For example, if one wanted to compute relative support for social transmission (equivalent to ΔAIC in the NBDA package):

$$se(\widehat{\text{ELPD}}^A - \widehat{\text{ELPD}}^B) = \sqrt{nV_{i=1}^n(\widehat{\text{ELPD}}_i^A - \widehat{\text{ELPD}}_i^B)}. \tag{14}$$

If the SE of a difference crosses zero, the user should be less certain that the best predictive model is any better than the alternative. This comparison can be accomplished in STbayes with `STb_compare()`, which automates the comparison between two or more models. We direct readers to the vignettes for further detail.

5 Accounting for network uncertainty

All social networks are inferred from observations, meaning that there is inherent uncertainty in network edges. One straightforward recommendation has been to use a threshold rule, removing individuals with few observations from social networks [37, 38]. However, such a strategy is not ideal for NBDA-type models, as missing edges can lead to the inference that an event occurred due to the intrinsic rate, rather than social transmission. Wild and Hoppitt [39] provided a tool for determining an appropriate threshold that would not be overly-conservative for NBDA models using simulations. However, recent Bayesian tools, such as the bisonR package [40] or STRAND package [41, 42] that estimate edge weights as distributions that reflect uncertainty, circumventing the need for using a threshold. Rather than calculating a single point estimate that summarizes the connection between individuals, STbayes can account for uncertainty in network measurement by feeding forward the posterior distribution of edge weights from another network model. Single model fits using bisonr or STRAND (or lists of model fits in the case of multi-network models) can be input to the `networks` argument of `import_user_STb()`). Following the approach of Hart et al. [40], the resulting STbayes model includes edge weights modelled by a multivariate normal approximation using the sample mean and covariance matrix calculated from the posterior edge weight samples from either the bisonr or STRAND model. This allows the joint uncertainty over edge weights to be accounted for.

To evaluate the impact of accounting for network uncertainty in social transmission models, we 1) simulated binary interaction data from a latent network with uneven sampling effort for different dyads, 2) fit a Bayesian network model using bisonr [40] to infer posterior edge distributions, and then 3) simulated 50 diffusion trials using the true network with known parameter values. We fit two transmission models to the resulting event data: one where edge weights were point estimates (only supplying the median value of edges from the network model), and another where edge weights were posterior distributions from the network model. Both models reasonably recovered the overall hazard (Figure 4A). The point estimate model over-estimated λ_0 (Table S1), while the model that marginalized over uncertainty recovered λ_0 accurately (Table S2). It better recovered the relative relationship between intrinsic and social rates (Figure 4B) and obtained significantly better predictive performance (LOOIC = 24520.6) than the point-estimate model (LOOIC = 24636.3). In summary, marginalizing over the posteriors of edges can help prevent the model from confidently over-fitting the data, especially in cases where there is uneven sampling effort among individuals.

There may be other types of measurement uncertainty in ILVs or missing data. Due to the complexity of automated model generation, it is difficult to create a generalisable solution in STbayes itself. However, advanced users can easily save and directly modify the Stan code for such cases, and pass the file path of the modified model to `fit_STb`. An automated solution may be supported in future releases of the package.

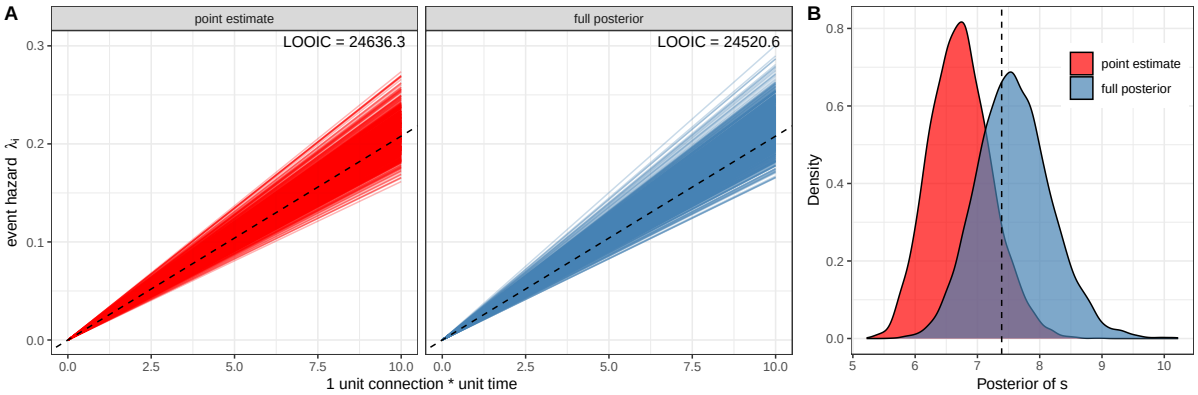


Figure 4: **Estimated relationship between number of informed associates and hazard rate.** STbayes allows for the propagation of uncertainty from network models to the estimation of diffusion parameters. A) We compared a model fitted with networks as point estimates (calculated as a simple ratio index, left panel) against a model incorporating uncertainty from the network model (right panel). Dashed lines indicate the ground truth hazard function used to simulate the data. Incorporating uncertainty in network edges can provide a more accurate estimate of parameters with better out-of-sample predictions as measured by LOOIC. B) Marginalizing over uncertainty can improve the estimate of the relative strength of social transmission.

6 Complex transmission rules

STbayes also allows for customized likelihoods that represent non-linear "complex" transmission mechanisms, such as frequency-dependent transmission [22, 23]. The user can choose between the standard linear transmission model, or a complex transmission model using the `transmission_func` argument of `generate_STb_model()`. Setting `transmission_func="freqdep_f"` introduces frequency-dependent bias parameter f and modifies the transmission function of the model to

$$T(a_i, z, w, t) = \frac{\sum_j a_{ij} z_{jt} {w_{jt}}^f}{\sum_j a_{ij} z_{jt} {w_{jt}}^f + \sum_j a_{ij}(1 - z_{jt})^f}. \tag{15}$$

The frequency-dependent bias parameter f is constrained as $f \geq 0$. We suggest an alternative parameterization of frequency-dependent bias (`transmission_func="freqdep_k"`) that is based on another tuneable sigmoid function that translates the proportion of "knowledgable" or "infected" associates into a normalized output y

$$\begin{aligned} x &= \frac{\sum_j a_{ij} z_{jt} w_{jt}}{\sum_j a_{ij} z_{jt} w_{jt} + \sum_j a_{ij}(1 - z_{jt})} \\ x' &= 2x - 1 \\ T(a_i, z, w, t) &= \frac{\left(\frac{x' - kx'}{k - 2k|x'| + 1} + 1\right)}{2}. \end{aligned}$$

The transformation depends on parameter k, which is constrained between $(-1, 1)$. Either parameterization results in a similar effect on how proportions of associates become converted into weights (Figure 5). The second parameterization has several useful qualities. The effect of k on the shape of the distribution is symmetrical whereas f is asymmetrical. Fitting k requires only sampling a narrow range of values. Finally, there is no exponentiation, meaning sampling is more numerically stable as the terms do not explode in the same way as equation 15 when sampling of f tends towards large numbers.

We first validated both complex models with simulated data from 10 trials of 50 individuals each with known parameter values using static networks. Both parameterizations supported significant terms for positive frequency dependent biases, and $\Delta ELPD_{loo}$ showed that they were better fit to the data than a model of simple transmission, but there was no significant difference in predictive performance between the two parameterizations (Figure 5C, D). Sampling efficiency was poorer with the first parameterization: a model with 5 chains of 5000 iterations had a mean chain execution time of 519.1 seconds under f parameterization, and 200 seconds under k parameterization. We strongly suggest that researchers use multiple trials (≥ 10 trials), as fits of individual trials produced wide error margins for k and f estimates.

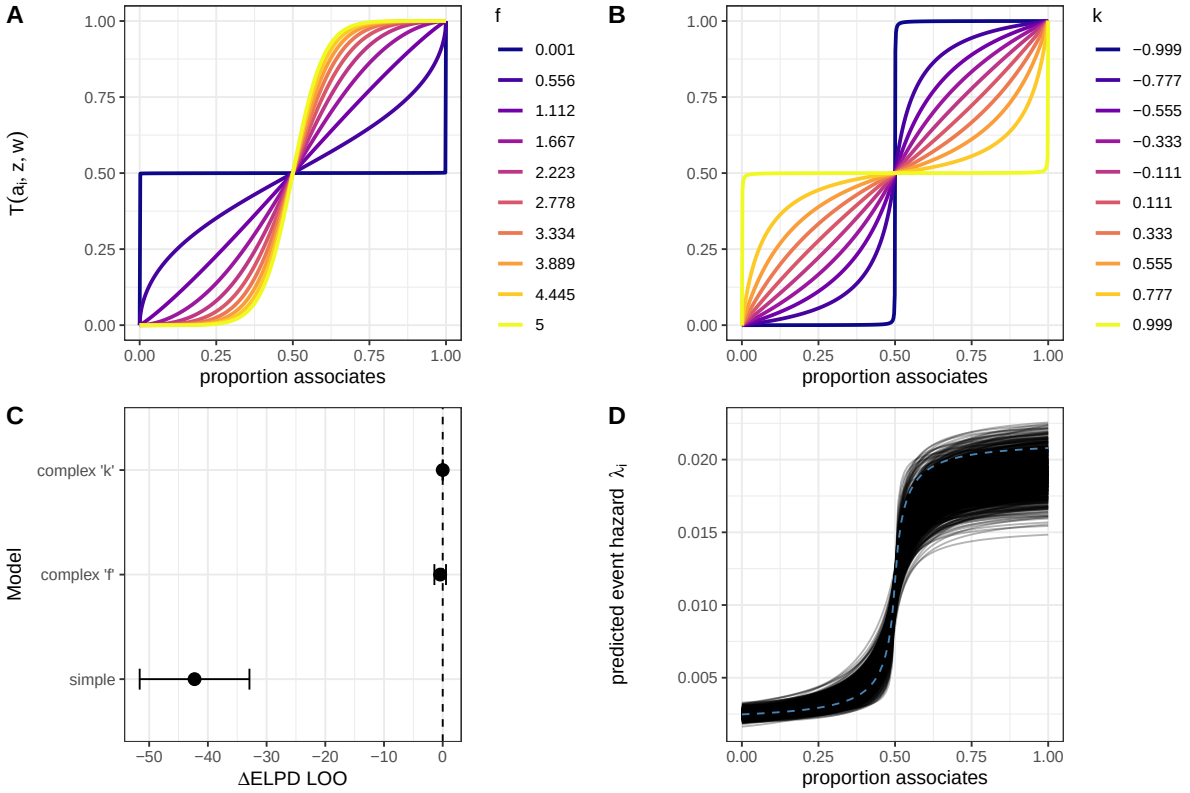


Figure 5: Alternative parameterizations of complex transmission. A) Traditional frequency-depended formulation with f parameter. B) Normalized tuneable sigmoid with parameter k. Both parameterizations result in similar transformations and interpretations of the shape parameters. C) Both parameterizations result in similarly good predictive performance as measured by differences in $\Delta ELPD_{loo}$ on simulated data. D) Estimated hazard as the proportion of informed associates increases from the complex k parameterization.

7 Dynamic networks, dynamic transmission weights

The NBDA package accepts static transmission weights (w_i) that represent a summary rate of production or usage of the target behaviour by a given individual over the observation period. STbayes extends this feature to allow for dynamic transmission weights (w_{it}) in the case that researchers have detailed behavioural time-series data. This is particularly important for accounting for variable behavioural production rates (in the context of social learning of a novel behaviour) or variable shedding rates (in the context of pathogen transmission) [43]. For example, in the context of animal social learning, if an individual knows a target behaviour, but doesn't use a target behaviour during an inter-event interval, then its influence on others should be null during that time period. Applying dynamic transmission weights is readily done in STbayes by providing a long format data-frame to `import_user_STb()` using argument `t_weights` that includes ID, time (as measured by inter-acquisition period), and the transmission weight value (e.g. rate of target behavioural production per unit time).

In order to illustrate the usefulness of including both dynamic networks and dynamic transmission weights, we simulated diffusion data from a spatially explicit, motile population of learners similar to the approach of Chimento and Farine [23]. The details of these simulations may be found in supplementary text S3. We recorded their interaction network each timestep (dynamic network A_{ijt}, Figure 6A), as well as an overall summary of their interaction for the duration of the observation period (static network A_{ij}, Figure 6B). We also recorded their usage of the target behaviour over time (dynamic transmission weight w_{it}), and a summary for the duration of the observation period (static transmission weight w_i). We then fit incrementally more complex models, illustrating that inclusion of all relevant data results in a better model fit (Figure 6C,D,E).

7.1 "High-resolution" data mode

Until now, NBDA models have been limited to incorporating dynamic networks and time-varying ILVs that only change at event cut-points. Above, we introduced the ability to allow transmission weights to change at cut-points. Of course, all of these variables are continuously changing in real diffusions. When data

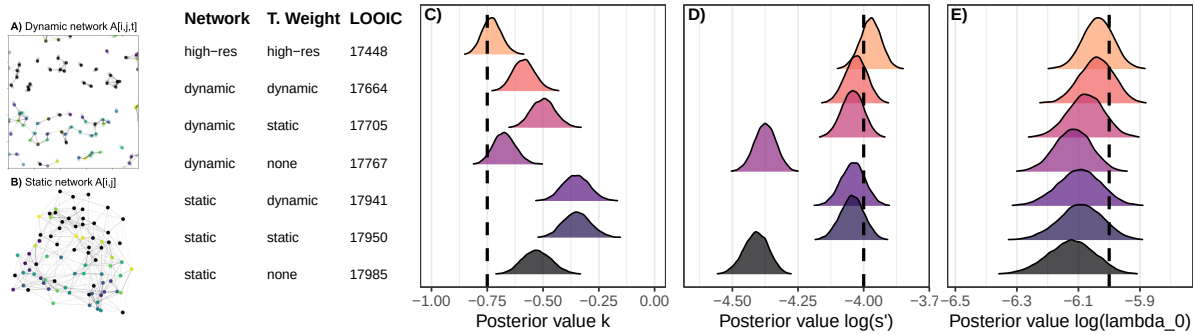


Figure 6: **Using dynamic transmission weights and networks can improve model fit and estimates.** A) We simulated diffusions using a positive frequency-dependent complex transmission rule on dynamic networks of motile agents, and then B) summarized their network over the duration of the observation period, as would be done to an empirical study. Using these dynamic and static networks, we fit increasingly complex models that contained more information about the diffusion. Posterior distributions of the frequency dependent parameter, $\log(s')$ and $\log(\lambda_0)$ are shown in panels C, D and E. Providing dynamic networks and transmission weights improved predictive performance and estimates. "High-resolution" models, which contained dynamic networks and dynamic transmission rates measured at each timestep resulted in the best model fit and most accurate parameter estimates.

was generated using motile agents who could constantly move, model estimates were less accurate than expected. The discrepancy was due to the omission of fine-scaled detail about the diffusion, as dynamic networks and transmission weights were summarized over inter-event intervals. With STbayes, we introduce "high-resolution" data mode, where these values may vary independently of event times. Providing this level of detail is useful specifically when network connections and transmission weights (e.g. usage of the target behaviour) are misaligned due to the movement of individuals, and this misalignment is not captured by a summary over inter-event intervals. While it has been practically difficult to collect high-resolution data throughout the course of a diffusion, recent methodological advances, such as 3D multi-individual tracking have opened up the opportunity to do so [44, 45, 46]. STBayes now provides updated tools to make full use of the resolution of that data.

Complex transmission creates especially difficult conditions to estimate parameters from cumulative representations of networks [23]. Parameter estimates were most accurate when using high resolution mode (Figure 6). However, we do warn users that depending on the amount of data, this may take a long time. Model fitting of the "high-resolution" model with complex transmission took 170 minutes ($N = 10$ trials, 50 individuals, 5 parallel chains for 5000 iterations), compared to the tens to several hundreds of seconds for all other models presented in this study. We find that providing dynamic transmission weights provides a good improvement in parameter estimates for s', although not for the frequency dependent parameter (Figure 6C,D). However, providing dynamic networks, even without dynamic transmission weights, helps identify the frequency dependent parameter (Figure 6C). We recommend care when interpreting the precise values of k or f when applied to real datasets in the absence of highly detailed dynamic networks and transmission weights. While we did not directly test this, it is also logical that inference about complex transmission becomes more difficult when the average degree of a network is low, as it might not provide enough information about the shape of the relationship between the unit connection to informed individuals and hazard rates.

We additionally tested the same set of models on data simulated from motile agents using the standard transmission rule. Since this rule is linear, we could optimized pre-processing of the data, and the high-resolution model fit a dataset of the same size as above in under 8 minutes. For standard transmission, providing dynamic networks or dynamic transmission weights significantly improved model fit and parameter estimates (Figure S4). "High-resolution" mode improved the model fit, but didn't significantly improve parameter estimates from dynamic networks or dynamic weights models, as they were already quite close to the true values. However, we suspect that this might depend on many factors, including the types of movements, length of diffusion, and strength of social transmission. Beyond the case of complex transmission in motile populations, more research is needed to understand when it is critical to include extremely high resolution data.

8 Discussion

STbayes introduces a flexible yet user-friendly framework for modelling social transmission using Bayesian inference. While the frequentist NBDA approach has been a valuable tool, it has lacked a unified, user-

friendly Bayesian alternative. STbayes fills this gap, allowing users to specify complex models of diffusion processes with fine control over priors, model structure, and uncertainty propagation. By building on the Stan ecosystem, STbayes benefits from contemporary advances in Bayesian computation.

While NBDA-type models has been largely been applied in the field of social learning and cultural transmission, we emphasize that this family of models can be used in a variety of contexts, as long as there are measurable events and measurable interactions. Our introduction of dynamic transmission weights is not only useful for including data about changing behavioural usage over time in the context of social learning, but importantly expands the applicability of these models to recurrent events that are relevant for the study of social contagion: yawning, sleep or other state statuses that may vacillate over time. We also highlight that dynamic transmission weights offer researchers creativity in hypothesis testing. If a researcher wants to ask which specific socially observed cues influence the hazard rate of a target event, they may perform model comparison between models fit on dynamic transmission weight data of different cues.

Another contribution of STbayes is the ability to propagate uncertainty from network models to transmission models: a meaningful step toward more robust inference in studies of social transmission. Social network data are almost always estimates based on limited observations, yet most analyses have treated them as fixed and error-free. STbayes allows users to incorporate posterior distributions from upstream network models (e.g., BISON or STRAND) directly into the inference process, thereby accounting for uncertainty in network structure [40, 41, 42]. This is especially important when sampling efforts are known to differ between dyads.

Another key extension is the inclusion of non-linear, frequency-dependent transmission functions. These allow researchers to test for frequency-dependent biases in the transmission process. STbayes includes two parameterizations of complex contagion, both of which yield similar predictive performance, though the alternative sigmoid formulation is more numerically stable. We note however that the interpretation of the s parameter changes here, as it is no longer per unit connection.

Finally, STbayes encourages a shift away from information criteria such as AIC toward more contemporary Bayesian methods for model comparison, such as LOO-PSIS or WAIC [34, 36]. These methods integrate over the full posterior distribution rather than relying on point estimates, making them more appropriate for potentially sparse and noisy data that might be collected in studies of social transmission.

As researchers move toward more complex and nuanced models of transmission, STbayes offers a platform for inference that is adaptable to a wide range of research questions. We hope that this framework not only improves the reproducibility and robustness of inference, but also encourages further exploration of social transmission processes in both natural and experimental settings. While we have provided example code within the manuscript, we strongly encourage readers to explore the vignettes available at the Github repository for more worked examples.

9 Data Availability

Package documentation, including installation instructions and vignettes can be found at `https://michaelchimento.github.io/STbayes/index.html`. The package will receive an official release until after acceptance. Code and data for statistical analyses and main text figures is available on Edmond [47].

10 Acknowledgements

MC received support from the Centre for the Advanced Study of Collective Behaviour, funded by the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy (EXC 2117-422037984) and the Swiss State Secretariat for Education, Research and Innovation (SERI) under contract number MB22.00056. The Max Planck Society provided open access data storage.

11 Competing Interests

The authors declare no competing interests, financial or otherwise.

References

[1] William Hoppitt and Kevin N Laland. *Social learning: an introduction to mechanisms, methods, and models*. Princeton University Press, 2013.

[2] Mauricio Cantor et al. "The importance of individual-to-society feedbacks in animal ecology and evolution". In: *Journal of Animal Ecology* 90.1 (2021), pp. 27–44.

[3] Christos C Ioannou and Kate L Laskowski. "A multi-scale review of the dynamics of collective behaviour: from rapid responses to ontogeny and evolution". In: *Philosophical transactions of the royal society B* 378.1874 (2023), p. 20220059.

[4] Mathias Franz and Charles L Nunn. "Network-based diffusion analysis: a new method for detecting social learning". In: *Proceedings of the Royal Society B: Biological Sciences* 276.1663 (2009), pp. 1829–1836.

[5] William Hoppitt, Neeltje J Boogert, and Kevin N Laland. "Detecting social transmission in networks". In: *Journal of Theoretical Biology* 263.4 (2010), pp. 544–555.

[6] Matthew J Hasenjager, Ellouise Leadbeater, and William Hoppitt. "Detecting and quantifying social transmission using network-based diffusion analysis". In: *Journal of Animal Ecology* 90.1 (2021), pp. 8–26.

[7] Rachel L Kendal et al. "Evidence for social learning in wild lemurs (Lemur catta)". In: *Learning & Behavior* 38.3 (2010), pp. 220–234.

[8] Catherine Hobaiter et al. "Social network analysis shows direct evidence for social transmission of tool use in wild chimpanzees". In: *PLoS Biol* 12.9 (2014), e1001960.

[9] Stuart K Watson et al. "Socially transmitted diffusion of a novel behavior from subordinate chimpanzees". In: *American journal of primatology* 79.6 (2017), e22642.

[10] Charlotte Canteloup, William Hoppitt, and Erica van de Waal. "Wild primates copy higher-ranked individuals in a social transmission experiment". In: *Nature communications* 11.1 (2020), pp. 1–10.

[11] Jenny Allen et al. "Network-based diffusion analysis reveals cultural transmission of lobtail feeding in humpback whales". In: *Science* 340.6131 (2013), pp. 485–488.

[12] Sonja Wild et al. "Multi-network-based diffusion analysis reveals vertical cultural transmission of sponge tool use within dolphin matrilines". In: *Biology letters* 15.7 (2019), p. 20190227.

[13] Sonja Wild et al. "Integrating genetic, environmental, and social networks to reveal transmission pathways of a dolphin foraging innovation". In: *Current Biology* 30.15 (2020), pp. 3024–3030.

[14] Mike M Webster et al. "Environmental complexity influences association network structure and network-based diffusion of foraging information in fish shoals". In: *The American Naturalist* 181.2 (2013), pp. 235–244.

[15] Matthew J Hasenjager, William Hoppitt, and Lee A Dugatkin. "Personality composition determines social learning pathways within shoaling fish". In: *Proceedings of the Royal Society B* 287.1936 (2020), p. 20201871.

[16] Damien R Farine, Karen A Spencer, and Neeltje J Boogert. "Early-life stress triggers juvenile zebra finches to switch social learning strategies". In: *Current Biology* 25.16 (2015), pp. 2184–2188.

[17] Virginia K Heinen et al. "Food discovery is associated with different reliance on social learning and lower cognitive flexibility across environments in a food-caching bird". In: *Proceedings of the Royal Society B* 288.1951 (2021), p. 20202843.

[18] Barbara C Klump et al. "Innovation and geographic spread of a complex foraging culture in an urban parrot". In: *Science* 373.6553 (2021), pp. 456–460.

[19] Ipek G Kulahci et al. "Social networks predict selective observation and information spread in ravens". In: *Royal Society open science* 3.7 (2016), p. 160256.

[20] Zoltán Tóth et al. "The effect of social connections on the discovery of multiple hidden food patches in a bird species". In: *Scientific reports* 7.1 (2017), pp. 1–9.

[21] Damien R Farine et al. "Interspecific social networks promote information transmission in wild songbirds". In: *Proceedings of the Royal Society B: Biological Sciences* 282.1803 (2015), p. 20142804.

[22] Josh A Firth et al. "Analysing the Social Spread of Behaviour: Integrating Complex Contagions into Network Based Diffusions". In: *arXiv preprint arXiv:2012.08925* (2020).

[23] Michael Chimento and Damien R Farine. "The contribution of movement to social network structure and spreading dynamics under simple and complex transmission". In: *Philosophical Transactions B* 379.1912 (2024), p. 20220524.

[24] Glenna F Nightingale et al. "Bayesian spatial NBDA for diffusion data with home-base coordinates". In: *PLoS One* 10.7 (2015), e0130326.

[25] Glenna Nightingale et al. "Quantifying diffusion in social networks: a Bayesian approach". In: *Animal social networks* (2015), pp. 38–52.

[26] Neeltje J Boogert et al. "Perching but not foraging networks predict the spread of novel foraging skills in starlings". In: *Behavioural processes* 109 (2014), pp. 135–144.

[27] Andrew Whalen and William JE Hoppitt. "Bayesian model selection with network based diffusion analysis". In: *Frontiers in psychology* 7 (2016), p. 409.

[28] Bob Carpenter et al. "Stan: A probabilistic programming language". In: *Journal of statistical software* 76 (2017), pp. 1–32.

[29] Andrew Gelman et al. *Bayesian Data Analysis*. CRC Press, 2013.

[30] Franzi Korner-Nievergelt et al. *Bayesian data analysis in ecology using linear models with R, BUGS, and Stan*. Academic Press, 2015.

[31] Richard McElreath. *Statistical rethinking: A Bayesian course with examples in R and Stan*. Chapman and Hall/CRC, 2018.

[32] Alicia A Johnson, Miles Q Ott, and Mine Dogucu. *Bayes rules!: An introduction to applied Bayesian modeling*. Chapman and Hall/CRC, 2022.

[33] Jonah Gabry et al. *cmdstanr: R Interface to 'CmdStan'*. R package version 0.9.0, https://discourse.mc-stan.org. 2025. URL: https://mc-stan.org/cmdstanr/.

[34] Aki Vehtari, Andrew Gelman, and Jonah Gabry. "Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC". In: *Statistics and computing* 27 (2017), pp. 1413–1432.

[35] Hirotogu Akaike. "Information theory and an extension of the maximum likelihood principle". In: ed. by F. Caski B.N. Petrov. Akademiai Kiado, 1973, pp. 267–281.

[36] Sumio Watanabe and Manfred Opper. "Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory." In: *Journal of machine learning research* 11.12 (2010).

[37] Daniel W Franks, Graeme D Ruxton, and Richard James. "Sampling animal association networks with the gambit of the group". In: *Behavioral ecology and sociobiology* 64 (2010), pp. 493–503.

[38] Matthew J Silk et al. "The consequences of unidentifiable individuals for the analysis of an animal social network". In: *Animal Behaviour* 104 (2015), pp. 1–11.

[39] Sonja Wild and William Hoppitt. "Choosing a sensible cut-off point: assessing the impact of uncertainty in a social network on the performance of NBDA". In: *Primates* 60.3 (2019), pp. 307–315.

[40] Jordan Hart et al. "BISoN: a Bayesian framework for inference of social networks". In: *Methods in ecology and evolution* 14.9 (2023), pp. 2411–2420.

[41] Daniel Redhead, Richard McElreath, and Cody T Ross. "Reliable network inference from unreliable data: A tutorial on latent network modeling using STRAND." In: *Psychological methods* (2023).

[42] Cody T Ross, Richard McElreath, and Daniel Redhead. "Modelling animal network data in R using STRAND". In: *Journal of Animal Ecology* 93.3 (2024), pp. 254–266.

[43] Michael Chimento et al. "Cultural diffusion dynamics depend on behavioural production rules". In: *Proceedings of the Royal Society B* 289.1980 (2022), p. 20221001.

[44] Tristan Walter and Iain D Couzin. "TRex, a fast multi-animal tracking system with markerless identification, and 2D estimation of posture and visual fields". In: *eLife* 10 (Feb. 2021). Ed. by David Lentink, e64000. ISSN: 2050-084X. DOI: 10.7554/eLife.64000. URL: https://doi.org/10.7554/eLife.64000.

[45] Máté Nagy et al. "SMART-BARN: Scalable multimodal arena for real-time tracking behavior of animals in large numbers". In: *Science Advances* 9.35 (2023), eadf8068.

[46] Mathilde Delacoux and Fumihiro Kano. "Fine-scale tracking reveals visual field use for predator detection and escape in collective foraging of pigeon flocks". In: *Elife* 13 (2024), RP95549.

[47] Michael Chimento and William Hoppitt. *Data and Code to Reproduce "STbayes: An R package for creating, fitting and understanding Bayesian models of social transmission"*. Version DRAFT VERSION. 2025. DOI: 10.17617/3.IY3TKX. URL: https://doi.org/10.17617/3.IY3TKX.