Albatross

a scalable simulation-based inference pipeline for analysing stellar streams in the Milky Way

Alvey, J.; Gerdes, M.; Weniger, C.

DOI
10.1093/mnras/stad2458

Publication date
2023

Document Version
Final published version

Published in
Monthly Notices of the Royal Astronomical Society

Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **525,** 3662–3681 (2023) https://doi.org/10.1093/mnras/stad2458
Advance Access publication 2023 August 16

Albatross: a scalable simulation-based inference pipeline for analysing stellar streams in the Milky Way

James Alvey ,* Mathis Gerdes and Christoph Weniger
GRAPPA Institute, Institute for Theoretical Physics Amsterdam, University of Amsterdam, Science Park 904, NL-1098 XH Amsterdam, the Netherlands

Accepted 2023 August 10. Received 2023 August 10; in original form 2023 April 30

ABSTRACT

Stellar streams are potentially a very sensitive observational probe of galactic astrophysics, as well as the dark matter population in the Milky Way. On the other hand, performing a detailed, high-fidelity statistical analysis of these objects is challenging for a number of key reasons. First, the modelling of streams across their (potentially billions of years old) dynamical age is complex and computationally costly. Secondly, their detection and classification in large surveys such as *Gaia* renders a robust statistical description regarding e.g. the stellar membership probabilities, challenging. As a result, the majority of current analyses must resort to simplified models that use only subsets or summaries of the high quality data. In this work, we develop a new analysis framework that takes advantage of advances in simulation-based inference techniques to perform complete analysis on complex stream models. To facilitate this, we develop a new, modular dynamical modelling code `sstrax` for stellar streams that is highly accelerated using `jax`. We test our analysis pipeline on a mock observation that resembles the GD1 stream, and demonstrate that we can perform robust inference on all relevant parts of the stream model simultaneously. Finally, we present some outlook as to how this approach can be developed further to perform more complete and accurate statistical analyses of current and future data.

Key words: software: data analysis – software: simulations – galaxies: star clusters: general – Galaxy: structure – dark matter.

1 INTRODUCTION

1.1 Motivation

Stellar streams are very old, dynamical objects consisting of a collection of stars that originate from tidal disruptions of a dwarf galaxy (e.g. the Sagittarius stream (Belokurov et al. 2006; Gibbons, Belokurov & Evans 2014)) or globular cluster (e.g. the GD1 stream (Grillmair & Dionatos 2006; Eyre 2010; Carlberg & Grillmair 2013; Price-Whelan & Bonaca 2018; Bonaca et al. 2020)). In a galaxy such as the Milky Way, these systems have the potential to be an extremely sensitive probe of dark matter substructure (Erkal & Belokurov 2015b; Banik et al. 2018; Banik & Bovy 2019; Bechtol et al. 2019; Banik et al. 2021a, b; Hermans et al. 2021b; Malhan, Valluri & Freese 2021; Pavanel & Webb 2021), baryonic physics and Milky Way properties (Koposov, Rix & Hogg 2010; Bonaca et al. 2014; Sanderson, Helmi & Hogg 2014; Amorisco et al. 2016; Bovy et al. 2016; Bovy, Erkal & Sanders 2017; Erkal et al. 2019; Helmi 2020; Koposov et al. 2023), as well as the evolution history of the stream (Balbinot & Gieles 2018; Banik & Bovy 2021; Gialluca, Naidu & Bonaca 2021; Doke & Hattori 2022; Malhan et al. 2022). In principle, this can be achieved by combining high precision observations at facilities such as *Gaia* (Gaia Collaboration 2018, 2021) or the Vera Rubin observatory (Abell et al. 2009; Bechtol et al. 2019), and consistent modelling of these stellar orbits as the systems are disrupted over the course of $\mathcal{O}$(billions) of years.

1.2 Observational status

In the early 2000s, the first observations of cold (and hot) stellar streams in the Milky Way were obtained by the SDSS survey (Abazajian et al. 2009), the most well-known of which being the GD1 stream (Grillmair & Dionatos 2006; Eyre 2010; Carlberg & Grillmair 2013; Price-Whelan & Bonaca 2018; Bonaca et al. 2020). Since then, many more streams have been discovered in surveys such as SDSS and *Gaia* (Abazajian et al. 2009; Gaia Collaboration 2018, 2021; Malhan, Ibata & Martin 2018; Ibata et al. 2021a; Martin et al. 2022), but perhaps more importantly, the resolution of the observations has improved dramatically. Current observations have revealed, for example, interesting substructure and features in cold stellar streams such as GD1 (Ibata, Lewis & Irwin 2002; Johnston, Spergel & Haydn 2002; de Boer et al. 2018; Price-Whelan & Bonaca 2018; Bonaca et al. 2020). This is the context in which we want to consistently analyse both the large scale structure of the streams (such as its location on the sky and track), and the small scale structure that is sensitive to e.g. the dynamics and details of the tidal stripping process, or baryonic/dark matter interactions.

There are a number of relevant aspects to analysing stellar streams – stream modelling, inference, and observations. Since the main focus of this work is the statistical analysis of streams, we will briefly review its current status and the corresponding claims, although we will return to computational models for streams when we describe our dynamics code below. Previous analyses have typically focused on either the global structure of the stream, see e.g. Refs. (Koposov et al. 2010; Bonaca et al. 2014; Gibbons et al. 2014; Sanderson et al. 2014; Bowden, Belokurov & Evans 2015; Bovy et al. 2016; Gialluca et al. 2021; Pavanel & Webb 2021; Shipp et al. 2021; Dillamore et al.

* E-mail: j.b.g.alvey@uva.nl

2022) (where it is on the sky and the fits to the general stream track) or construct some sort of summary statistics to study perturbations in the stellar density along the stream object, see e.g.(Carlberg & Grillmair 2013; Erkal & Belokurov 2015b; Amorisco et al. 2016; Bovy et al. 2017; Banik et al. 2018; Bonaca et al. 2018; Banik & Bovy 2019; Erkal et al. 2019; Bonaca et al. 2020; Banik et al. 2021a, b; Hermans et al. 2021b; Doke & Hattori 2022).[1] The former of these analysis methodologies is well suited for studying properties and phenomena that are specific to the orbit and evolution of a given stream. For example, one can constrain quantities such as the Milky Way potential (Koposov et al. 2010; Bonaca et al. 2014; Gibbons et al. 2014; Sanderson et al. 2014; Bowden et al. 2015; Bovy et al. 2016; Erkal et al. 2019; Shipp et al. 2021; Craig et al. 2023; Nibauer et al. 2022; Nibauer, Bonaca & Johnston 2023), the age of the stream (Bovy et al. 2017; Hermans et al. 2021b), or possibly even gain information about close encounters with large perturbers which can leave large gaps or features in the stream track (Erkal & Belokurov 2015b; Amorisco et al. 2016; Bovy et al. 2017; Bonaca et al. 2020; Banik et al. 2021b, a). The classic examples that are often quoted in the literature along these latter lines are the so-called 'spur' and 'gaps' in the GD1 stream (Carlberg & Grillmair 2013; Bonaca et al. 2018; Price-Whelan & Bonaca 2018; Doke & Hattori 2022). On the other hand, the substructure of the stream is better suited to asking questions about e.g. the physics of tidal stripping mechanisms in the Milky Way, see e.g. (Baumgardt 1998; Takahashi & Portegies Zwart 2000; Taylor & Babul 2001; Baumgardt & Makino 2003; Drakos, Taylor & Benson 2022), the internal dynamics and nature of the progenitor, and population level information about smaller (or more distant) perturbers (Amorisco et al. 2016; Balbinot & Gieles 2018; Gialluca et al. 2021; Dillamore et al. 2022; Doke & Hattori 2022). From the perspective of the dark matter community, both the large and small perturbing objects are of huge significance in the context of the distribution of dark matter subhaloes in the Milky Way (and other galaxies). Indeed, one of the key goals of stellar stream analyses is to constrain possible subhalo populations (Banik et al. 2018, 2021a, b; Banik & Bovy 2019; Hermans et al. 2021b; Pavanel & Webb 2021; Delos & Schmidt 2022), or provide a detection of some larger mass (say, $10^7\,\mathrm{M}_\odot$) subhalo (Erkal & Belokurov 2015b; Bonaca et al. 2018). The main motivation behind our work is to provide a path towards a robust analysis pipeline to consistently (and simultaneously) analyse all of the above scenarios.

1.3 Statistical challenge

Making statistically robust statements about quantities of interest – the gravitational potential of the host, the disruption history, internal dynamics of the progenitor etc. – can be extremely challenging (Huang et al. 2019; Hermans et al. 2021b; Koposov et al. 2023). To do so requires us to have good control over the dynamical history and initial conditions of the stream (Penarrubia et al. 2006; Kuepper et al. 2010; Kuepper, Lane & Heggie 2012; Bovy 2014; Bovy 2015; Bowden et al. 2015; Buist & Helmi 2015; Fardal, Huang & Weinberg 2015; Qian, Arshad & Bovy 2022), its stochastic interactions with dark matter or baryonic substructures (Erkal & Belokurov 2015a; Bovy et al. 2017; Delos & Schmidt 2022), as well as a reasonable model for foreground and selection effects in the observations, see e.g. (Huang et al. 2019). As a result of the large number of free parameters this can introduce, together with relatively costly simulations, classical statistical methods scale quite poorly. Currently, this means that one must instead rely on constructing bespoke summary statistics such as the power spectrum of density perturbations along the stream, significantly reducing the dimensionality of the data via e.g. only considering the stream track, or ignoring a subset of effects in the modelling to lower the simulation overhead. This approach has been used to obtain relevant results regarding e.g. the properties of the Milky Way potential (Bonaca et al. 2014; Gibbons et al. 2014; Sanderson et al. 2014; Erkal et al. 2019; Helmi 2020; Panithanpaisal et al. 2022; Koposov et al. 2010, 2023), or the evolution history of progenitors (Balbinot & Gieles 2018; Banik & Bovy 2021; Gialluca et al. 2021; Doke & Hattori 2022; Malhan et al. 2022). In this paper, we propose using the modern tools and techniques of simulation based inference (Brehmer & Cranmer 2020; Cranmer, Brehmer & Louppe 2020) to analyse stellar streams and overcome some of these challenges.

1.4 Simulation-based inference

Given the context described above, we briefly argued that the analysis of stellar streams was a problem that is well-suited for the application of simulation-based inference (SBI) (Brehmer & Cranmer 2020; Cranmer et al. 2020). Currently, there are a wide range of available approaches and implementations that have been shown to be successful in a number of settings such as CMB data analysis (Cole et al. 2022), point source searches (Anau Montel & Weniger 2022), gravitational wave inference (Bhardwaj et al. 2023), and others, see e.g. (Dax et al. 2021; Hermans et al. 2021b; Montel et al. 2022; Gagnon-Hartman, Ruan & Haggard 2023; Karchev, Trotta & Weniger 2023). In general, the advantages of SBI techniques fall into three categories: (i) a consistent inference methodology for any forward simulator, irrespective of the complexity, stochasticity, or data dimensionality of the model, (ii) the possibility of extremely simulation efficient inference compared to traditional methods,[2] and (iii) the methods do not require an explicit likelihood to be written down, allowing for arbitrarily detailed physics simulations, and observational/detection models. The last point has interesting outlook for stellar streams as it allows for the possibility to significantly improve the modelling and to investigate the implications of e.g. selection effects, observation strategies, and instrument errors. This could have important implications for inference results based on e.g. small-scale structure in the observed streams or concrete features such as the GD1 spur and gaps (Carlberg & Grillmair 2013; Bonaca et al. 2018; de Boer et al. 2018; Price-Whelan & Bonaca 2018).

1.5 Key contributions

This work contributes in a number of ways to the problems and analysis challenges identified above. First, and most importantly, we develop and test a brand new analysis pipeline that leverages recent advances in SBI. We argue that the use of SBI to study stellar streams is motivated for a number of reasons. In particular, it allows one to make use of the highest fidelity modelling and observational models via the fact that it is an implicit-likelihood framework. It has also been shown in numerous settings to be highly simulation-efficient (Cole

[1] In this regard, the case of GD1 is interesting since there is some evidence that the observed density variations exhibit periodicity along the stream track consistent with the well-known epicyclic variations. See e.g. fig. 14 in Ibata et al. (2020) which constructs the power spectrum as a function of wavenumber along the stream track and highlights a clear peak at $k_s^{-1} \simeq 2.64\,\mathrm{kpc}$.

[2] This is not necessarily generic across the various methods, but has been observed empirically in a number of settings (Cole et al. 2022).

et al. 2022) compared to more traditional methods such as Markov Chain Monte Carlo (MCMC) (Mackay 2003; Foreman-Mackey et al. 2013).[3] This is of high relevance to the analysis of streams, since modelling of the complex and varied physics can be computationally costly, making sampling the posterior for large dimensional models typically infeasible. One way we overcome this in this work is to use a specific targeted (in the sense of analysing a particular observation) SBI algorithm known as truncated marginal neural ratio estimation (TMNRE) (Miller et al. 2022b), implemented within the framework of `swyft` (Miller et al. 2021, 2022b). Secondly, we also developed and will release a public code called `sstrax` for the modelling of stellar streams in the Milky Way. The current version of the code is designed to be highly modular and extendable for any aspect of streams modelling (e.g. the gravitational dynamics or tidal stripping). It is written in PYTHON but is highly accelerated through the use of `jax` (Bradbury et al. 2018), allowing for fast ($\mathcal{O}(1)$ s) sampling of realistic forward models. This speed is crucial for doing sampling on large dimensional models. Our implementation of the TMNRE algorithm, coupled to the `sstrax` modelling code will also be made publicly available in the package `albatross`.

1.6 Structure of the work

The rest of this work is structured as follows: In Section 2 we describe the physics behind the forward modelling of stellar streams, and highlight our numerical implementation in `sstrax`. Then, in Section 3, we describe the use of SBU for studying and analysing stellar streams, including a detailed explanation of the TMNRE algorithm. In Section 4, we demonstrate that our analysis pipeline can reliably perform parameter inference on *all* of the parameters in our forward model and discuss the sort of validation tests we can perform on the resulting posteriors. Finally, in Section 5, we present the key conclusions to the study as well as some outlook as to the relevant use cases and data analysis challenges.

2 MODELLING STELLAR STREAMS

Arguably one of the most challenging aspects for analysing stellar streams is balancing the complexity of the modelling with the ability to do full parameter inference without resorting to e.g. fixing a number of parameters. One of the key arguments we will make later in this work is that SBI can be a path towards a highly sample efficient analysis framework (Cole et al. 2022). This opens up the possibility for using higher fidelity forward models for the dynamics and observation of stellar streams. It is for this reason that we decided to simultaneously develop and test a new modelling code for stellar streams, `sstrax`, that is modular and designed to be extendable in all aspects with the aim to move towards highly realistic stream modelling for sampling tasks. For the purposes of this work, we have developed what we believe is a simulator that contains all the key elements for a robust proof-of-principle inference analysis. It will highlight the fact that the analysis and inference pipeline that we develop in later sections is not reliant on particularly symmetric or statistically simple (e.g. at the level of the data likelihood) models. We do note, however, that as far as the analysis methodology is concerned, *any* forward model could be used (introducing its own set of modelling assumptions, of course), including e.g. the current state-of-the-art models developed in `galpy` (Bovy 2014, 2015) or other works (Bowden et al. 2015; Erkal & Belokurov 2015a; Fardal et al. 2015; Bovy et al. 2017; Delos & Schmidt 2022).

In this section, we describe the key components to our modelling code, and discuss in each case some relevant improvements that could be made. The generation of a single stream is split broadly into five steps:

(i) *Cluster trajectory.* Given some current position $\boldsymbol{x}_c$ and velocity $\mathbf{v}_c$ for the disrupted cluster, we trace the trajectory back for some time $t_{\rm age}$ in the relevant gravitational potential to find the initial conditions.
(ii) *Cluster mass-loss.* We then solve an equation for the evolution of the mass of the cluster $M_c(t)$, due to e.g. tidal disruption events, given its trajectory from Step 1, the gravitational potential, and choices for the parameters in the mass-loss model.
(iii) *Star stripping times.* Given this mass-loss history, we can then generate a set of stripping times $\{t_i\}_{i=1..N_{\rm stars}}$ for stars released from the cluster. These are chosen to be a random sample from a probability distribution that is a normalized version of $\mathrm{d}M_c/\mathrm{d}t$.
(iv) *Stream stars evolution.* For each stripping time t_i, we generate initial conditions for a star released from the cluster and evolve the star forward in the gravitational potential for a time $(t_{\rm age} - t_i)$ before noting its final position and velocity.
(v) *Observation.* Given the full set of stream stars, we construct an observation by projecting to a co-ordinate frame relevant for the stream and accounting for errors in the measurements of e.g. the positions and proper motions of the stream stars. We also account for possible background contamination and misidentification that may occur when applying selection cuts.

We will discuss each step in detail below. A concrete example of each step of the analysis process is shown in Fig. 2 along with the mock observation used later in the case study.

2.1 Cluster trajectory

The first step in the modelling is to take the cluster position[4] $\boldsymbol{x}_c = (\boldsymbol{x}_c, y_c, z_c)$ and velocity $\boldsymbol{v}_c = (\boldsymbol{v}_{x,c}, \boldsymbol{v}_{y,c}, v_{z,c})$ at time $t = 0$ (today) and construct the trajectory for all times $t \in [-t_{\rm age}, 0]$. In other words, we project the current position and velocity backwards to find the initial conditions of the cluster a time $t_{\rm age}$ ago. To do so, we need to know the gravitational potential $\Phi(\boldsymbol{x}, t)$ and solve the equation $\ddot{\boldsymbol{x}}_c(t) = -\nabla\Phi(\boldsymbol{x}_c, t)$. In terms of implementation, we use the publicly available `diffrax` differential equation solver library (Kidger 2021), written in `jax` (Bradbury et al. 2018).

In principle, the gravitational potential $\Phi(\boldsymbol{x}, t)$ can include all contributions from, e.g. the Milky Way dark matter halo, baryonic structures, dark matter sub-haloes, dynamical clusters, or dwarf galaxies etc. In this work, we restrict our attention to a fixed, time-independent Milky Way potential $\Phi = \Phi_{\rm MW}(\boldsymbol{x})$ which consists of a dark matter halo, and baryonic disc and bulge components. Specifically, we choose the `MWPotential2014` implementation in `galpy` (Bovy 2015), whose parameters are given in table 1 of Ref. (Bovy 2015). We note, however, that it is trivial to include arbitrarily complex potentials in our modelling framework. One should also check the level of impact mild to strong mismodelling has in this regards if e.g. the true potential is not exactly the one with which the simulations are generated. One reason for this is that we do not need to analytically construct action-angle co-ordinates (although in principle, this could be possible numerically,

[3] In the current context, the application of MCMC techniques to the analysis of streams was pioneered in Varghese, Ibata & Lewis (2011).

[4] All of our dynamical modelling is performed in a Cartesian co-ordinate frame (x, y, z) with its origin at the galactic centre.

see e.g. Ibata et al. 2021b). Instead, we take advantage of `jax`-accelerated differential equation solvers to efficiently evolve the cluster and stars. With this choice, it is also simple to include time-dependent potentials that would arise from either the evolution of the Milky Way itself (Penarrubia et al. 2006; Buist & Helmi 2015; Hammer et al. 2023), or through interactions with dynamical objects such as dark matter sub-haloes or dwarf galaxies (Carlberg & Grillmair 2013; Erkal & Belokurov 2015b; Bonaca et al. 2018, 2020; Doke & Hattori 2022). These can be modelled without any additional approximations, and are represented simply by an additional term in the gravitational force. It would also be straightforward to let the parameters in the Milky Way potential vary and constrain them at the same time as the other model parameters.

2.2 Cluster mass-loss

Once we have the trajectory of the cluster $\boldsymbol{x}_c(t)$, we want to solve for the evolution of its mass as a function of time $M_c(t)$. This mass-loss typically occurs for a number of reasons, due to, e.g. disruption as a result of tidal forces, stellar evolution, or dissolution, see e.g. (Baumgardt 1998; Takahashi & Portegies Zwart 2000; Taylor & Babul 2001; Baumgardt & Makino 2003; Drakos et al. 2022). None the less, the vast majority of semi-analytic mass-loss models take the form (van den Bosch et al. 2018; Delos 2019; Drakos, Taylor & Benson 2020),

$$\frac{\mathrm{d}M_c}{\mathrm{d}t} = -\frac{f(M_c, r_t)}{\tau_{\mathrm{orb}}}, \tag{1}$$

where r_t is the instantaneous tidal radius, f is some model-dependent function of the cluster mass and tidal radius, and τ_{orb} is some characteristic time-scale. In this initial implementation of `sstrax`, we choose to work with a semi-analytic model given by (Baumgardt 1998),

$$\frac{\mathrm{d}M_c}{\mathrm{d}t} = -\left(\frac{\xi_0}{t_{\mathrm{rh}}}\right)\sqrt{1+\left(\alpha\frac{r_h}{r_t}\right)^3}\,M_c, \tag{2}$$

where ξ_0 and α are dimension-less parameters that in initial works were fitted to N-body simulations, r_h is the half-mass radius of the cluster, and t_{rh} is the relaxation time given by (Baumgardt 1998),

$$t_{\mathrm{rh}} = 0.138\,\frac{\sqrt{M_c}\,r_h^{3/2}}{\bar{m}\sqrt{G}\log(0.4N)}. \tag{3}$$

In this expression, $\bar{m}$ is the average mass of a star in the cluster, and $N(t) = M_c(t)/\bar{m}$ is the total number of stars in the cluster. In addition, r_t is the tidal radius, and is computed using (Bowden et al. 2015),

$$r_t(\boldsymbol{x}, t) = \left(\frac{GM_c(t)}{\Omega^2 - \mathrm{d}^2\Phi_{\mathrm{MW}}/\mathrm{d}r^2}\right)^{1/3}, \tag{4}$$

where Ω is the instantaneous angular frequency of the cluster around the galactic centre, $r = |\boldsymbol{x}|$, and we compute the second derivative of the potential using the autodifferentiation capabilities of `jax`.

This model quantitatively reproduces interesting features in the mass-loss such as the fact that more stars should be stripped near the pericentre of the orbit, which can introduce density variations that are totally separate from e.g. epicycles in the stream evolution (Kuepper et al. 2010, 2012; Ibata et al. 2020). Again, as in the case of the gravitational potential, this mass-loss model can be improved, either by generalizing the form, or through a modern calibration to high-resolution N-body simulations of cluster evolution (Baumgardt & Makino 2003; Loyola & Hurley 2013; Rossi, Bekki & Hurley 2016; Madrid et al. 2017; Banik & Bovy 2021; Stücker et al. 2023). Of course, there is a 'gold standard' approach which would be to perform N-body evolution in every simulation. However, we do not expect simulation efficiencies for this type of computation to drop significantly enough for this to become viable in parameter inference. As such, in any inference analysis, one will almost certainly have to resort to a semi-analytic form.

At the level of implementation, we solve this mass-loss differential equation numerically using `diffrax` (Kidger 2021), taking the (densely interpolated) cluster trajectory solution $\boldsymbol{x}_c(t)$ and initial cluster mass M_{sat} as input. As mentioned above, since we directly forward model the mass-loss, the code can be modified to use any form of $\mathrm{d}M_c/\mathrm{d}t$, including e.g. contributions due to impacts with sub-haloes or other transient interactions (Carlberg & Grillmair 2013; Erkal & Belokurov 2015b; Bonaca et al. 2018, 2020; Doke & Hattori 2022).

2.3 Stripping times

Once we have obtained the cluster mass $M_c(t)$ as a function of time t, we want to stochastically generate a set of stripping times $\{t_i\}$. These times define the moment the stars which will ultimately form the final stream are released. To go from cluster mass to stripping times, we identify $(-\,\mathrm{d}M_c(t)/\mathrm{d}t)$ as the instantaneous stripping rate. We could model this faithfully as an inhomogeneous Poisson process, however a simple approximate scheme, which we outline below, is sufficient for our purposes.

First, we introduce the average mass of a star $\bar{m}$ as a new parameter, although we note that this is a somewhat toy simulation parameter since real systems are known to not have monochromatic mass functions. We then compute the total number of stars that should be in the final stream as $N_{\mathrm{stars}} = \Delta M/\bar{m}$, where $\Delta M = (M_{\mathrm{sat}} - M_c(t=0))$ is the total mass-loss of the cluster.[5] Now, each of the N_{stars} stripping time can be sampled individually according to the distribution $(-\,\mathrm{d}M_c(t)/\mathrm{d}t)/\Delta M$. This can be done in a number of ways, but in `sstrax` we choose to construct the cumulative distribution function and sample uniformly from $U[0, 1]$ before projecting back to the t-space.

Note that this scheme does not explicitly use a distribution over star masses. Nevertheless, if one were to compute the differences of the heuristic cluster mass function between stripping times, one would obtain some mass distribution clustered around $\bar{m}$. The important point to realize is that we already make an approximation in using a continuous cluster mass $M_c(t)$. If the particular distribution of star masses becomes relevant, both the stripping and the cluster mass modelling would have to be replaced by a more realistic framework. This could be achieved, for example, by sampling the next star mass $m_s \sim p(m_s)$ (Schulz, Pflamm-Altenburg & Kroupa 2015), treating the cluster mass-loss in equation (2) as the instantaneous event frequency of an inhomogeneous Poisson process, and only reducing the cluster mass by discrete steps m_s whenever a star is released. It would be interesting to explore the implications of these two effects (i.e. star mass distributions and modelling the cluster mass via an instantaneous Poisson process instead of a continuous function) in the context of limits derived from density variations along the stream tracks, see e.g. (Banik et al. 2018, 2021b; Hermans et al. 2021b). We have also assumed that the cluster system is collision-less, whereas in some systems populations of e.g. dark masses (see e.g. Vitral et al. 2022) towards their centre or accretion of halo clusters (see Mackey

[5]For context, in a stream such as GD-1, there are typically around 1000 stars reported as probable members (with some variation according to the detection method). In the mock observation we present here, there are $N_{\mathrm{stars}} = 968$ stars, so represents qualitatively the same scale of system.

et al. 2019; Malhan et al. 2019) could break this assumption and should be modelled properly before targeting real data.

Finally, note that our stripping process is stochastic and can therefore lead to different realizations of the density profile along the stream track if the same stream is generated multiple times.

2.4 Stream star evolution

The final dynamical step for generating the stream is quite simple – we just need to release stars from nearby the cluster at the times t_i and evolve them forward in the same gravitational potential $\Phi(\boldsymbol{x}, t)$[6] as the cluster until today $t = 0$. The only choice left to be made is one regarding the initial conditions for the stars, which we choose in accordance with observations made in N-body simulations of tidally disrupted clusters (Baumgardt & Makino 2003; Loyola & Hurley 2013; Rossi et al. 2016; Madrid et al. 2017; Stücker et al. 2023). It has been shown that the majority of stars escape from near one of the two Lagrange points $\boldsymbol{x}_{1,2} = (1 \pm (r_t/r))\boldsymbol{x}_c$ of the cluster (Varghese et al. 2011; Bowden et al. 2015) (one on either side of the radial line joining the galactic centre and the cluster centre), where $r = |\boldsymbol{x}_c|$.

In the `sstrax` implementation, we generalize this slightly and introduce three additional parameters: $\lambda_{\rm rel}$, $\lambda_{\rm match}$, and $p_{\rm near}$. Respectively, these describe how far away from the cluster the star is released, i.e. $\boldsymbol{x}_{\rm rel} = (1 \pm \lambda_{\rm rel}(r_t/r))\boldsymbol{x}_c(t_i)$, at what distance the velocity matching is done (specifically, the velocity is matched so that the angular velocity of the star and the cluster agree at a distance $\boldsymbol{x}_{\rm match} = (1 \pm \lambda_{\rm match}(r_t/r))\boldsymbol{x}_c(t_i)$), and finally the probability $p_{\rm near}$ of being released from the closer Lagrange point. Finally, to model the velocity dispersion of the cluster itself, we choose the initial velocity of the star to be this matching velocity plus an additional random vector $\Delta\boldsymbol{v}$ sampled on the unit sphere and rescaled by a factor $\sqrt{3}\sigma_v$, where σ_v is the velocity dispersion.

In much the same way as the mass-loss model, the most realistic way to actually model this process would be to account for the full dynamics inside the cluster via some N-body approach. For the same reason, this is still too costly for parameter inference tasks, so a semi-analytic approach like the one above needs to be used. Again, and in line with the prescription we chose for the mass-loss, since we directly forward model the evolution of the stars, the generation of these initial conditions can be tuned arbitrarily to either analytic expectations, or some new high-resolution simulations. In any case, the analysis pipeline will remain the same.

2.5 Observational model

An important aspect of SBI approaches is that the forward model *must* also include the detector response, observational model, or noise generation. This is in contrast perhaps to traditional approaches where typically some clean signal output of the forward model is input into an explicit data likelihood. In practice, the statistics results should be identical in either formulation. In the context of stellar streams, given some final stream configuration $\{\boldsymbol{x}_\star^i, \boldsymbol{v}_\star^i\}_{i=1\ldots N_{\rm stars}}$, we need to model, (i) the observational measurement errors, (ii) the detection of the stream in the sky, and (iii) the contaminating background of other stars.

In this work, we develop a simple initial observational model, meant mostly as a proof of principle. Specifically, we assume that the stream has been 'detected' through some form of selection cuts and vetoes in survey data (Malhan & Ibata 2018; Huang et al. 2019; Borsato, Martell & Simpson 2020; Shih et al. 2021; Shih, Buckley & Necib 2023). We use this to define an observational window which we choose to focus on (i.e. we do not model the full sky). In the rest of the analysis, we will be focusing on a mock stream that is supposed to resemble the GD1 stream (Grillmair & Dionatos 2006; Eyre 2010; Price-Whelan & Bonaca 2018). There are a standard set of co-ordinates used in the literature (Koposov et al. 2010) to describe the phase-space structure of this stream. Specifically, there are two angle co-ordinates (ϕ_1, ϕ_2) which are approximately aligned with the stream track at $\phi_2 \simeq 0$ deg, the corresponding proper motions $(\mu_{\phi_1}, \mu_{\phi_2})$, and radial distances and velocities $(d, v_{\rm rad})$. The definitions for these can be found in Appendix A.

Given these definitions, to construct the observation from the list of stream stars, we first define the region of interest in the sky/velocity phase space, i.e. we ignore all stars with $(\phi_1, \phi_2, ..., v_{\rm rad}) \notin [\phi_1^{\rm min}, \phi_1^{\rm max}] \times \cdots \times [v_{\rm rad}^{\rm min}, v_{\rm rad}^{\rm max}]$. Then, we add random observational errors to the values generated by `sstrax` via sampling e.g. $\phi_1^{\rm obs} \sim \mathcal{N}(\phi_1, \delta\phi_1)$. Finally, we model two aspects of stream detection and selection effects. In particular, we assume that we have some selection efficiency $\epsilon_{\rm sel}$ that measures how often we accidentally miss a star in a given detection algorithm that should have been correctly classified as part of the stream (Malhan & Ibata 2018; Huang et al. 2019; Borsato et al. 2020; Shih et al. 2021; Shih et al. 2023). We also model the fact that there can be contamination from the background stars that are not part of the stream, but are none the less, not removed by the detection algorithm and are in the observing window (Huang et al. 2019). This is quantified by assuming there is some number $N_{\rm background}$ stars, of which we are able to successfully remove $(1 - \epsilon_{\rm background})$ per cent via the selection process. We then distribute $N_{\rm background}\epsilon_{\rm background}$ stars uniformly across the observational windows to model the background contamination. Finally, we bin the remaining data into three channels of size $(N_{\rm bins}^x, N_{\rm bins}^y)$ each: (ϕ_1, ϕ_2), $(\mu_{\phi_1}, \mu_{\phi_2})$, and $(d, v_{\rm rad})$.[7] All the choices for the particular values of the observational model described here are given in Table 2.

As in the other components, there is significant room for more detailed modelling. For example, we know just from looking at *Gaia* data that the background stars will not be uniformly distributed across the sky (Gaia Collaboration 2018, 2021; Boubert & Everall 2020), with higher concentrations near the galactic centre. Similarly, the efficacy of the sort of selection criteria or cuts that are applied based on e.g. metallicity or proper motions are likely at least stream- and sky location-dependent. The extent to which this impacts the inference is a different question, and something that we can actually test in our framework by modifying the observational model.

2.6 Acceleration with `jax`

Making the decision to directly forward model the evolution of the stream, rather than construct either some effective description (De-

[6] In principle, one can also evolve them in the gravitational potential of the cluster as well as the Milky Way, but we found that this was indistinguishable at the level of inference results. It is also well-known that we do not need to include the self-gravity of the stream itself (Delos & Schmidt 2022).

[7] It is worth noting that for streams with relatively low stellar counts, binning the data may not be the most appropriate data representation. Arguably one of the key benefits of the SBI paradigm, however, is that if a more relevant data choice can be made/simulated, then the statistical implications will be automatically taken into account. This final point also holds if e.g. some aspects of the data are unavailable for some reason (such as the radial positions and velocities in the current context).

los & Schmidt 2022), or accelerate the dynamical solutions through action-angle co-ordinate constructions opens up the possibility for far more general simulation frameworks. On the other hand, it is also potentially much more computationally intensive, e.g. if we include the effect of a large population of subhaloes in the future. This is compounded by the additional simulation budget that is potentially required to perform inference on the large number of parameters any augmentation of the model can introduce.

As such, an important component of our implementation is its computational efficiency. We have achieved this by using the `jax` framework (Bradbury et al. 2018), which allows for just-in-time compilation caching and highly optimized custom vectorization.

3 SIMULATION BASED INFERENCE FOR STELLAR STREAMS

In this section, we will give a brief review of general SBI methods before describing the specific implementation we will use in this work. We will end the section by presenting some of the algorithm design choices that are relevant to stellar stream analysis.

3.1 Overview of simulation-based inference

Recently, there have been significant advances in high fidelity physics simulations, and machine learning techniques for processing complex data structures, alongside the emergence of increasingly challenging data analysis problems. This has led to the rapid development of 'SBI' as a competitive alternative to traditional techniques as far as scalability, model realism, and unbiased analysis pipelines are concerned (Brehmer & Cranmer 2020; Cranmer et al. 2020). At its heart, the field of SBI asks: *given some forward model or simulator, can we perform efficient and correct Bayesian inference?* Ultimately, the goal of SBI is to develop a robust statistical pipeline that can make use of the most realistic and state-of-the-art modelling tools.

To be more concrete, suppose we have some forward model $p(x, \theta)$ that takes the model parameters θ – which could be a range of physical parameters, effective model components, nuisance parameters etc. – to some data x that resembles the real observed data x_0. In a Bayesian context, we sample θ from some chosen[8] prior distribution $p(\theta)$ so that the forward model takes the form $p(x, \theta) = p(x|\theta)p(\theta)$. This expression is at the heart of simulation-based methods, since it formally represents the notion that 'running your simulator' is the same as sampling from the (simulated-)data likelihood $p(x|\theta)$. Indeed, this is the origin of the terms 'likelihood-free' or 'implicit likelihood' inference to describe SBI (Brehmer & Cranmer 2020; Cranmer et al. 2020). These descriptions are supposed to convey the distinction between analytically evaluating some expression to compute the likelihood $p(x|\theta)$ and sampling from it.

To understand the different ways in which SBI methods approach the Bayesian inference problem, it is useful to briefly review how the forward model fits into Bayes' theorem. As far as scientific conclusions are concerned, we are typically[9] interested in computing the posterior $p(\theta|x)$ of the parameters given some data x,

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}. \tag{5}$$

[8]Ideally with some physical motivation for the ranges chosen, or some maximally uninformative choice otherwise.

[9]Of course, there are use cases e.g. in model comparison or goodness-of-fit tests (Spurio Mancini et al. 2022), where computing other quantities such as the data evidence or maximum likelihood is more relevant.

Here, as above, $p(x|\theta)$ is the data likelihood, $p(\theta)$ is the prior over our parameters $\theta = (\theta_1, \cdots)$, and $p(x)$ is the evidence. Given this setup, there are various ways that SBI algorithms tackle posterior estimation given the ability to sample from the forward model $(x, \theta) \sim p(x, \theta) = p(x|\theta)p(\theta)$. Specifically, these can be categorized as follows (with the method we use highlighted in bold):

(i) *Neural posterior estimation (NPE).* In NPE (Papamakarios & Murray 2016; Zeghal et al. 2022), the goal is to directly estimate the posterior distribution $p(\theta|x)$ by representing it as some flexible parametrized probability density. This has been applied successfully in a number of contexts, e.g. gravitational wave analysis (Dax et al. 2021) and open source implementations are available (Tejero-Cantero et al. 2020).

(ii) *Neural likelihood estimation (NLE).* In contrast, NLE (Alsing et al. 2019; Papamakarios, Sterratt & Murray 2019) attempts to construct an estimator for the (simulated-)likelihood function itself $p(x|\theta)$. This can then be used to carry out standard inference techniques such as MCMC (Mackay 2003; Foreman-Mackey et al. 2013) or nested sampling (Skilling 2006; Handley, Hobson & Lasenby 2015; Ashton et al. 2022) and generate samples from the posterior.

(iii) *Neural ratio estimation (NRE).* Finally, NRE (Hermans, Begy & Louppe 2019; Durkan, Murray & Papamakarios 2020; Rozet & Louppe 2021; Delaunoy et al. 2022; Miller et al. 2022b) considers the ratio $p(x|\theta)/p(x)$ appearing on the right-hand side of equation (5). This particular approach will be the focus of this work, in the form of an algorithm known as TMNRE (Miller et al. 2022b), implemented within the framework of `swyft` (Miller et al. 2021).

3.2 The TMNRE algorithm

We will now focus on the specific implementation of SBI used in this work. This is known as TMNRE (Miller et al. 2022b), and is implemented in the `swyft` software (Miller et al. 2021). We have summarized the method in Fig. 1 for reference, however, there are a number of features we wish to emphasize in terms of its applicability to stellar streams.

(i) *Targeted inference.* TMNRE is both a 'targeted' and 'sequential' algorithm in the sense that it performs inference on a specific target observation x_0 (as opposed to amortizing over all possible model outputs) over a number of discrete rounds. In each round, the prior is truncated based on inference in the current round (see description below) to avoid simulating in parameter regions which do not contribute significantly to the likelihood ratio, given the fixed target observation.

(ii) *Marginal posteriors.* There are a number of quantities of interest in Bayesian inference, including the full joint posterior $p(\theta|x_0)$ given some observation x_0, the evidence of a particular observation $p(x_0)$, or marginalized posteriors[10] $p(\theta_i|x_0)$ for some individual parameter θ_i or subset of parameters in θ. In TMNRE, a significant portion of the achieved simulation efficiency arises due to the fact that we directly estimate the marginal posterior, rather than marginalizing over samples from the full joint distribution.

In combination, these two properties are the key to achieving a highly simulation efficient inference strategy. For more discussions along these lines, see e.g. works discussing the application of TMNRE to CMB (Cole et al. 2022) and gravitational wave analyses (Bhardwaj et al. 2023).

[10]In the strict technical sense that $p(\theta_i^\star|x) = \int \mathrm{d}^n\theta\; p(\theta|x)\delta(\theta_i - \theta_i^\star)$

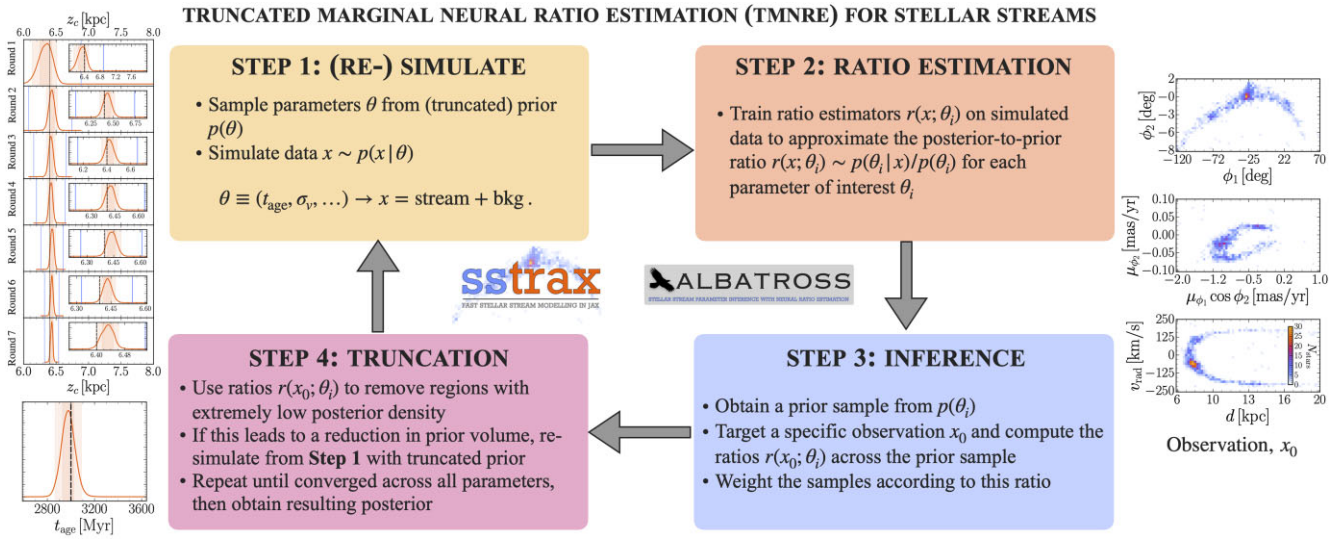

Figure 1. A schematic illustration of the data analysis pipeline developed in this work. We use the TMNRE algorithm (see Section 3) to carry out parameter inference on Milky Way stellar streams (see Section 4), using our new modelling code `sstrax` (see Section 2). We also publicly release the `albatross` analysis code.

Although the details of the method can be found in the original literature (Miller et al. 2022b), it is useful to give a brief overview of the setup of the ratio estimation problem. This will highlight the features described above and how they will be beneficial for the analysis of stellar streams. The goal of TMNRE is to estimate the following ratio,

$$r(x;\theta) = \frac{p(x|\theta)}{p(x)} = \frac{p(\theta|x)}{p(\theta)} = \frac{p(x,\theta)}{p(x)p(\theta)}, \tag{6}$$

where the last two equalities follow from an application of Bayes' theorem in equation (5) and the definition of the joint distribution $p(x, \theta) \equiv p(x|\theta)p(\theta)$. In other words, access to the ratio $r(x; \theta)$ is equivalent to estimating (i) the likelihood-to-evidence ratio $p(x|\theta)/p(x)$, (ii) the posterior-to-prior ratio $p(\theta|x)/p(\theta)$ which will be used for parameter estimation, and (iii) the joint-to-marginal ratio $p(x, \theta)/p(x)p(\theta)$ which will be the technically important form to perform ratio estimation in practice.

If we focus on the last form, $r(x; \theta) = p(x, \theta)/p(x)p(\theta)$, we can make the observation that given a set of simulations $\{(x, \theta)\}$ from our forward model $p(x, \theta) = p(x|\theta)p(\theta)$, we can construct two distinct classes of sample. The first is simply a sample from the full joint distribution $p(x, \theta)$, which amounts to picking an individual simulation pair (x, θ). The second is a sample from the combined marginal distribution $p(x)p(\theta)$ which can be obtained by picking two random samples, then taking x from one, and θ from the other. Having these two distinct distributions is the origin of ratio estimation as a binary classification task[11] – it asks the question *given a pair (x, θ), did θ generate x?* Intuitively, the relative precision of the posterior distribution in this case reflects how difficult it is to discriminate between joint and marginal samples. For instance, the larger the observational error is, the more overlap there will be between these two classes and the posterior will be wider.

More formally, we can frame this binary classification task and ratio estimation as an attempt to optimize (specifically minimize) the following loss function[12] (Miller et al. 2021),

$$\mathcal{L}[f_\phi] = -\int \mathrm{d}x\mathrm{d}\theta\; p(x,\theta)\ln\left(\sigma(f_\phi(x,\theta))\right) + p(x)p(\theta)\ln\left(1-\sigma(f_\phi(x,\theta))\right). \tag{7}$$

Here $\sigma(x) = [1 + \exp(-x)]^{-1}$ is the sigmoid function, and $f_\phi(x, \theta)$ is the classifier with some set of free parameters ϕ that should be optimized. One of the key justifications for the correctness of TMNRE as an inference algorithm is to realize that this loss can actually be minimized analytically. In particular, one can show that the optimal classifier is given by $f^\star_\phi(x, \theta) = \ln r(x; \theta)$ (Miller et al. 2021). In other words, if one can successfully minimize the loss in equation (7), then *one directly obtains the posterior-to-prior ratio r(x; θ).*

Practically, this is where the 'N(eural)' part of TMNRE is relevant, especially for very high dimensional data/parameter spaces. Modern machine learning methods, architectures, and hardware allow for very flexible parametrizations of the classifier f_ϕ, and there is a well established methodology to optimize their parameters ϕ (also more commonly called their weights). As far as the analysis of stellar streams is concerned, this gives us access to data representations that are as close as possible to real data from e.g. *Gaia*, without any need for compression into hand-crafted summary statistics, spline fits, or similar data reductions. In this work, the task of optimizing is achieved through the use of the software `swyft`, which is built on top of `pytorch`.

With (neural) ratio estimation set up this way, we can now see how to directly estimate marginal posteriors in this framework. Suppose we wish to estimate the fully marginalized posterior for a single parameter[13] θ_i in θ, then we can start by taking our full suite of simulations $(x, \theta) \sim p(x, \theta)$ which (crucially) vary *all* parameters θ. Then, however, instead of constructing the loss based on all

[11] This can actually be generalized in interesting ways to form multiclass classification problems that are applicable to e.g. hierarchical models (Miller, Weniger & Forré 2022a).

[12] This is nothing other than the binary cross-entropy for a classifier that tries to discriminate between joint $(x, \theta) \sim p(x, \theta)$ and marginal $(x, \theta) \sim p(x)p(\theta)$ samples.

[13] Or any subset of parameters $(\theta_{i_1}, \theta_{i_2}, \ldots, \theta_{i_k})$ to generate the k-dimensional marginal posterior $p(\theta_{i_1}, \ldots, \theta_{i_k}|x)$. We will give explicit examples for $k = 2, 3$ in Section 4.

parameters, we can only 'show' the single parameter θ_i. This is equivalent to replacing $\theta \to \theta_i$ in equation (7) above. Importantly, however, the analytic arguments will still hold and allow us to obtain directly the marginal posterior-to-prior ratio $p(\theta_i|x)/p(\theta_i)$. In contrast to e.g. MCMC where this marginalization is performed *after* obtaining samples from the posterior, we implicitly marginalize by varying all parameters in the simulations simultaneously, but constructing the marginal posterior directly rather than via the joint. As far as analysing stellar streams is concerned, this is not just a useful trick for quickly obtaining the marginal posteriors, but is crucial in making the algorithm simulation efficient. Looking forwards, if the goal is to perform inference with extremely high fidelity stream simulations in order to extract the maximum possible information from the data, analysis methods that break the traditional scaling of sampling algorithms such as MCMC or nested sampling will be vital.

The final aspect to discuss before we summarize the algorithm and the design choices relevant to stellar streams is the truncation process that allows us to target a particular observation x_0. As far as simulation efficiency is concerned, the idea behind truncation is to minimize the number of simulations performed in regions where there is extremely low posterior density, since, by definition, they provide almost no information about the parameter estimation problem. Formally we achieve this by performing the inference sequentially in several rounds. In each round, we generate a set of simulations $(x, \theta) \sim p(x, \theta)$ from the full model. Then we train and optimize our classifiers $f^i_\phi(x, \theta_i)$ for the parameters of interest from which we can obtain marginal posteriors on each parameter $p(\theta_i|x_0)$ for some specific target observation x_0. This will highlight regions of parameter space where the posterior density for θ_i is both very high, and of course other regions where the density is low, indicating that given the observation x_0, this particular set of parameters is unlikely. We use these latter regions to truncate our prior region by imposing the condition $r_i(x_0, \theta_i) < \epsilon$ on the estimated ratios.[14] Then, we re-simulate by sampling from this truncated prior, repeat the inference and then truncate again. Eventually, once the posteriors converge to the level of statistical uncertainty, the truncation will just return the restricted prior and the algorithm will terminate. This truncation process is highlighted below in Fig. 5.

In summary, the TMNRE algorithm splits into four steps that are highlighted in the schematic shown in Fig. 1:

(i) **Step 1:** Sample a set of simulations[15] from the full forward model $(x, \theta) \sim p(x, \theta) = p(x|\theta)p(\theta)$.

(ii) **Step 2:** Train a set of classifiers $f^i_\phi(x, \theta_i)$ to obtain an estimate of the ratio $r_i(x; \theta_i) = p(\theta_i|x)/p(\theta_i)$.

(iii) **Step 3:** Use this trained ratio to obtain estimates of the marginal posteriors $p(\theta_i|x_0)$ for a specific target observation x_0.

(iv) **Step 4:** Take these marginal posterior distributions and derive bounds on the prior region to truncate for the next round of inference by imposing the condition $p_i(\theta_i|x_0)/\max_{\theta_i} p_i(\theta_i|x_0) < \epsilon$.

(v) Repeat from **Step 1** until the truncation procedure stabilizes, then take the final round of inference as the set of posteriors $p(\theta_i|x_0)$ and terminate the algorithm.

3.3 Design choices for stellar streams

In order to use the TMNRE algorithm in practice, we must make a number of design choices. These include (i) building or using a pre-implemented forward model that generates the data x (here a representation of the stellar stream) given parameters θ, (ii) designing a neural network architecture that is able to efficiently process the data format of x and θ, (iii) making choices for the prior distributions over the parameters θ, and (iv) choosing the hyperparameters relevant to the TMNRE algorithm.

3.3.1 Forward simulator

To generate stellar stream simulations, we use the implementation of our modelling approach described in detail above (see Section 2). To very briefly recap, we solve for the full evolution history of the stream including e.g. the orbit-dependent tidal stripping in a framework that can accommodate for any time-dependent or time-independent gravitational potential. In addition, we develop a simple observational model that is supposed to represent experimental and statistical uncertainties at the level of a current survey. This is implemented in the `jax`-accelerated modelling code `sstrax`, which we couple directly to the `swyft` software (Miller et al. 2021) in our analysis code `albatross`. The parameters that we vary in this analysis $\theta = (t_{\rm age}, M_{\rm sat}, \dots)$ are described in Tables 1 (stream modelling) and 2 (observation model). An example output of our simulator (and the case study that we investigate below) is shown in Fig. 2.

3.3.2 Inference network

As discussed above, the main aim of (neural) ratio estimation is to design a procedure that can reliably train a classifier, or set of classifiers $f^i_\phi(x, \theta_i)$ to distinguish between joint and marginal samples (Hermans et al. 2019; Durkan et al. 2020; Miller et al. 2021; Rozet & Louppe 2021; Delaunoy et al. 2022). To do this in practice, we need a flexible way to parametrize f_ϕ, and although there are arguments from e.g. the loss function in equation (7) that any flexible enough parametrization (i.e. just having enough trainable parameters in ϕ) will be able to optimize the loss, in reality this is only in some infinite training data limit. As such, we should try to make sensible design choices regarding the network to take full advantage of the known structure and physics associated to our signal and data format. Empirically, making physics-informed choices at this stage leads to huge increases in performance, robustness, simulation efficiency, and the general applicability of the method.

In our case of stellar streams, the signal is a collection of stars and their properties (positions, velocities, perhaps even metallicity etc.). As described in Section 2, we focus on the phase-space information in this work, including e.g. the selection procedure in our observational model. As part of our forward model, we chose to bin the data into a 3-channel image to preserve the spatial structure and morphology of the signal. This data format is then well suited for the application of standard image processing network structures.

More precisely, we know that a stream binned at a given resolution can have structure on a range of scales. For instance, the large-scale orbit of the stream is typically governed by e.g. the ambient gravitational potential, as well as the initial conditions of the cluster. On the other hand, the smaller scale features (gaps, spurs etc.) are more likely

[14] Of course, this will introduce a slight error in the estimate of the marginal posterior proportional to $\Delta p(\theta_i^\star|x_0) \sim \int_{\Gamma(\epsilon)} {\rm d}^n\theta\; p(\theta|x_0)\delta(\theta_i - \theta_i^\star)$, where $\Gamma(\epsilon)$ is the region excluded by the truncation procedure. However, it is exactly in this region where the joint posterior density is (necessarily) low, and as such, the error induced is small and strictly controlled by ϵ. To be conservative, we typically choose $\epsilon \sim 10^{-5}$, which corresponds to exclusion at around the 4.5σ level for a Gaussian distribution (Miller et al. 2022b). Provided ϵ is not too large, any other choice should not change our results at all, only affecting the time that the algorithm takes to converge.

[15] Note that this step can be fully parallelized, something that is implemented directly in `albatross`.

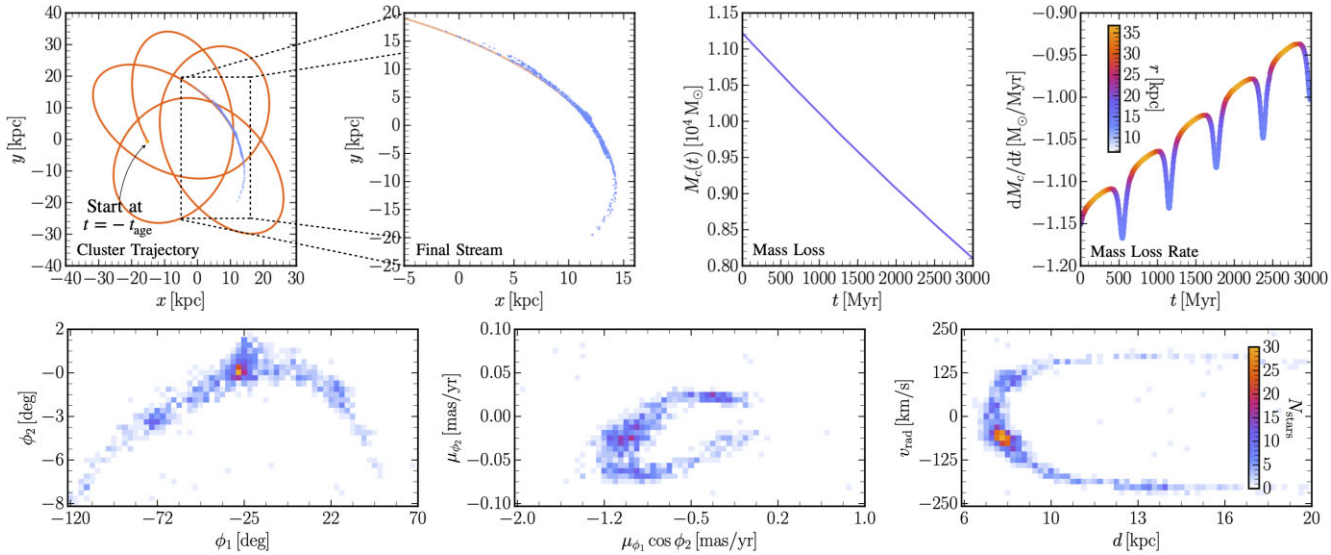

Figure 2. Upper panels: Illustration of the various modelling steps (finding the cluster trajectory, mass-loss history, final stream formation etc.) described in Section 2. Lower panels: Example mock observation generated using the `sstrax` modelling code with the parameters in Table 1. Analysed as a case study in Section 4.

Table 1. Parameters, prior range choices, and injection values for the stream model parameters described in Section 2.

Parameter	Prior range	True value
(Log of) Initial cluster mass $\log_{10}(M_{sat}/M_\odot)$	[3.0, 4.5]	4.05
Cluster velocity dispersion σ_v	[0.1, 5.0] km s^{-1}	1.1
Cluster final pos. $\boldsymbol{x}_{sat} = (x_c, y_c, z_c)$	Stream dependent*	(11.8, 0.79, 6.4)
Cluster final vel. $\boldsymbol{v}_{sat} = (v_{x,c}, v_{y,c}, v_{z,c})$	Stream dependent*	(109.5, −254.5, −90.3)
Stream age t_{age}	[500, 5000] Myr	3000
Release distance parameter λ_{rel}	[0.1, 2.0]	1.405
Release velocity parameter λ_{match}	[0.1, 2.0]	1.846
Stripping asymmetry p_{near}	[0, 1]	0.5
Mass-loss prefactor ξ_0	$[10^{-4}, 10^{-2}]$	0.001
Mass-loss parameter α	[10, 30]	20.9
Half-mass radius r_h	$[10^{-4}, 10^{-2}]$ pc	0.001
Average stellar mass $\bar{m}$	[1.0, 20] M$_\odot$	3

Note. *In particular, we choose these parameters to span the observational window of interest for an individual observation. In the analysis presented in Section 4, we choose the priors ([10, 14], [0.1, 2.5], [6, 8]) and ([90, 115], [− 280, −230], [− 120, −80]) on the cluster position and velocity respectively.

Table 2. Choices for observational model parameters described in Section 2.

Observation model parameter	Value
Observing window ϕ_1	[−120, 70] deg
Observing window ϕ_2	[−8, 2] deg
Observing window $\mu_{\phi_1} \cos\phi_2$	[−2, 1] mas yr^{-1}
Observing window μ_{ϕ_2}	[−0.1, 0.1] mas yr^{-1}
Observing window d	[6, 20] kpc
Observing window v_{rad}	[−250, 250] km s^{-1}
Number of bins	[64, 32]
Observational error $\delta\phi_1$	0.001 deg
Observational error $\delta\phi_2$	0.15 deg
Observational error $\delta\mu_{\phi_1} \cos\phi_2$	0.1 mas yr^{-1}
Observational error $\delta\mu_{\phi_2}$	0.0 mas yr^{-1}
Observational error δd	0.25 kpc
Observational errors δv_{rad}	5 km s^{-1}
Stream selection success rate $\epsilon_{sel.}$	95 per cent
Background stars in window $N_{background}$	10^6
Background contamination rate $\epsilon_{background}$	10^{-3} per cent

to be impacted by the dynamical evolution history, tidal stripping, or interactions with perturbers. The aim is to analyse both of these classes of signal simultaneously, and as such we should choose a network architecture accordingly. In particular, with this observation, it is simple to see that applying e.g. a standard convolutional network which applies the same kernel to each part of the image identically would be a poor choice and unlikely to be able to simultaneously extract the small-scale and large-scale information. With this in mind, we choose to use the well-known `unet` architecture (Ronneberger, Fischer & Brox 2015), which is well suited for image analysis and segmentation. It is designed to simultaneously analyse the image at a larger scale, before performing follow up analysis on each identified segment and then combining the results.

There is another part of the inference network (a full description and network diagram can be found in Appendix B) which performs the ratio estimation. Schematically, one can understand the overall structure as first performing some data compression through the `unet` and a small linear network to extract an optimal set of summary statistics. Then, these summary statistics (which are *automatically* learned and optimized during the training), are passed

Table 3. Choices for the hyperparameters and settings for the TMNRE algorithm in this work, as described in Section 3.

TMNRE setting	Value
Number of rounds	7⋆
Simulation schedule	30k, 30k, 30k, 30k, 30k, 60k, 150k
Bounds threshold ϵ	10^{-5}
Noise shuffling	`True`
Min./Max. training epochs	0/50
Early stopping patience	20
Initial learning rate	5×10^{-4}
Training/Validation batch size	64/64
Train : Validation ratio	0.9 : 0.1

Note. ⋆This is the minimum number of rounds, if the algorithm has not converged, we continue rounds of inference until the truncation procedure terminates.

to the default ratio estimator implemented in `swyft` along with the model parameters θ. All the details regarding the implementation can be found in the `albatross` library. In terms of specificity, we expect the network to be broadly applicable to the analysis of any stream model or observation, since it only assumes that the signal has structure on various scales.

3.3.3 *Prior choices*

The prior choices for all the parameters of interest are shown in Table 1. They are chosen to either represent our knowledge about the physics from current astrophysical observations or simulation results (e.g. the mass-loss parameter α), or to be maximally uninformative. An example of the latter case are the cluster position and velocity priors which are chosen in the first instance to span the full observational window.

3.3.4 *TMNRE hyperparameters*

There are a number of hyperparameters that need to be set when one runs the TMNRE algorithm. Broadly these can be categorized as either parameters that control the network training process, or parameters specific to the TMNRE algorithm. For the inference and analysis detailed in this work, the particular choices can be found in Table 3, as well as in the example configuration files supplied with `albatross`. Briefly, the training parameters describe how long to train the network for (min./max. training epochs), how many epochs to wait before the validation loss should decrease again (early stopping patience),[16] the split between training and validation data (Train : Validation ratio), and the batch sizes shown to the network during training (training/validation batch size). The TMNRE settings consist of the minimum number of rounds (number of rounds), the schedule for the number of simulations per round (simulation schedule),[17] and the threshold for truncation (ϵ). Finally, we have the 'noise shuffling' setting, which breaks down the data x into the stream and background components. In a given batch it then randomly permutes the background elements, essentially showing the network a brand new example (with the same signal component) every epoch at zero simulation cost. We found this to be an extremely effective way of reducing the possibility of overfitting, especially in the early rounds where we have small simulation batches.[18] Indeed, this strategy should be applicable to any additive noise model, see for example its application to gravitational wave data analysis (Bhardwaj et al. 2023).

In this section we have discussed the broad field of SBI and a specific algorithm, known as TMNRE that we have used to build our data analysis pipeline. We argued that the targeted and marginal-focused approach could be a key advantage for stellar stream analysis, including the resulting simulation efficiency, statistical robustness, and the opportunities for increased model complexity. Finally, we discussed some of the design choices that need to be made in order to successfully apply TMNRE to a given problem. In the next section, we will present a case study for a mock stream to illustrate the application of our modelling and analysis strategy.

4 RESULTS: GD1-LIKE CASE STUDY

Now that we have set up the framework of SBI, and specifically described the application of the algorithm to the analysis of stellar streams, we can present a case study to highlight its functionality. In this section, we will illustrate the full analysis and validation of a mock observation that is generated using our stellar streams modelling code `sstrax`. This is in order to have full control over the reconstruction of the parameters, as well as the physics input to the model. Of course, the longer term goal is to analyse current and future state-of-the-art spectroscopic and photometric surveys, such as *Gaia* and the Vera Rubin observatory (Abell et al. 2009; Gaia Collaboration 2018, 2021; Bechtol et al. 2019).

4.1 Case study description

Perhaps the most well-studied and well-observed Milky Way stellar stream is the GD1 stream (Grillmair & Dionatos 2006; Eyre 2010; Price-Whelan & Bonaca 2018). It was first identified in the SDSS catalogue in the early 2000s (Grillmair & Dionatos 2006; Eyre 2010), but recently it has been observed by e.g. *Gaia* in significantly more detail (Price-Whelan & Bonaca 2018; Gaia Collaboration 2018, 2021). Indeed, observations are currently at the level where individual substructures (e.g. the so-called 'gaps' and 'spur') are reasonably well resolved (de Boer et al. 2018; Price-Whelan & Bonaca 2018). In some sense, the purpose of developing our analysis method is to take full advantage of these improvements and perform

[16] During the training, we track both the current loss on the training data set, as well as the loss evaluated on some separate validation set. Looking for good performance on the validation data set is typically a good strategy to avoid overfitting, and therefore we use it as a metric to indicate whether we are starting to overfit to the training data. The early stopping criterion waits for a specified number of passes through the training data (or epochs), over which the validation loss has not decreased before terminating the training. It then re-initializes the network parameters to the state where the minimum validation loss was observed.

[17] It is typically the case that in the early rounds, only a small number of parameters are meaningfully constrained, and so it is more efficient to have a more reduced simulation batch, truncate, and then re-simulate again. In the last rounds, however, to achieve the correct level of statistical precision, significantly more training data is required.

[18] As an aside, we also explicitly tested that resampling the stripping times and regenerating the 'same' stream (at least statistically) also lead to improvements in the smoothness of the training and validation losses, but importantly did not affect the precision of the posteriors. This approach was especially effective for small simulation batches.

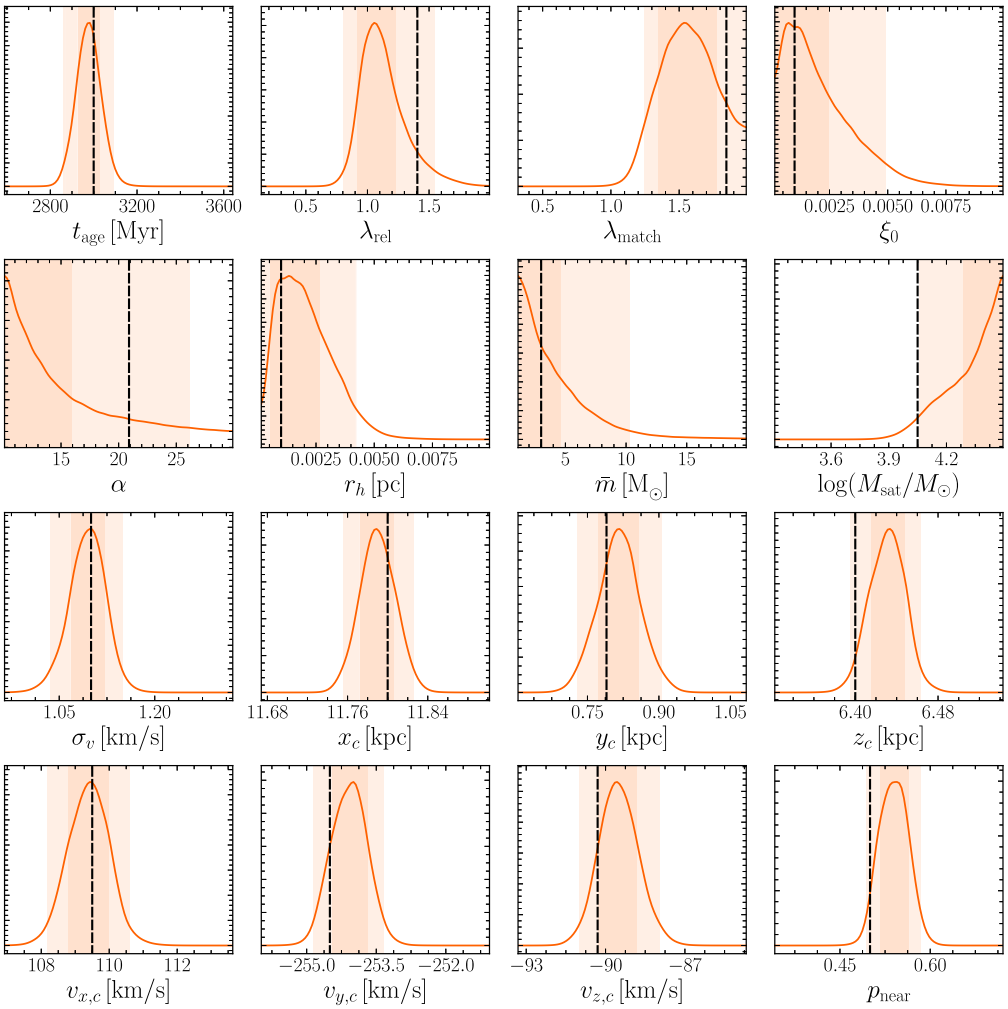

Figure 3. Full set of 1d marginal posteriors (orange curves) for all parameters in the `sstrax` stream model described in Section 2 applied to the mock observation in Fig. 2. The 1σ, 2σ, and 3σ contours are overlaid behind. Finally, the black vertical lines indicate the true injected parameters from Table 1.

inference on streams like GD1 in as realistic as possible simulation framework.

To illustrate and test our method, we construct a case study to closely resemble the sky location and structure of the GD1 stream. To do so, we choose a stream closely aligned with the $\phi_2 = 0\,\text{deg}$ plane in the GD1 specific co-ordinate system defined in Section 2. We centre the location of the cluster (remnant) at $\phi_1 \sim -25\,\text{deg}$, and choose the age of the stream such that it extends across a significant portion of the sky, as in the case of the real GD1 observation (Grillmair & Dionatos 2006; Eyre 2010; Price-Whelan & Bonaca 2018). Similarly, the dominant part of the stream is located around $6 - 10\,\text{kpc}$ away from the galactic centre. The full set of parameter values that we choose for the mock observation are shown in Table 1, along with the priors for the subsequent Bayesian inference. The mock observation that these parameters generate, and the focus of the analysis below is shown in Fig. 2. We do note, however, that whilst the mock observation we present here has general features that represent a GD1-like stream, it is important to acknowledge that some aspects of the modelling could be improved in this regard. One important example is the presence of a clear remnant at the cluster centre, which is not present in the real GD1 data. Concretely, this has the effect that the reconstruction of the cluster position in our analysis may be overly optimistic compared to a more realistic case.

4.2 Parameter estimation with TMNRE

We carry out parameter estimation using the priors for the model parameters indicated in Table 1, the observational model described in Table 2, and the TMNRE algorithm settings given in Table 3. The key results for this section are given in Figs 3, 4, and 5.

4.2.1 Overview

There are a few levels at which to discuss our results from applying the TMNRE algorithm described in Section 3 to the mock GD1-like observation in Fig. 2. The first is simply in the context of robust and faithful inference – in Fig. 3, we show the full set of converged 1d-posteriors for all parameters in the model. We see that we reconstruct the true value in all cases either via a direct measurement (e.g. the final position or velocity of the cluster) or as some clear upper or lower bound (e.g. the mass-loss parameter α). Importantly, we can reconstruct with very high precision the age of the stream ($t_{\rm age}$), the velocity dispersion of the cluster (σ_v), the current cluster position and velocity ($[x_c, y_c, z_c]$, $[v_{x,c}, v_{y,c}, v_{z,c}]$), and the relative asymmetry in the tidal stripping ($p_{\rm near}$). This sort of constraint is easy to motivate physically via e.g. the length and width of the stream, which is strongly affected by the age and velocity dispersion, as well as the

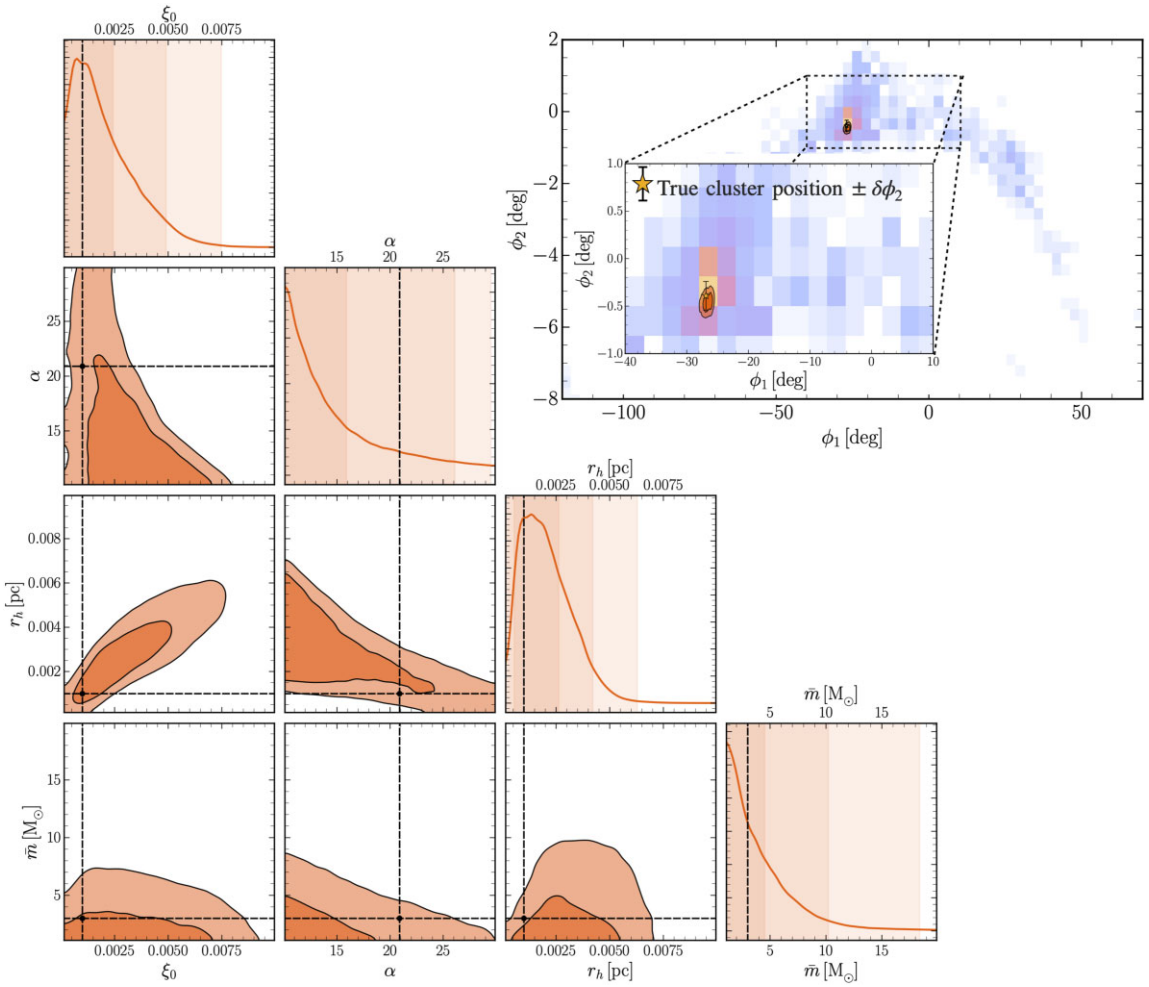

Figure 4. Corner plot: Follow up analysis on the mass-loss model given the 1d marginals in Fig. 3. The orange contours show the trained 2d posteriors on the parameters relevant to the mass-loss model (ξ_0, α, r_h, $\bar{m}$). Upper right-hand panel and inset: Derived marginal posteriors (orange contours) in the (ϕ_1, ϕ_2)-plane on the final cluster position overlaid on top of the mock target observation. In addition, we highlight the observational errors (black error bars) and the true value (yellow star) to be reconstructed.

stream's spatial location and orientation which is controlled by the relative cluster position and velocity.[19]

4.2.2 *Degeneracies*

Of course, not every parameter is measured with high precision, such as the parameters (ξ_0, α, r_h, $\bar{m}$) that control the mass-loss rate of the cluster as it orbits the Milky Way. Whilst we can set relevant upper or lower bounds on these parameters,[20] it is interesting to explore the degeneracies between these parameters also. This is where we can use the flexibility of the TMNRE algorithm to efficiently estimate posteriors of choice. Specifically, we only need to estimate the relevant 2d posteriors for exploring the degeneracy structure of the mass-loss model. To do so, we train additional ratio estimators $r(x; \{\theta_i, \theta_j\})$ with $\theta_i, \theta_j \in (\xi_0, \alpha, r_h, \bar{m})$. The results for these 2d-posteriors, along with the corresponding 1d-posteriors are shown in Fig. 4. We see some clear degeneracies highlighted in the parameter inference such as those between ξ_0 and r_h. Again, whilst we do not investigate these degeneracies in detail, this could be expected from e.g. the scaling of $\mathrm{d}M_c/\mathrm{d}t$ in equation (2).

4.2.3 *Precision*

As discussed in Section 3, SBI has now been used extensively in other fields, and has been shown to qualitatively and quantitatively reproduce known results and results obtained using traditional methods. In the present case, a full comparison to e.g. a traditional method such as MCMC is challenging because we only have a forward simulator for the observational and noise models. This is another way to say that we do not have an explicit form of the data likelihood. Of course, this is a key strength of the class of simulation-based methods, since it allows for arbitrarily complex data simulators, which can account for complicated aspects of detection and selection in a statistically

[19]Indirectly, the exact orientation will be affected, of course, by the gravitational potential of the Milky Way, see e.g. (Koposov et al. 2010; Bonaca et al. 2014; Gibbons et al. 2014; Sanderson et al. 2014; Bowden et al. 2015; Bovy et al. 2016; Erkal et al. 2019; Shipp et al. 2021), which we have fixed in this analysis. It is easy to generalise to a case where the potential is allowed to freely vary with an analysis pipeline that would remain identical.

[20]Which again we might be able to motivate physically – for example, it makes sense that we can set a lower bound on the total mass of the cluster just by having some count of the total number of observed stream stars and multiplying by the average stellar mass.

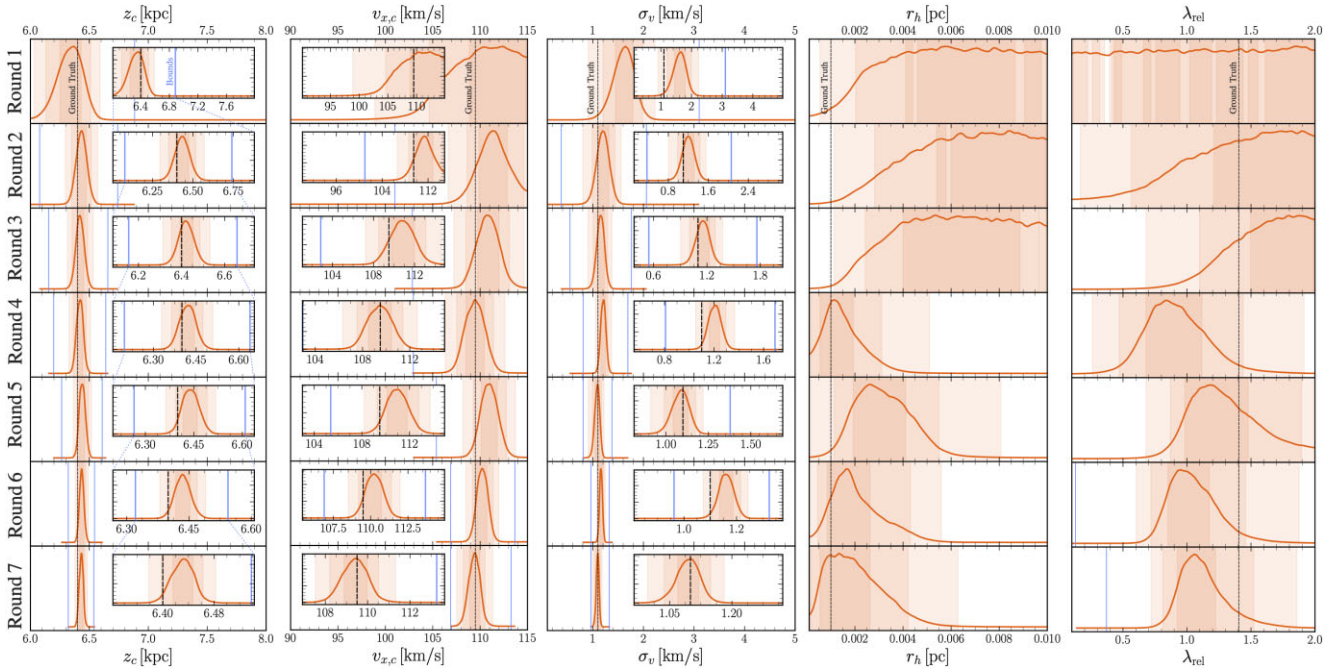

Figure 5. Examples of the truncation procedure in TMNRE applied to five of the model parameters. From left to right, we illustrate the evolution of the posterior estimates for z_c, $v_{x,c}$, σ_v, r_h, and $\lambda_{\rm rel}$. From top to bottom we show the development over the number of rounds of the TMNRE algorithm. The *insets* zoom in on the bounded region (blue vertical lines) to highlight the coverage of the true value (vertical black dotted line).

meaningful way. On the other hand, this means we should consider additional ways to test our results. A simple qualitative test we can perform is to compare the precision (and accuracy) with which we reconstruct the cluster position to the intrinsic observational errors on the stellar positions. To test this, we construct the 3d-joint posterior $p(x_c, y_c, z_c|x_0)$ by training a 3d ratio estimator $r(x; \{x_c, y_c, z_c\})$ on the final round of simulations. From this joint posterior, we can generate posterior samples in the (ϕ_1, ϕ_2) parameter space[21] for the current position of the cluster that are distributed as $\phi_1, \phi_2 \sim p(\phi_1, \phi_2|x_0)$. These are shown in the top right-hand panel (and inset) of Fig. 4 along with the observational model errors on the positions of the stars $\delta\phi_1$, $\delta\phi_2$. We see that we are able to reconstruct the cluster position to a good degree of accuracy and precision.

4.2.4 *Simulation efficiency*

One of the key arguments we made for using TMNRE was the fact that it gave us the ability to use high fidelity simulators. This is both from a statistical perspective in the sense that we can perform Bayesian inference without explicit likelihoods, but also from the scalability point of view. Indeed, one of the main obstacles for a full analysis of stellar streams is that fact that performing enough simulations to do inference on a large number of parameters is typically infeasible. This is where the marginal and targeted aspects of TMNRE are relevant, as well as the acceleration of the simulator. To be more specific, in the case study described above, we required a total of only 350k simulations to perform inference on all 16 parameters simultaneously. Crucially, this simulation budget was split across a total of seven rounds, as illustrated in Table 3. In between the rounds, the truncation procedure described in Section 3 was applied, which ensures that we are targeting the specific observation of interest, and that the variance in the training data is significantly reduced compared to the previous round. This is very important for simulation efficiency, and results in much higher quality inference results on targeted observations compared to e.g. the case where a fixed simulation budget is used in a single round.[22] This truncation process is highlighted in Fig. 5, where we see how the different classes of parameter respond to the truncation process. For example, the first three columns of parameters (one component of the position and velocity, and the velocity dispersion of the stream) are extremely well constrained once the algorithm converges. On the other hand, the last two panels show parameters that are only broadly reconstructed (r_h and $\lambda_{\rm rel}$). For this second class of parameter, however, we see that in the initial rounds, the marginal posterior estimates of e.g. $p(r_h|x_0)$ are quite poor.[23] As the rounds evolve and the well-measured parameters are better constrained, subsequently reducing the training data variance, the posterior estimates on the poorly reconstructed parameters significantly improve. This is a general feature of TMNRE, where convergence and truncation in one set of parameters leads to marked improvements in the inference of other model parameters, even if they themselves are not well measured.

In terms of actual run time, we performed this analysis on a 72 CPU core cluster node, with a single `NVIDIA A100` GPU to train the ratio estimators. The total run time for the analysis was around 19 h, of which approximately 90 per cent was simulation time. Note

[21] Note that, of course, it would have been statistically incorrect to generate these from the individual marginal posteriors on the cluster positions, even though they are well measured.

[22] Of course, if the goal is to perform some sort of amortized inference across all possible observations, then one should use this hypothetical simulation budget differently. For an example in the context of gravitational waves, see e.g. (Bhardwaj et al. 2023).

[23] In fact, it is a good example of where we should be careful not to interpret these early-round ratio estimators as strict posteriors, since the algorithm has not converged.

that this can therefore be improved immediately by either (i) further speeding up the simulator, or (ii) having access to more CPU cores where the simulations can be further parallelized.

4.3 Consistency and validation tests

The posterior sanity checks and explicit evidence for excellent reconstruction of the true values for the parameters in our case study are an important step towards developing and testing our analysis pipeline. On the other hand, given that our goal is to target data analysis challenges where there are no traditional methods available – either because they scale too poorly with the number of parameters, or because they have an analytically intractable data likelihood – we need to develop additional consistency checks to validate our results. This is very much an active field of research in SBI, and a set of established methods now exist (Hermans et al. 2021a; Lueckmann et al. 2021).

The most common, and the one that we will present here, are known as coverage tests (Hermans et al. 2021a). We will focus on expected coverage tests of our inference pipeline, which make precise the idea of variations in posterior estimates over various observational or statistical fluctuations. In particular, expected coverage tests ask the following question: *how often does the x per cent credible interval contain the true value, averaged over observations generated from the joint distribution x, $\theta \sim p(x, \theta)$?* By definition, a well-calibrated posterior distribution will contain the true value inside the x per cent credible internal x per cent of the time. As such, to carry out this test, we can generate a set of simulations[24] from the truncated prior in the final round of inference (so that we test the most relevant region of parameter space), and then perform inference on each simulation using our trained final round ratio estimators. For each confidence level x per cent $\in [0, 1]$, we can then count how many simulations have inference results that contain the corresponding true value within this confidence interval. A posterior will pass this test if this results in an approximately diagonal line in the *expected* versus *empirical* coverage plane. Importantly, since the inference must be done individually for each mock observation, it is typically infeasible to perform this sort of coverage test with fully sequential methods (including e.g. MCMC or nested sampling), especially in scenarios with high simulation cost such as stellar streams. Finally, one should note that this coverage test is diagnostic in the sense that a failure indicates a poorly calibrated posterior estimate, but success does not guarantee that the correct posterior has been found.

We provide the coverage test results for all 16 parameters in the Appendix (see Figs B2 and B3), but also give a specific example for the age of the stream $t_{\rm age}$ in Fig. 6 opposite. We see that in all cases we achieve good coverage results, which can easily be improved further by allocating a slightly larger simulation budget. This coverage test diagnostic will remain applicable irrespective of the forward model or parameter choices, and is one of the key metrics for being able to validate SBI methods.

5 CONCLUSIONS AND OUTLOOK

In this work, we have presented the development and application of a brand new SBI data analysis pipeline for modelling (see Section 2) and analysing stellar streams (see Sections 3 and 4). In this last section, we present our key conclusions and provide some outlook as to the classes of analysis challenge we can now attempt to tackle,

[24]Here we generate 1000 new simulations.

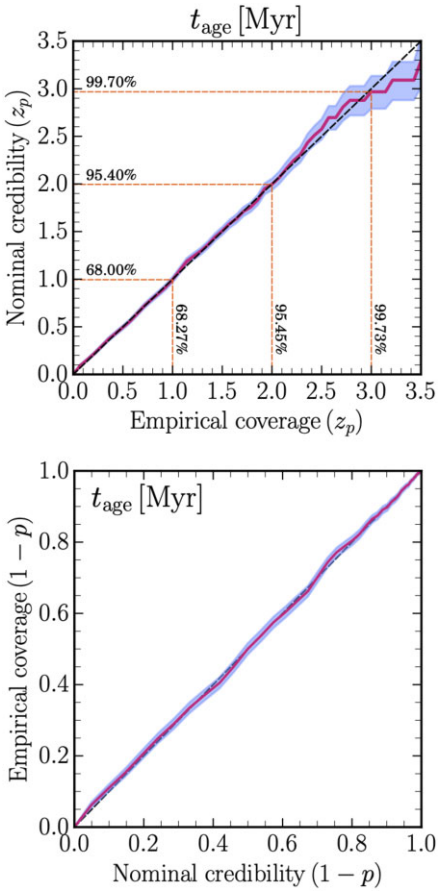

Figure 6. Example of the coverage tests applied in this work for the age of the stream $t_{\rm age}$. Top panel: Empirical (observed) against the nominal or expected coverage. Bottom panel: The same information as the top panel but plotted in terms of the corresponding p values. The red lines indicate the actual coverage results, whilst the blue contours represent the 1σ confidence interval on this estimate.

as well as the steps that would be required to achieve them. The key contributions in the work are as follows:

(i) *Scalable SBI pipeline.* We have developed a brand new SBI algorithm to analyse stellar streams in the Milky Way (see Section 3 for a discussion on the application of simulation-based methods to stellar streams). In particular, we have implemented the TMNRE algorithm (Miller et al. 2022b) with the aim to develop a scalable inference method. The motivation for choosing this algorithm for the analysis of stellar streams is mainly due to simulation efficiency that results from targeting individual observations and focusing on marginals. We showed in Section 4 that we were able to perform inference on all 16 parameters of our model with only 350k simulations. This sort of performance is the key argument for the ability of our approach to analyse streams with far more realistic forward models.

(ii) *Robust and flexible method.* Another important aspect of the analysis methodology developed here is its flexibility and robustness to changes in observation strategy or simulation model. By definition, our approach is simulation-based and therefore has the advantages that it does not (i) assume any explicit likelihood for e.g. the observational model, only the existence of a forward simulator, and (ii) assume any particular form of the data output. On the latter point, whilst we have developed the algorithm alongside our modelling

code `sstrax`, the analysis pipeline would remain *identical* for any simulator. This is the crucial aspect that will allow our method to be used for making direct comparisons between different stream simulation strategies and observational models.

(iii) *Public analysis code.* We have built our analysis method on top of the `swyft` software which is a `pytorch`-based implementation of TMNRE (Miller et al. 2022b). Specifically, we have publicly released the `albatross` code that is currently coupled to the `sstrax` modelling code by default. The `albatross` code is highly modular and can in principle be coupled to any forward model, for example `galpy` (Bovy 2015), without any change in the analysis methodology. This will eventually allow for direct comparisons in the inference between different modelling strategies.

(iv) *Public modelling code.* As mentioned above, one of the key motivations for developing the `albatross` implementation of the TMNRE algorithm was to create a framework that allows for robust, scalable inference on complex models. In the same vein, we developed a new modelling code `sstrax` that is accelerated through the `jax` programming paradigm (Bradbury et al. 2018). This allows for fast (around a second per simulation) and realistic forward modelling of streams. We have designed the code to be readily extendable to include any physical effects such as subhalo impacts, varying gravitational potentials, higher fidelity tidal disruption models etc. As above, regardless of the modelling choices, the inference pipeline will crucially remain identical.

5.1 Outlook

We argued in the introduction that stellar streams are an exciting probe of galactic and dark matter physics. This is particularly true as the quality of observations continues to significantly improve in the eras of *Gaia* and the Vera Rubin observatory (Abell et al. 2009; Gaia Collaboration 2018, 2021; Bechtol et al. 2019). Taking full advantage of this data is challenging, however, both in terms of robust statistical analysis and the complexity of simulations required. Ultimately, if we are interested in using stellar streams to analyse scenarios such as the origin and statistics of substructure in the stream (Banik et al. 2018, 2021a, b; Banik & Bovy 2019), or the impact of a large population of low mass subhaloes on streams in the Milky Way (Erkal & Belokurov 2015a, b; Bonaca et al. 2018), we will have to overcome these hurdles. This is the context we had in mind when developing `albatross` and `sstrax`. The aim was to develop a scalable, simulation-efficient framework that did not make any assumption about the complexity of the forward simulator. This is exactly the sort of task that SBI methods were developed to address. In terms of specific outlook, we believe there are a number of interesting avenues to pursue given the capabilities developed here.

On the analysis side, there are a number of interesting claims in the literature about the origin and characterization of the gaps and features in streams such as GD1 (Grillmair & Dionatos 2006; Eyre 2010; de Boer et al. 2018; Price-Whelan & Bonaca 2018). More specifically, it would be an extremely important result to classify e.g. the gap in GD1 as being due to a compact object or subhalo collision (Carlberg & Grillmair 2013; de Boer et al. 2018; Price-Whelan & Bonaca 2018). To obtain a definitive answer, however, one needs to show that the features cannot (at least to some degree of statistical certainty) arise by chance as a result of some stochasticity in the tidal stripping process, selection effects at the level of stream detection, or as a result of a more complex model of the Milky Way potential including known substructure such as dwarf galaxies or globular clusters (Amorisco et al. 2016; Dillamore et al. 2022; Doke & Hattori 2022). Similarly, it would be interesting to provide a conclusive answer as to the relative shape and size of the Milky Way gravitational potential from an analysis of individual or multiple streams (Shipp et al. 2021). The key advantage of the framework we have put forward here is that one can (and should) ask all of these questions simultaneously. This is a more precise version of the statement in the introduction where we argued that we would ideally like to analyse the large- and small-scale structures present in stellar streams at the same time. On a more cautionary note, in order to move towards analysing real data in its full complexity with this class of SBI methods, it will be important to characterize and quantify the sensitivity of the inference method to perturbations or misspecifications in the forward model. This is particularly relevant in the case of stellar streams where the physics is highly complex, and it is unlikely to be possible for a simulation model to be developed that is simultaneously fully self-consistent *and* fast enough for parameter inference. One step towards this goal could include analysing mock streams generated from N-body simulations (see e.g. Varghese et al. 2011) with identifiable parameters that match those in the model (such as the cluster velocity dispersion, analytic potential, or the age of the stream). One should bear in mind, however, that this is not necessarily an issue directly with SBI, but also affects traditional approaches such as MCMC if e.g. the modelling or data likelihood is miscalibrated.

Of course, to achieve these analysis goals, we must also make progress on modelling. Having a flexible analysis and simulation pipeline that does not assume e.g. symmetry in the Milky Way potential, or uniform stripping times in the evolution of the cluster motivates us to focus on improving the realism of each aspect. In particular, there are a number of key developments that would place the analysis questions above on a much more solid footing and allow us to analyse real data with confidence. First, we should focus attention on the observational model – in this work we constructed a very simple framework to describe the detection and measurement of Milky Way streams. In reality, however, data such as that from *Gaia* is significantly more complicated (Gaia Collaboration 2018, 2021), accounting for e.g. position dependent errors, selection effects based upon proper motions and metallicities, and spatially varying background densities (Gaia Collaboration 2018, 2021). Realistic modelling of this will be particularly relevant for robustly studying e.g. small-scale features in streams. Secondly, the dynamics of tidal disruption and the release of stars from the cluster is vital for generating realistic density perturbations along the stream track. Again, since this could be an interesting observable for studying e.g. the collective implications of a population of small perturbers (Banik et al. 2018; Bonaca et al. 2018; Banik & Bovy 2019; Delos & Schmidt 2022), or the internal properties of globular clusters (Gialluca et al. 2021), development of the model realism will inevitably lead to more informative inference results. Thirdly, we know that on the sort of time-scales relevant to stellar streams, the Milky Way and its potential are dynamical, both in terms of its global structure, as well as the large amount of substructure in the form of dwarf galaxies, other clusters, or gas clouds (Amorisco et al. 2016; Dillamore et al. 2022; Doke & Hattori 2022). It would be interesting to take input from e.g. N-body simulations of Milky Way formation and trace the evolution of streams in such a dynamical potential. As we argued above, this could be done without any change in the analysis pipeline.

In summary, the development of a scalable and flexible SBI approach to analysing stellar streams can allow us to answer important questions about the evolution of, and substructure in our own galaxy. Aided by high quality observations by the latest surveys (Abell et al. 2009; Gaia Collaboration 2018, 2021; Bechtol et al. 2019), we can use this to start asking concrete questions regarding the nature of dark matter, the evolution, and structure of the Milky Way, or the dynamics

of dwarf galaxies and globular clusters. To achieve this will require development from the perspective of modelling stream dynamics and survey observations. However, having a robust simulation efficient inference strategy is strong motivation for starting to move further towards this ambitious goal.

ACKNOWLEDGEMENTS

JA is supported through the research program 'The Hidden Universe of Weakly Interacting Particles' with project number 680.92.18.03 (NWO Vrije Programma), which is partly financed by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (Dutch Research Council). CW and MG are supported by a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 864035). We are extremely grateful to Noemi Anau Montel, Gianfranco Bertone, and Sam Witte for helpful discussions regarding this project. The main analysis for this work was carried out on the Lisa and Snellius Computing Clusters at SURFsara.

DATA AVAILABILITY

The code for generating the data underlying this manuscript is available in the `sstrax` library (at this link) and the `swyft`-based inference library `albatross` (available here). **Github**: Our new `jax`-accelerated stellar streams modelling code `sstrax` can be found here. The `swyft`-based inference library `albatross` is available at this link.

REFERENCES

Abazajian K. N. et al., 2009, ApJS, 182, 543
Abell P. A. et al., 2009, preprint (arXiv:0912.0201)
Alsing J., Charnock T., Feeney S., Wandelt B., 2019, MNRAS, 488, 4440
Amorisco N. C., Gòmez F. A., Vegetti S., White S. D. M., 2016, MNRAS, 463, L17
Anau Montel N., Weniger C., 2022, preprint (arXiv:2211.04291)
Ashton G. et al., 2022, Nature, 2, 39
Balbinot E., Gieles M., 2018, MNRAS, 474, 2479
Banik N., Bovy J., 2019, MNRAS, 484, 2009
Banik N., Bovy J., 2021, MNRAS, 504, 648
Banik N., Bertone G., Bovy J., Bozorgnia N., 2018, J. Cosmol. Astropart. Phys., 2018, 061
Banik N., Bovy J., Bertone G., Erkal D., de Boer T. J. L., 2021a, J. Cosmol. Astropart. Phys., 2021, 043
Banik N., Bovy J., Bertone G., Erkal D., de Boer T. J. L., 2021b, MNRAS, 502, 2364
Baumgardt H., 1998, A&A, 330, 480
Baumgardt H., Makino J., 2003, MNRAS, 340, 227
Bechtol K. et al., 2019, preprint (arXiv:1903.04425)
Belokurov V. et al., 2006, ApJ, 642, L137
Bhardwaj U., Alvey J., Miller B. K., Nissanke S., Weniger C., 2023, preprint (arXiv:2304.02035)
Bonaca A., Geha M., Küpper A. H. W., Diemand J., Johnston K. V., Hogg D. W., 2014, ApJ, 795, 94
Bonaca A., Hogg D. W., Price-Whelan A. M., Conroy C., 2019, ApJ, 880, 38
Bonaca A. et al., 2020, ApJ, 892, L37
Borsato N. W., Martell S. L., Simpson J. D., 2020, MNRAS, 492, 1370
Boubert D., Everall A., 2020, MNRAS, 497, 4246
Bovy J., 2014, ApJ, 795, 95
Bovy J., 2015, ApJS, 216, 29
Bovy J., Bahmanyar A., Fritz T. K., Kallivayalil N., 2016, ApJ, 833, 31
Bovy J., Erkal D., Sanders J. L., 2017, MNRAS, 466, 628
Bowden A., Belokurov V., Evans N. W., 2015, MNRAS, 449, 1391
Bradbury J. et al., 2018, google/jax, available at: http://github.com/google/jax
Brehmer J., Cranmer K., 2020, preprint (arXiv:2010.06439)
Buist H. J. T., Helmi A., 2015, A&A, 584, A120
Carlberg R. G., Grillmair C. J., 2013, ApJ, 768, 171
Cole A., Miller B. K., Witte S. J., Cai M. X., Grootes M. W., Nattino F., Weniger C., 2022, J. Cosmol. Astropart. Phys., 2022, 004
Craig P., Chakrabarti S., Sanderson R. E., Nikakhtar F., 2023, ApJ, 945, L32
Cranmer K., Brehmer J., Louppe G., 2020, Proc. Nat. Acad. Sci., 117, 30055
Dax M., Green S. R., Gair J., Macke J. H., Buonanno A., Schölkopf B., 2021, Phys. Rev. Lett., 127, 241103
de Boer T. J. L., Belokurov V., Koposov S. E., Ferrarese L., Erkal D., Côté P., Navarro J. F., 2018, MNRAS, 477, 1893
Delaunoy A., Hermans J., Rozet F., Wehenkel A., Louppe G., 2022, preprint (arXiv:2208.13624)
Delos M. S., 2019, Phys. Rev. D, 100, 063505
Delos M. S., Schmidt F., 2022, MNRAS, 513, 3682
Dillamore A. M., Belokurov V., Evans N. W., Price-Whelan A. M., 2022, MNRAS, 516, 1685
Doke Y., Hattori K., 2022, ApJ, 941, 129
Drakos N. E., Taylor J. E., Benson A. J., 2020, MNRAS, 494, 378
Drakos N. E., Taylor J. E., Benson A. J., 2022, MNRAS, 516, 106
Durkan C., Murray I., Papamakarios G., 2020, preprint (arXiv:2002.03712)
Erkal D., Belokurov V., 2015a, MNRAS, 450, 1136
Erkal D., Belokurov V., 2015b, MNRAS, 454, 3542
Erkal D. et al., 2019, MNRAS, 487, 2685
Eyre A., 2010, MNRAS, 403, 1999
Fardal M. A., Huang S., Weinberg M. D., 2015, MNRAS, 452, 301
Foreman-Mackey D., Hogg D. W., Lang D., Goodman J., 2013, PASP, 125, 306
Gagnon-Hartman S., Ruan J., Haggard D., 2023, MNRAS, 520, 1
Gaia Collaboration, 2018, A&A, 616, A1
Gaia Collaboration, 2021, A&A, 649, A1
Gialluca M. T., Naidu R. P., Bonaca A., 2021, ApJ, 911, L32
Gibbons S. L. J., Belokurov V., Evans N. W., 2014, MNRAS, 445, 3788
Grillmair C. J., Dionatos O., 2006, ApJ, 643, L17
Hammer F. et al., 2023, MNRAS, 519, 5059
Handley W. J., Hobson M. P., Lasenby A. N., 2015, MNRAS, 453, 4384
Helmi A., 2020, Ann. Rev. Astron. Astrophys., 58, 205
Hermans J., Begy V., Louppe G., 2019, preprint (arXiv:1903.04057)
Hermans J., Delaunoy A., Rozet F., Wehenkel A., Begy V., Louppe G., 2021a, preprint (arXiv:2110.06581)
Hermans J., Banik N., Weniger C., Bertone G., Louppe G., 2021b, MNRAS, 507, 1999
Huang Y., Chen B. Q., Zhang H. W., Yuan H. B., Xiang M. S., Wang C., Tian Z. J., Liu X. W., 2019, ApJ, 877, 13
Ibata R. A., Lewis G. F., Irwin M. J., 2002, MNRAS, 332, 915
Ibata R., Thomas G., Famaey B., Malhan K., Martin N., Monari G., 2020, ApJ, 891, 161
Ibata R. et al., 2021a, ApJ, 914, 123
Ibata R., Diakogiannis F. I., Famaey B., Monari G., 2021b, ApJ, 915, 5
Johnston K. V., Spergel D. N., Haydn C., 2002, ApJ, 570, 656
Karchev K., Trotta R., Weniger C., 2023, MNRAS, 520, 1056
Kidger P., 2021, PhD thesis, University of Oxford
Koposov S. E., Rix H.-W., Hogg D. W., 2010, ApJ, 712, 260
Koposov S. E. et al., 2023, MNRAS, 521, 4936
Kuepper A. H. W., Kroupa P., Baumgardt H., Heggie D. C., 2010, MNRAS, 401, 105
Kuepper A. H. W., Lane R. R., Heggie D. C., 2012, MNRAS, 420, 2700
Loyola G. R. I. M., Hurley J. R., 2013, MNRAS, 434, 2509
Lueckmann J.-M., Boelts J., Greenberg D. S., Gonçalves P. J., Macke J. H., 2021, preprint (arXiv:2101.04653)
Mackay D., 2003, Informaiton Theory, Inference, and Learning Algorithms. Cambridge Univ. Press, Cambridge
Mackey D. et al., 2019, Nature, 574, 69
Madrid J. P., Leigh N. W. C., Hurley J. R., Giersz M., 2017, MNRAS, 470, 1729
Malhan K., Ibata R. A., 2018, MNRAS, 477, 4063
Malhan K., Ibata R. A., Martin N. F., 2018, MNRAS, 481, 3442

Malhan K., Ibata R. A., Carlberg R. G., Valluri M., Freese K., 2019, ApJ, 881, 106
Malhan K., Valluri M., Freese K., 2021, MNRAS, 501, 179
Malhan K., Valluri M., Freese K., Ibata R. A., 2022, ApJ, 941, L38
Martin N. F., Ibata R. A., Starkenburg E., Yuan Z., Malhan K. et al., 2022, MNRAS, 516, 5331
Miller B. K., Cole A., Forré P., Louppe G., Weniger C., 2021, preprint (arXiv:2107.01214)
Miller B. K., Weniger C., Forré P., 2022a, preprint (arXiv:2210.06170)
Miller B. K., Cole A., Weniger C., Nattino F., Ku O., Grootes M. W., 2022b, J. Open Source Softw., 7, 4205
Montel N. A., Coogan A., Correa C., Karchev K., Weniger C., 2022, MNRAS, 518, 2746
Nibauer J., Belokurov V., Cranmer M., Goodman J., Ho S., 2022, ApJ, 940, 22
Nibauer J., Bonaca A., Johnston K. V., 2023, preprint (arXiv:2303.17406)
Panithanpaisal N., Sanderson R. E., Arora A., Cunningham E. C., Baptista J., 2022, preprint (arXiv:2210.14983)
Papamakarios G., Murray I., 2016, preprint (arXiv:1605.06376)
Papamakarios G., Sterratt D., Murray I., 2019, in Chaudhuri K., Sugiyama M., eds, Proc. Machine Learning Research Vol. 89, Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics. PMLR, p. 837
Pavanel N., Webb J. J., 2021, MNRAS, 503, 1932
Penarrubia J., Benson A. J., Martinez-Delgado D., Rix H.-W., 2006, ApJ, 645, 240
Price-Whelan A. M., Bonaca A., 2018, ApJ, 863, L20
Qian Y., Arshad Y., Bovy J., 2022, MNRAS, 511, 2339
Ronneberger O., Fischer P., Brox T., 2015, preprint (arXiv:1505.04597)
Rossi L. J., Bekki K., Hurley J. R., 2016, MNRAS, 462, 2861
Rozet F., Louppe G., 2021, preprint (arXiv:2110.00449)
Sanderson R. E., Helmi A., Hogg D. W., 2014, in Feltzing S., Zhao G., Walton N. A., Whitelock P. A., eds, Proc. IAU Symp. 298, Setting the Scene for Gaia and LAMOST. Cambridge Univ. Press, Cambridge, p. 207
Schulz C., Pflamm-Altenburg J., Kroupa P., 2015, A&A, 582, A93
Shih D., Buckley M. R., Necib L., Tamanas J., 2021, MNRAS, 509, 5992
Shih D., Buckley M. R., Necib L., 2023, preprint (arXiv:2303.01529)
Shipp N. et al., 2021, ApJ, 923, 149
Skilling J., 2006, Bayesian Anal., 1, 833
Spurio Mancini A., Docherty M. M., Price M. A., McEwen J. D., 2022, preprint (arXiv:2207.04037)
Stücker J., Ogiya G., Angulo R. E., Aguirre-Santaella A., Sánchez-Conde M. A., 2023, MNRAS, 521, 4432
Takahashi K., Portegies Zwart S. F., 2000, ApJ, 535, 759
Taylor J. E., Babul A., 2001, ApJ, 559, 716
Tejero-Cantero A., Boelts J., Deistler M., Lueckmann J.-M., Durkan C., Gonçalves P. J., Greenberg D. S., Macke J. H., 2020, J. Open Source Softw., 5, 2505
van den Bosch F. C., Ogiya G., Hahn O., Burkert A., 2018, MNRAS, 474, 3043
Varghese A., Ibata R., Lewis G. F., 2011, MNRAS, 417, 198
Vitral E., Kremer K., Libralato M., Mamon G. A., Bellini A., 2022, MNRAS, 514, 806
Zeghal J., Lanusse F., Boucaud A., Remy B., Aubourg E., 2022, preprint (arXiv:2207.05636)

APPENDIX A: CO-ORDINATE TRANSFORMATIONS IN `SSTRAX`

Here we detail the co-ordinate transformations we use in `sstrax` to move from the Cartesian simulation frame $X_{\rm halo} \equiv (x, y, z)$ to the GD1 co-ordinates (r, ϕ_1, ϕ_2). This is implemented in the `projection.py` module, and is explicitly given by the following set of relations,

$$X_{\rm halo} \equiv (x, y, z), \tag{A1}$$

Then, in a frame centred at the sun with $x_{\rm sun} = 8\,{\rm kpc}$,

$$X_{\rm sun} \equiv (\tilde{x}, \tilde{y}, \tilde{z}) = (x_{\rm sun} - x, y, z). \tag{A2}$$

We can convert to galactic co-ordinates $X_{\rm gal} \equiv (r, b, l)$ via,

$$r = \sqrt{\tilde{x}^2 + \tilde{y}^2 + \tilde{z}^2}, \tag{A3}$$

$$b = \arcsin(\tilde{y}/r), \tag{A4}$$

$$l = \arctan(\tilde{y}/\tilde{x}). \tag{A5}$$

Then, we can rotate to equatorial co-ordinates $X_{\rm equat} \equiv (r, \alpha, \delta)$ through,

$$\alpha = \tan^{-1}\left(\frac{\cos b \sin(l_{\rm NGP} - l)}{\cos\delta_{\rm NGP}\sin b - \sin\delta_{\rm NGP}\cos b\cos(l_{\rm NGP} - l)}\right) + \alpha_{\rm NGP} \tag{A6}$$

$$\delta = \arcsin\left(\sin\delta_{\rm NGP}\sin b + \cos\delta_{\rm NGP}\cos b\cos(l_{\rm NGP} - l)\right), \tag{A7}$$

with $\delta_{\rm NGP} = 27.12825118085622\,{\rm deg}$, $l_{\rm NGP} = 122.9319185680026\,{\rm deg}$, and $\alpha_{\rm NGP} = 192.85948\,{\rm deg}$. After this, we can rotate to a Cartesian co-ordinate frame aligned with the stream $X_{\rm gd1,\,cart} \equiv (x_g, y_g, z_g)$ with,

$$\begin{pmatrix} x_g \\ y_g \\ z_g \end{pmatrix} = \begin{bmatrix} -0.4776303088 & -0.1738432154 & 0.8611897727 \\ 0.510844589 & -0.8524449229 & 0.111245042 \\ 0.7147776536 & 0.4930681392 & 0.4959603976 \end{bmatrix} \times \begin{pmatrix} r\cos\alpha\cos\delta \\ r\cos\alpha\sin\delta \\ r\cos\delta \end{pmatrix}, \tag{A8}$$

taken from Ref. (Koposov et al. 2010). Finally, we can construct our GD1 co-ordinates $X_{\rm gd1} \equiv (r, \phi_1, \phi_2)$ via,

$$\phi_1 = \arctan(y_g/x_g), \quad \phi_2 = \arcsin(z_g/r). \tag{A9}$$

The final step in the co-ordinate transformations is to construct the velocity in a different co-ordinate frame given the velocity in the simulation frame. For this, we take advantage of the autodifferentiation capability of `jax` and numerically compute the Jacobian $\mathcal{J}_{\rm ij} \equiv \partial X^i_{\rm gd1}/\partial X^j_{\rm halo}$. The velocity in the GD1 co-ordinate frame $V_{\rm gd1} \equiv (v_{\rm rad}, \dot{\phi}_1, \dot{\phi}_2)$ is given by $V_{\rm gd1} = \mathcal{J} \cdot V_{\rm halo}$. The proper motions μ_{ϕ_j} are then given by $\mu_{\phi_j} = \dot{\phi}_j/r$.

APPENDIX B: NETWORK ARCHITECTURE

In Fig. B1, we show the network architecture used in the `albatross` code to process output from the simulator and estimate the relevant likelihood-to-evidence ratios.

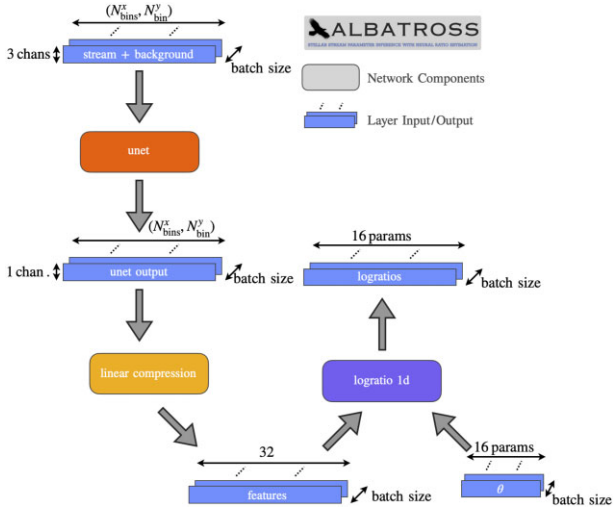

Figure B1. Schematic network diagram illustrating the data processing and ratio estimation network architecture employed in `albatross`.

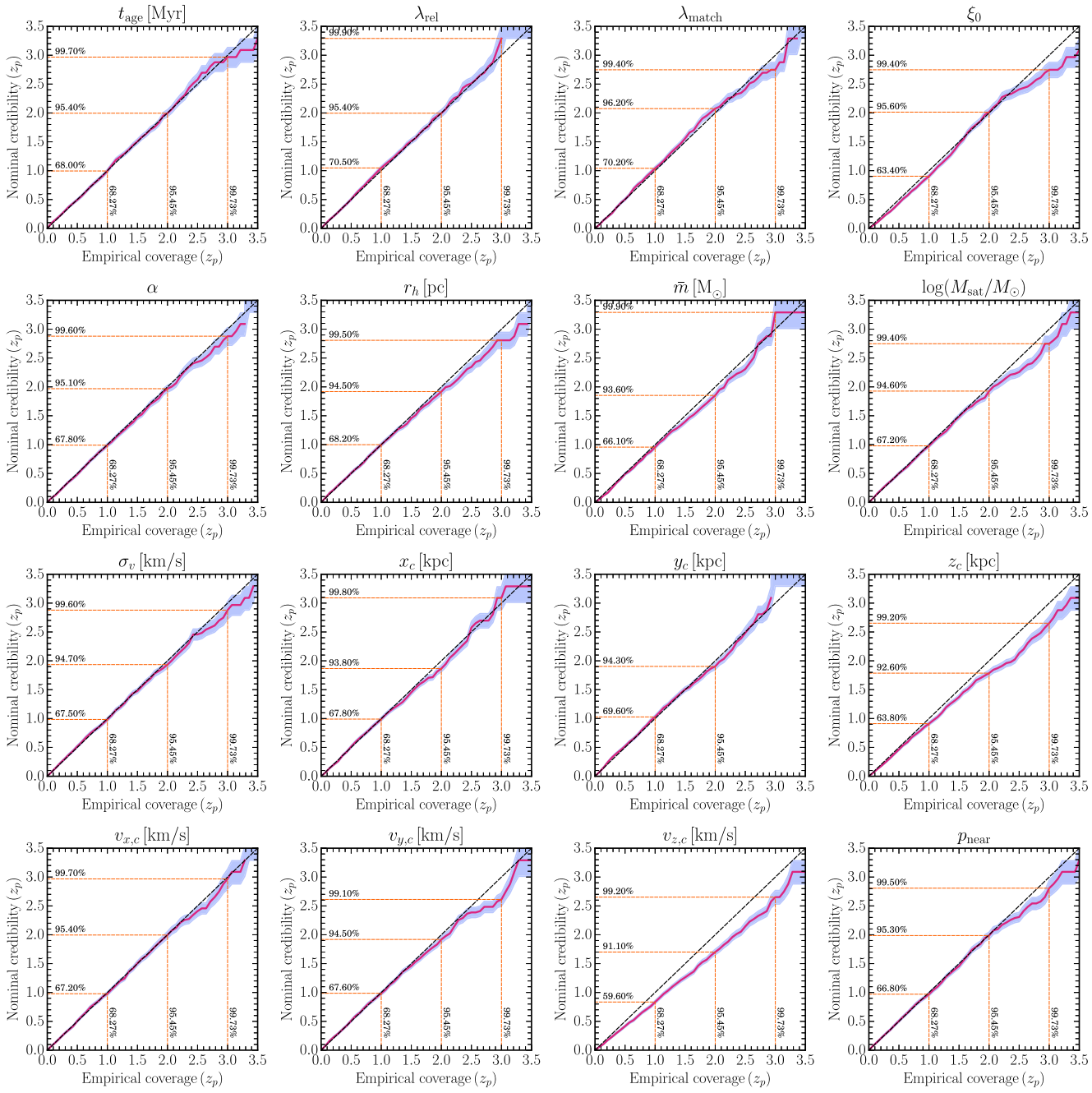

Figure B2. Coverage results for the case study given in Section 4 for all parameters in the `sstrax` stream model. This is the same information as Fig. B3, but with more emphasis placed on the tail regions via the definition $p = \int_{-z_p}^{z_p} \mathrm{d}z\,(1/\sqrt{2\pi})\exp(-z^2/2)$. The pink curves indicate the average coverage, whilst the blue contours represent the 1σ uncertainty of this estimate.

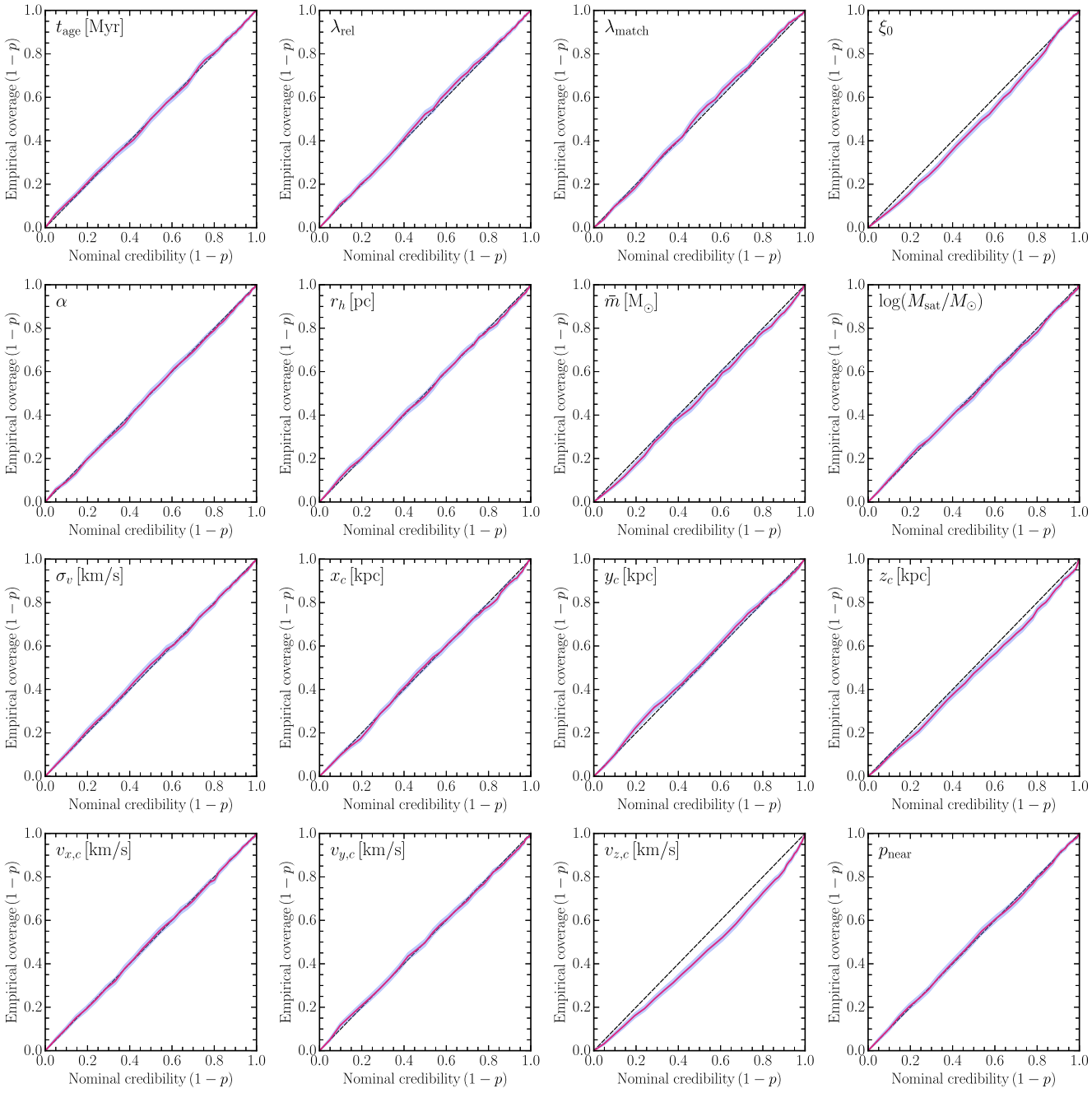

Figure B3. Coverage results for the case study given in Section 4 for all parameters in the `sstrax` stream model. The pink curves indicate the average coverage, whilst the blue contours represent the 1σ uncertainty of this estimate.

This paper has been typeset from a TEX/LATEX file prepared by the author.