

# CASBI: Chemical Abundance Simulation Based Inference

Giuseppe Viterbo

June 11, 2024

# Contents

<b>1</b>	<b>Abstract</b>	<b>2</b>
<b>2</b>	<b>Previous work</b>	<b>3</b>
2.1	The reconstruction of the Assembly history of the Milky Way . . . . .	3
<b>3</b>	<b>CASBI</b>	<b>5</b>
3.1	SBI . . . . .	5
3.2	NIHAO . . . . .	6
3.3	Two step Inference . . . . .	7
3.4	Surrogate Simulator . . . . .	8
<b>4</b>	<b>Analysis</b>	<b>9</b>
4.1	NIHAO UHD . . . . .	9
<b>5</b>	<b>Conclusion</b>	<b>10</b>
5.1	Future work . . . . .	10
5.1.1	semianalicit model . . . . .	10
5.1.2	Number halos has a free parameter: hierarchical sbi . . . . .	10
5.1.3	True test: GAIA . . . . .	10

# Chapter 1

## Abstract

Galaxies evolve through merging events and destroy lower-mass systems over their lifetimes. The contribution that those lower-mass systems bring to the modern picture has been frozen in stellar halos by the long orbital timescales, making the relicts of these objects retain part of their initial progenitor orbit. But dynamical information is not enough to disentangle these components, and the complementary chemical information helps to characterize these building blocks. In fact, merging events tend to quench the star formation rate of these objects, making the chemical abundance plane (iron abundance against  $\alpha$  element abundance) a distinct imprint that retains information on the conditions of formation of their stars, like the total mass and the age of the system until the merging event. This theoretical background allows us to attempt to decompose the stellar halo into its components, unraveling the merging history. In the modern era of large N-body galaxy simulations, we recast this problem into an SBI pipeline to recover the properties of this building block, (e.g. total stellar mass, infall time, ...) using the chemical abundance plane as observables. We therefore present CASBI (Chemical Abundance Simulation Based Inference), a python package to recover the posterior probability of properties of building blocks of Milky Way like galaxy's halo. Moreover, CASBI incorporates conditional neural network architectures as a generator to obtain observables from parameters, smoothly interpolating on regions of the parameters space that weren't fully covered during the N-body simulations.

## Chapter 2

# Previous work

### 2.1 The reconstruction of the Assembly history of the Milky Way

Inferring the assembly history of the Milky Way is a challenging task, even in the era of the astrometric Gaia mission and its 6 dimensional phase space data, and the complementary chemical information obtained from the wide-field spectroscopic programs such as the GALAH survey (De Silva et al. 2015), the H3 survey (Conroy et al. 2019b), APOGEE (Majewski et al. 2017), RAVE (Steinmetz et al. 2006), SEGUE (Yanny et al. 2009), and LAMOST (Cui et al. 2012). The dynamical times of the accreted objects are far longer than the age of the host galaxy, allowing the phase space to retain part of the information on the original orbit parameters. On the other hand, the chemical space is dependent on the star formation history, in particular type II SNe produce  $\alpha$ -elements and iron with a almost constant ratio, while type Ia SNe produce more efficiently iron. Another factor that governs the chemical space is the total mass of the galaxy, since the more massive galaxies are more capable to resist the expulsion of metals due to feedback mechanism. The crossmatch between Gaia and spectroscopic data allowed for the discovery of the "Gaia-Sausage-Enceladus" (GSE) Belokurov et al. 2018; Helmi et al. 2018, a massive accretion event whose remnant now dominates the observation of the inner stellar halo of our Galaxy. Robustly identify distinct structure is challenging, and disentangle the components in fully phase mix situations is nearly impossible. In order to characterize the assembly history [1] propose to use the "CARDS", the chemical abundance ratio distributions of the stars, obtained from the FIRE-2 zoom-in cosmological simulations of MW-mass galaxies (Wetzel et al. 2016). Although similar to CASBI in how to leverage N-body simulations, this method do not recovers posteriors for the parameters of the accreted objects, but rather considers the host halo as a linear combinations of templates CARDS, treating each coefficient as the fraction of mass contribution from the accretion event, and tries to recover those coefficients by maximize a loss that compares the observed CARDS with the combination of the templates. This method and CASBI share two more similarities: 1. Both of these methods are meant to be used on simulations, and the integrations of observational data is not yet implemented, even though theoretically possible. 2. Both rely on the assumption that the chemical space of accreted and isolated dwarf galaxies is very similar, due to ram pressure quenching the star formation history of the accreted object and hence 'freezing' these abundance ratios at the infall time. This assumption is investigated further in section 6 of [1]. Another approach is presented in [2], which takes advantage of the mass-metallicity relation to decompose the metallicity distribution functions (MDF) of the host galaxy as a mixture

of accreted halo's MDF, assumed gaussian for each of these building blocks. The objective of this work is to obtain a posterior distribution for the number of galaxies in each luminosity bin, which can be considered a proxy for the star mass. In CASBI we adopt the same superimposition philosophy of the components contribution, but we do not assume neither a prefix or an analytical form for the joint distribution of the chemical abundance, relaxing these assumption and relying only on the available samples from the N-body simulations.

# Chapter 3

## CASBI

### 3.1 SBI

CASBI is a Simulation Based Inference (SBI) package to recover the properties of building blocks of Milky Way like galaxy's halo from observations of the chemical abundance plane. The SBI framework has existed along side the more traditional likelihood based inference methods for quite some years already, and has its root in the Approximate Bayes Computation (Rubin 1984), and it has been used in a variety of fields, from cosmology to particle physics. The main difference between SBI and likelihood based methods, like MCMC, is that the former do not require the likelihood function to be known, but rather rely on a simulator to generate synthetic data  $\mathbf{x}$  once the input parameters  $\theta$  are passed to it, and the inference pipeline is trained based on data-parameters pairs  $(\mathbf{x}, \theta)$ .

Recent advance of this technique was made possible by the use of machine learning models to emulate conditional probability distributions, a technique know as Neural Density Estimation (NDE) (Papamakarios 2019). The NDE is achieved by training a Normalizing Flow architecture, a generative model that allows to obtain samples from a complex distribution  $p(x)$  by constructing a series of **bijective** transformations  $f_{\phi_i}^i$  that map  $x$  to a latent space  $z$  that is distributed as a simple distribution, like a Gaussian. Accordingly to [5], in the end the models learns the following schema parameters  $\phi_i$ :

$$p(x) \sim x \xleftrightarrow{f_{\phi_1}^1} h_1 \xleftrightarrow{f_{\phi_2}^2} h_2 \dots \xleftrightarrow{f_{\phi_K}^K} z \sim \mathcal{N}(z; 0, I), \quad (3.1)$$

by maximizing the negative log likelihood as loss function and using the change of variable formula as follows:

$$\begin{aligned} \log p(x) &= \log p(z) + \log \left| \det \left( \frac{\partial z}{\partial x} \right) \right| \\ &= \log p(z) + \sum_{i=1}^K \log \left| \det \left( \frac{\partial h_i}{\partial h_{i-1}} \right) \right|, \end{aligned} \quad (3.2)$$

where the last term is sum of the log determinant of the Jacobian of the transformations  $f_{\phi_i}^i$ . Once the model is trained it is easy to sample from the distribution  $p(x)$  by sampling from the latent space  $z$  and applying the inverse transformations  $(f_{\phi_1}^1)^{-1} \circ \dots \circ (f_{\phi_K}^K)^{-1}$ . In order to keep the sum of log determinant tractable, the use of *Coupling layers* allows to split the input  $x$  along its dimensions and apply a transformation only to a subset of the dimensions, using the other as input for the transformation and keeping it fixed. The subset is than changed at each layers, allowing to have a

permutation invariant transformation. The transformations  $f_{\phi_i}^i$  are usually very simple invertible transformation like a translation and a scaling, or splines functions.

Following the discussion presented in [4], in Bayesian analysis we have the choice to approximate either the Posterior, the Likelihood or the Likelihood ratio, and this choice depend mostly on the problem that one wants to solve. In our case, due to the complexity of the Likelihood distribution of the chemical abundance space, we choose to approximate the Posterior distributions, and so we adopted the Neural Posterior Estimate that can be trained using the negative loglikelihood as loss function:

$$\begin{aligned}\mathcal{L}_{NPE}(\theta) &= -\mathbb{E}_{\mathcal{D}_{train}} \log \hat{\mathcal{P}}(\theta_i | x_i) \\ &= -\mathbb{E}_{\mathcal{D}_{train}} \log \left( \frac{p(\theta)}{\tilde{p}(\theta)} q_{\omega}(\theta_i, x_i) \right),\end{aligned}\tag{3.3}$$

where our Posterior distribution  $\hat{\mathcal{P}}(\theta_i | x_i)$  is approximated by the product of the ratio of the prior  $p(\theta)$  and proposal distribution  $\tilde{p}(\theta)$  and the neural conditional distribution  $q_{\omega}(\theta_i, x_i)$ , parametrized by the parameters  $\omega$ .

Many excellent framework for handling SBI analysis are already available, and CASBI is build on top of the `ltu-ili` python package [4]. In particular, CASBI analysis were performed relying on the `sbi` backend [7] to train a *Neural Posterior Estimate*<sup>1</sup> of the parameters' posteriors. The preprocessing of the data is described in Section 3.2, the details of the training of the NPE is described in Section 3.3, and the final package is described in Section ??.

## 3.2 NIHAO

The data-parameters pairs are obtained from a set of N-body simulations of Milky Way like galaxies, the **NIHAO** project [8]. The **NIHAO** (Numerical Investigation of a Hundred Astrophysical Objects) is a set of 100 cosmological zoom-in hydrodynamical simulations with halos that range from dwarf ( $M_{star} \sim 5 \times 10^9 M_{\odot}$ ) to Milky Way ( $M_{star} \sim 2 \times 10^{12} M_{\odot}$ ). In order to handle these simulations, in CASBI the preprocessing is done with the use of the functions available in `pynbody` [6]. Similarly to [1] and [2], we rely on the assumption that once the accreted object falls into the gravitational potential of the Milky Way like galaxy its star formation rate is halted, so we can treat each of the snapshot in this simulations as a possible building block of galactic halo. In order to create observables we construct 2D histogram, referred to as  $\mathbf{x}^i$ , of counts of the chemical abundance plane  $[O/Fe]$  over  $[Fe/H]$  ( $\alpha$  element abundance over metallicity) for each of the snapshot available in **NIHAO**, after filtering in order to have galaxies with a total stellar mass  $M_{star} < 6 \times 10^9 M_{\odot}$ , which is the stellar mass of the Large Magellanic Cloud, the largest accreted object by the Milky Way. These 2D histogram have  $64 \times 64$  pixels, they have minimum and maximum values set after filtering all the stars that were outside the 0.01 percentile in either metallicity or  $\alpha$  element abundance, and they are linked to galaxy snapshot trough the `Galaxy_name` attribute. The actual observable  $\mathbf{x}^j = \sum_i^{N_{sub}^j} x_i^j$  used in CASBI is then a super imposition of  $N_{sub}$  of these 2D histograms, where the  $N_{sub}^j$  is the number of accreted objects present in the  $j$ -th galaxy halo. In CASBI the inference is done on two set of parameters, the  $N_{sub}^j$  and the  $\theta^j = (M_{star,i}^j, M_{dm,i}^j, \tau_i^j)$  with  $i = 2, \dots, N_{sub}$ , which are respectively the total stellar mass, the total dark matter mass, and the infall time (in Gyr) of the  $i$ -th accreted object in the  $j$ -th sample. The inference pipeline is further described in the following section.

---

<sup>1</sup>The `sbi` backed implement NPE using `nflows` [3]

### 3.3 Two step Inference

The objective of the inference is not trivial, because in order to recover the parameters of the building blocks of the Milky Way like galaxy we need to know how many parameters we should look for in order to fix the dimensionality of the priors. This is equivalent to have perfect knowledge on the number of substructure that are present in the galactic halo. In the case of not fully phase mixed structure, the dynamical information could be use to help to disentangle this structure, and also to separate them from the host halo background. In CASBI we do not leverage on this information because it would be hard to construct unbiased couples of observables-parameters, and we leave this task for future work. Instead we separate the inference in two steps:

1. **Inference of the number of substructure:** In this step we train a NPE to recover the posterior distribution of the number of substructure  $N_{sub}$ , by using the observable  $\mathbf{x}^j$ . The prior for the parameter is assumed to be uniform between 2 and 100. The boundaries were selected in accordance to the order of magnitude of substructures found in [2]. For each of the possible  $N_{sub}$  we extract 1000 random samples of  $N_{sub}$  snapshots from the **NIHAO** simulations, and we construct the observable  $\mathbf{x}^j$  as described in Section 3.2, so in total we have  $1000 \times 100 = 10^5$  training samples, with 20 % used as validation, and we use the same process to generate  $100 \times 100 = 10^4$  test set samples, making sure that the same combinations of **Galaxy\_name** attribute weren't shown in training and test. The training of the NPE is done using the **sbi** backend, using 4 **nsf** (neural spline flow) with 10 layers and 100 neurons each. In order to take full advantage of the image-like structure of the data, we adopt a CNN embedding network to reduce the dimensionality of the input of the NPE from  $64 \times 64$  to 128. The learning rate was set to ..., the batch size was set to .... The hyperparameter were tuned using the **optuna** package by maximizing the validation log likelihood. In this step we have not impose the constrain that the  $N_{sub}$  must be a discrete variable, and we have decide to approximate the sample estimate to an integer. To the knowledge of the author no SBI framework has implemented a way of dealing with the inference of discrete random variables, so we leave a more precise implementation as a future work. We propose instead another method to obtain the number of substructure, by casting this inference as a classification problem. We use a SkipConnection CNN, considering the number of substructure as the label to assign to each  $x^j$ . The performance for both method are reported in Section 4.
2. **Inference of  $\theta^j$ :** Once we have the estimate  $\tilde{N}_{sub}$ , whether using dynamical information, or the inference pipeline or the classification method, we can proceed to the inference of the parameters  $\theta^j$ . The prior for the parameters are assumed to be uniform between the minimum and maximum values available for the galaxies that we have filtered from the **NIHAO** simulations. We extract  $10^5$  random samples of  $N_{sub}$  snapshots from the **NIHAO** simulations, and we construct the observable  $\mathbf{x}^j$  as described in Section 3.2, with 20 % used as validation and the rest as training. We repeat the same process to generate  $10^3$  test set samples, making sure that the same combinations of **Galaxy\_name** attribute weren't shown in training and test to perform calibration of the inference model. The training of the NPE is done using the **sbi** backend, using 4 **nsf** (neural spline flow) with 10 layers and 100 neurons each. Once again we use a CNN as embedding for our observation  $x^j$ . The learning rate was set to ..., the batch size was set to .... The hyperparameter were tuned using the **optuna** package by maximizing the validation log likelihood.

The performance and calibration of the two inference pipeline are presented in the section 4.



### 3.4 Surrogate Simulator

## Chapter 4

# Analysis

### 4.1 NIHAO UHD

## Chapter 5

# Conclusion

### 5.1 Future work

#### 5.1.1 GRUMPY

#### 5.1.2 Number halos has a free parameter: hierarchical sbi

#### 5.1.3 True test: GAIA

# Bibliography

- [1] Emily C. Cunningham, Robyn E. Sanderson, Kathryn V. Johnston, Nondh Panithanpaisal, Melissa K. Ness, Andrew Wetzel, Sarah R. Loebman, Ivanna Escala, Danny Horta, and Claude-André Faucher-Giguère. Reading the CARDS: The Imprint of Accretion History in the Chemical Abundances of the Milky Way’s Stellar Halo. *The Astrophysical Journal*, 934(2):172, August 2022.
- [2] Alis J Deason, Sergey E Koposov, Azadeh Fattahi, and Robert J J Grand. Unravelling the mass spectrum of destroyed dwarf galaxies with the metallicity distribution function. *Monthly Notices of the Royal Astronomical Society*, 520(4):6091–6103, February 2023.
- [3] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Nflows: Normalizing flows in PyTorch. Zenodo, November 2020.
- [4] Matthew Ho, Deaglan J. Bartlett, Nicolas Chartier, Carolina Cuesta-Lazaro, Simon Ding, Axel Lapel, Pablo Lemos, Christopher C. Lovell, T. Lucas Makinen, Chirag Modi, Viraj Pandya, Shivam Pandey, Lucia A. Perez, Benjamin Wandelt, and Greg L. Bryan. LtU-ILI: An All-in-One Framework for Implicit Inference in Astrophysics and Cosmology, February 2024.
- [5] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions, July 2018.
- [6] Andrew Pontzen, Rok Roškar, Greg Stinson, and Rory Woods. Pynbody: N-Body/SPH analysis for python. *Astrophysics Source Code Library*, page ascl:1305.002, May 2013.
- [7] Alvaro Tejero-Cantero, Jan Boelts, Michael Deistler, Jan-Matthis Lueckmann, Conor Durkan, Pedro J. Gonçalves, David S. Greenberg, and Jakob H. Macke. Sbi: A toolkit for simulation-based inference. *Journal of Open Source Software*, 5(52):2505, August 2020.
- [8] Liang Wang, Aaron A. Dutton, Gregory S. Stinson, Andrea V. Macciò, Camilla Penzo, Xi Kang, Ben W. Keller, and James Wadsley. NIHAO project I: Reproducing the inefficiency of galaxy formation across cosmic time with a large sample of cosmological hydrodynamical simulations. *Monthly Notices of the Royal Astronomical Society*, 454(1):83–94, November 2015.