



Modelling the redshift-space cluster–galaxy correlation function on Mpc scales with emulation of the pairwise velocity distribution

Andrew Robertson ¹*, Eric Huff ¹, Katarina Markovič ¹ and Baojiu Li ²

¹*Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA*

²*Institute for Computational Cosmology, Department of Physics, Durham University, South Road, Durham DH1 3LE, UK*

Accepted 2024 June 11. Received 2024 June 6; in original form 2024 January 23

ABSTRACT

We present a method for modelling the cluster–galaxy correlation function in redshift space, down to \sim Mpc scales. The method builds upon the so-called galaxy infall kinematics (GIK) model, a parametric model for the pairwise velocities of galaxies with respect to nearby galaxy clusters. We fit the parameters of the GIK model to a suite of simulations run with different cosmologies, and use Gaussian processes to emulate how the GIK parameters depend upon cosmology. This emulator can then be combined with knowledge of the real-space clustering of clusters and galaxies, to predict the cluster–galaxy correlation function in redshift space, ξ_{cg}^s . Fitting this model to an observed ξ_{cg}^s enables the extraction of cosmological parameter constraints, and we present forecasts for a survey like that currently being done by the Dark Energy Spectroscopic Instrument (DESI). We also perform tests of the robustness of our constraints from fitting to mock data extracted from N -body simulations, finding that fitting to scales $\lesssim 3 h^{-1}$ Mpc leads to a biased inference on cosmology, due to model mis-specification on these scales. Finally, we discuss what steps will need to be taken in order to apply our method to real data.

Key words: galaxies: clusters: general – galaxies: haloes – large-scale structure of Universe – cosmology: theory.

1 INTRODUCTION

Understanding the cause of the accelerated expansion of the late-time Universe is one of the primary goals of 21st century cosmology (Frieman, Turner & Huterer 2008). While current observational data appears to be consistent with the expansion being driven by a cosmological constant, Λ , an exciting alternative is that the accelerated expansion is pointing towards gravity behaving in a different manner from the predictions of general relativity (GR; Joyce, Lombriser & Schmidt 2016). There are numerous modified gravity theories, which – as well as providing potential explanations for the accelerated expansion – generally predict regimes in which there are enhancements to the total gravitational force, above that expected with GR.

Because gravity on the scale of the Solar system is known to be consistent with GR to high precision, these additional forces must somehow be screened within the Solar system. Screening mechanisms typically lead to the enhancements to gravity disappearing in high-density, or deep gravitational potential environments (Brax 2013). In this context, the infall of galaxies onto galaxy clusters provides an interesting test of these modified gravity theories, because galaxies will transition from being unscreened to screened as they fall into the cluster potential (Zu et al. 2014).

Clusters and their masses are also an interesting area for current exploration, because of current tensions between measurements of cosmological parameters from cluster abundance studies and measurements made using other methods. For example, the Dark Energy

Survey’s (DES) Year 1 Results indicated that the number and/or masses of galaxy clusters disagreed with that expected from the cosmology fit to combined cosmic shear, galaxy–galaxy lensing, and photometric galaxy clustering (Abbott et al. 2020). To quantitatively explain these results, the masses inferred through weak lensing for DES’s lowest richness clusters would have to be 30–40 per cent lower than expected from the DES 3×2 pt cosmology, or else the richness-selected sample of clusters must be highly (~ 50 per cent) incomplete.

The cluster–galaxy cross-correlation function in redshift space, ξ_{cg}^s , is an observable quantity that depends both on the clustering of galaxies around galaxy clusters, and on the relative velocities of galaxies with respect to nearby clusters. These velocities in turn depend upon the mass distribution of clusters, as well as the theory of gravity. $\xi_{\text{cg}}^s(r_p, r_\pi)$ measures the excess probability (above that expected for randomly distributed clusters and galaxies) of measuring cluster galaxy pairs with an inferred separation along the line of sight of r_π , and a separation perpendicular to this of r_p . Cluster–galaxy pairwise velocities get imprinted into $\xi_{\text{cg}}^s(r_p, r_\pi)$ because the line-of-sight separations between cluster–galaxy pairs are inferred from differences in their redshifts, which – as well as depending upon the true line-of-sight separations – receive a contribution from the line-of-sight components of their peculiar velocities. The effects of peculiar velocities on the observed clustering signal are known as redshift-space distortions (RSDs).

If we wish to use RSD in the cluster–galaxy correlation function to probe cluster mass distributions as well as modifications to GR, then we need to include data from the small scales on which the cluster potential makes a significant contribution to the galaxy velocities. Clustering on these small scales also contains a large fraction of the statistical constraining power from current and future surveys, mak-

* E-mail: andrew.a.robertson91@gmail.com

ing the accurate modelling of redshift-space clustering on these scales a crucial ingredient for extracting the maximum amount of information. For example, for a BOSS-like survey, the constraints on the growth rate of structure from $r < 30 h^{-1}$ Mpc scales alone are nearly a factor of two tighter than those from the fiducial Baryon Oscillation Spectroscopic Survey (BOSS) analysis of redshift-space clustering using perturbation theory on larger scales (Zhai et al. 2019).

Given the ever increasing volume and resolution of N -body simulations, one approach to modelling galaxy clustering on small scales is to directly use the results from a simulation with a given cosmology as the model prediction for that cosmology. In practice, simulations are still too time consuming to run one for every likelihood evaluation in a cosmological likelihood analysis. However, assuming that the simulated data vector varies smoothly as the cosmological parameters are varied, some sort of interpolation scheme can be used to predict the data vector at an arbitrary cosmology, from the data vector calculated from simulations run at other cosmologies.

This procedure is known as emulation (see Euclid Collaboration 2019; Zhai et al. 2019; Yuan et al. 2022, for a few different examples in cosmology). Emulating some cosmological observable from simulations requires that we trust the simulations to reliably predict the observable. It also requires simulation volumes comparable with or larger than observed survey volumes, so that noise in the simulations does not dominate the error budget – although there have been recent developments to suppress the variance in simulation predictions (e.g. Angulo & Pontzen 2016; Maion, Angulo & Zennaro 2022; Kokron et al. 2022).

Here, we explore an alternative to directly emulating observables, which is to use a physically motivated model, the parameters of which we expect to depend systematically on the values of the cosmological parameters. We can then build an emulator for the parameters of the model, rather than for the observed data vector itself. This approach has some advantages and disadvantages over straight emulation of the data vector (discussed in Section 5.1). In short, a physical model can suppress noise and also aids with physical understanding, but opens the door to the potential for model mis-specification. We build such a model for the redshift-space cluster–galaxy correlation function, employing the galaxy infall kinematics (GIK) model from Zu & Weinberg (2013, ZW13 hereafter) to describe the distribution of cluster–galaxy pairwise velocities. We build an emulator for the parameters of the GIK model using the FORGE suite of $f(R)$ modified gravity simulations (Arnold et al. 2022), and then demonstrate fitting this to mock observations of the cluster–galaxy correlation function in redshift space.

This paper is structured as follows. In Section 2, we provide some basic definitions useful in the rest of the paper. Then in Section 3, we introduce the GIK model from ZW13, highlighting some small changes that we have made. In Section 4, we demonstrate how to combine the real-space correlation function with the GIK model in order to calculate the redshift-space correlation function. Then in Section 5, we describe using simulations run with different cosmological parameters to build an emulator for the GIK model parameters. In Section 6, we demonstrate the efficacy of our model, by fitting it to mock data. In Section 7, we outline some of the advancements required before our method could be applied to real observational clustering data, before concluding in Section 8.

2 THE CLUSTER–GALAXY CORRELATION FUNCTION IN REDSHIFT SPACE

The cluster–galaxy correlation function in real space, $\xi_{\text{cg}}(r)$, is defined such that for a sample of galaxies with number density n_g

and galaxy clusters with number density n_c ,

$$dP_{\text{cg}} = n_c n_g [1 + \xi_{\text{cg}}(r)] dV_c dV_g \quad (1)$$

is the joint probability of finding a galaxy cluster in the volume dV_c and a galaxy in the volume dV_g , where the separation between the two (small) volumes is r . This probability (and hence $\xi_{\text{cg}}(r)$) depends only on the distance between the two volumes (and not on the direction of their separation or their absolute positions) because of the assumptions of isotropy and homogeneity.

Calculating $\xi_{\text{cg}}(r)$ is rather simple for the case of clusters and galaxies found within a cosmological simulation in a cubic (periodic) box. For some definition of ‘galaxy’ and ‘galaxy cluster’, we can count the number of galaxy–cluster pairs that have a separation between r and $r + dr$, which we call $dN_{\text{cg}}^{\text{sim}}$. Assuming that the total simulation volume is V , the expected number of clusters is $n_c V$. If we just consider one of these clusters, then the expected number of galaxies (if they were randomly distributed) a distance between r and $r + dr$ away from the cluster is $4\pi r^2 dr n_g$, such that when accounting for all clusters we expect $dN_{\text{cg}}^{\text{rand}} = 4\pi r^2 dr n_g n_c V$ pairs. Our estimate for the cluster–galaxy correlation function is then $\xi_{\text{cg}}(r) = dN_{\text{cg}}^{\text{sim}} / dN_{\text{cg}}^{\text{rand}} - 1$.

This same notion can be extended to the case of anisotropic clustering in redshift space, where the isotropic nature of galaxy clustering is broken by RSD. In an analogous manner to the real-space clustering case discussed above, we can count the number of cluster–galaxy pairs that have a separation with line-of-sight component in the range $[r_\pi, r_\pi + dr_\pi]$, and a plane-of-sky component in the range $[r_p, r_p + dr_p]$, and compare this with the expectation in the case of randomly distributed clusters and galaxies to find $\xi_{\text{cg}}^s(r_p, r_\pi)$, where the superscript s denotes that this correlation function is measured in redshift space.

2.1 The INDRA simulations

We use the publicly available INDRA suite of simulations (Falck et al. 2021). These will be useful for first elucidating some of the workings of the GIK model, and then later for calculating a covariance matrix for $\xi_{\text{cg}}^s(r_p, r_\pi)$. The INDRA suite consists of 384 simulations, all run with the same cosmological parameters as one another, but with different random phases used for the initial conditions. The simulations are dark-matter-only (DMO), and are of boxes with a side length of $1 h^{-1}$ Gpc, with a particle mass of $7 \times 10^{10} h^{-1} M_\odot$. In total, each INDRA simulation has 64 snapshots saved from the time of the initial conditions (at a redshift of $z = 127$) down to $z = 0$, but for simplicity we focus only on $z = 0$ in this paper.

The INDRA snapshots were analysed with the standard friends-of-friends (FOF) algorithm (Davis et al. 1985), after which the SUBFIND algorithm (Springel et al. 2001) was run, which calculates spherical overdensity quantities such as M_{200} for each FOF group,¹ and also identifies gravitationally bound substructures within the FOF groups.

While there are numerous possible ways that one could populate a DMO simulation with galaxies, and then also identify galaxy clusters, for now we will take a simplistic approach. We consider galaxy clusters to be all FOF groups with M_{200} in some range, while galaxies are all SUBFIND subhaloes with v_{max} above some

¹Centred on the most gravitationally bound particle in the FOF group, we define r_{200} as the radius within which the mean enclosed density is $200 \times \rho_{\text{crit}}$, with the mass enclosed within r_{200} being M_{200} .

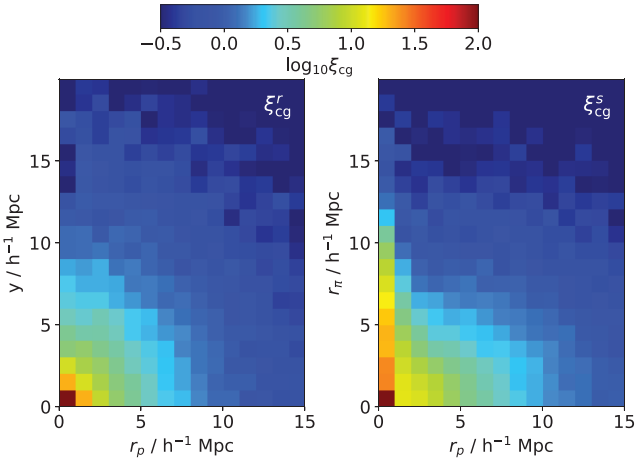


Figure 1. The measured small-scale cluster–galaxy correlation function from the $z = 0$ snapshot of an INDRA simulation. The left-hand panel shows the correlation function in real space, with y the true line-of-sight separation between cluster–galaxy pairs, while the right-hand panel uses the inferred line-of-sight separation (i.e. the separation in redshift space), r_π .

threshold value. Here, v_{\max} is the maximum value of the circular velocity, $v_{\text{circ}}(r) = \sqrt{GM(<r)/r}$, where $M(<r)$ is the mass enclosed within a radius r . Note that SUBFIND subhaloes include the central subhalo in each FOF group, so being tied to ‘subhaloes’ does not imply that our galaxies are necessarily substructures of something larger. Our galaxy assignment scheme can be thought of as subhalo abundance matching (SHAM), using v_{\max} as the abundance matching parameter and with zero scatter (Reddick et al. 2013).

For a cluster and galaxy at locations \mathbf{r}_c and \mathbf{r}_g , and with peculiar velocities \mathbf{v}_c and \mathbf{v}_g , respectively, we define $\mathbf{r} = \mathbf{r}_g - \mathbf{r}_c$ and $\mathbf{v} = \mathbf{v}_g - \mathbf{v}_c$ as the relative position and velocity of the galaxy with respect to the cluster. We also define the scalar cluster–galaxy separation, $r = |\mathbf{r}|$. We denote a unit vector along the line of sight as $\hat{\mathbf{y}}$, with the line-of-sight separation between the cluster and galaxy being $y = \mathbf{r} \cdot \hat{\mathbf{y}}$, and the separation perpendicular to this being $r_p = |\mathbf{r} - y\hat{\mathbf{y}}|$. The line-of-sight component of the relative velocity between the cluster and galaxy is $v_y = \mathbf{v} \cdot \hat{\mathbf{y}}$. The relative velocity along the line-of-sight changes the apparent line-of-sight separation between the galaxy and cluster to be $r_\pi = y + (1+z)v_y/H(z)$, where the factor of $1+z$ accounts for the fact that we define r_π and y in comoving coordinates.

For a minimum galaxy v_{\max} of 250 km s^{-1} , and clusters with $14.1 < \log_{10} M_{200}/h^{-1} M_\odot < 14.2$, we plot the cluster–galaxy correlation function in Fig. 1. For this purpose, we adopt one of the Cartesian axes of the simulation box as the line of sight, and then show the results both in real space (where the clustering is isotropic)² and in redshift space, where Kaiser (1987) squashing enhances the clustering at intermediate- r_p and low- r_π , and the fingers of god spread low- r cluster–galaxy pairs along the line of sight.

This cluster definition leads to ~ 3000 clusters in each $(1 h^{-1} \text{ Gpc})^3$ INDRA volume, while the number density of galaxies is $n_g \sim 10^{-3} (h^{-1} \text{ Mpc})^{-3}$. This galaxy number density is comparable with what is expected for some upcoming galaxy redshift survey

²While the mean clustering signal is isotropic, the noise is not because (e.g.) a pixel at $(r_p = 10.5 h^{-1} \text{ Mpc}, y = 0.5 h^{-1} \text{ Mpc})$ covers a larger volume than the pixel at $(r_p = 0.5 h^{-1} \text{ Mpc}, y = 10.5 h^{-1} \text{ Mpc})$, such that the first of these has smaller fractional error due to Poisson counts.

samples, such as DESI’s Luminous Red Galaxies which will have $n_g \approx 0.5 \times 10^{-3} (h^{-1} \text{ Mpc})^{-3}$ (Zhou et al. 2023).

3 THE ‘GALAXY INFALL KINEMATICS’ (GIK) MODEL

Our approach to modelling ξ_{cg}^s is built upon the GIK model introduced in ZW13. The GIK model describes the velocity distribution of galaxies with respect to nearby clusters, as a function of cluster–galaxy separation. At each cluster–galaxy separation, there are seven parameters. We will call these seven parameters the GIK functions, because they are functions of the cluster–galaxy separation, r , and because this allows us to use the term GIK parameters for another set of parameters that we will describe shortly.

3.1 The GIK functions

The seven GIK functions specify the joint probability distribution of radial and tangential velocity components, $P(v_r, v_t|r)$. Continuing on from our definitions of \mathbf{r} and \mathbf{v} above, we define a unit vector in the separation direction, $\hat{\mathbf{r}} = \mathbf{r}/r$. Then, $v_r = \mathbf{v} \cdot \hat{\mathbf{r}}$ is the radial velocity for this cluster–galaxy pair, where negative v_r represents a galaxy moving towards (i.e. falling in to) a cluster. Note that because we use peculiar velocities, the Hubble flow does not contribute to v_r ; put another way, a cluster–galaxy pair with $v_r = 0$ will have a proper separation that is growing in time due to the expanding universe. The total tangential velocity is $\mathbf{v}_T = \mathbf{v} - v_r \hat{\mathbf{r}}$. For the purpose of the GIK model, it is useful to follow ZW13 in defining the tangential velocity, v_t , as the projection of \mathbf{v}_T onto the plane containing the cluster galaxy separation, \mathbf{r} , and the adopted line of sight, $\hat{\mathbf{y}}$. Specifically, $v_t = (\mathbf{v} \cdot \hat{\mathbf{y}} - v_r \sin \theta) / \cos \theta$, where θ is the angle between \mathbf{r} and the plane of the sky, that is, $\sin \theta = \mathbf{r} \cdot \hat{\mathbf{y}}/r$.

A diagram of these various vector definitions is provided in fig. 2 of ZW13. To aid with intuition, we note that with v_r and v_t defined in this manner, if the distribution of \mathbf{v} is a zero-mean 3D Gaussian with equal dispersion along three orthogonal axes, then $P(v_r, v_t)$ will be a 2D Gaussian, centred on $v_r = v_t = 0$, and with equal dispersion in the v_r and v_t directions.

In Fig. 2, we show $P(v_r, v_t)$ for cluster–galaxy separations in the range $1 < r/h^{-1} \text{ Mpc} < 1.25$, for a stack of cluster–galaxy pairs from 24 INDRA simulations, using the same cluster and galaxy definitions as before. This radial shell corresponds to galaxies at distances close to the clusters’ virial radii, and an interesting feature of $P(v_r, v_t)$ is that it is bimodal (see also García et al. 2023). This demonstrates the motivation for the functional form of $P(v_r, v_t)$ in the GIK model, which is the sum of two distinct components: a virialized component, making up a fraction f_{vir} of galaxies in this radial shell, and an infalling component, which makes up the rest. The expectation is that f_{vir} is only appreciably non-zero at small radii, approximately within the splashback radii (More, Diemer & Kravtsov 2015) of the clusters.

The virialized component is modelled with a simple velocity distribution, which is an isotropic Gaussian, with a dispersion along each orthogonal direction of σ_{vir} . The infalling component has a more complex velocity distribution, known as a skewed- t distribution (Azzalini & Capitanio 2009). The expression for the infalling component’s velocity distribution is

$$\mathcal{T}(\mathbf{v}) = 2 t_2(\mathbf{v} - \mathbf{v}_{r,c}; \text{dof}, \Sigma) \times T_1 \left(\frac{\alpha(v_r - v_{r,c})}{\sigma_{\text{rad}}} \frac{\text{dof} + 2}{Q_v + \text{dof}}; \text{dof} + 2 \right), \quad (2)$$

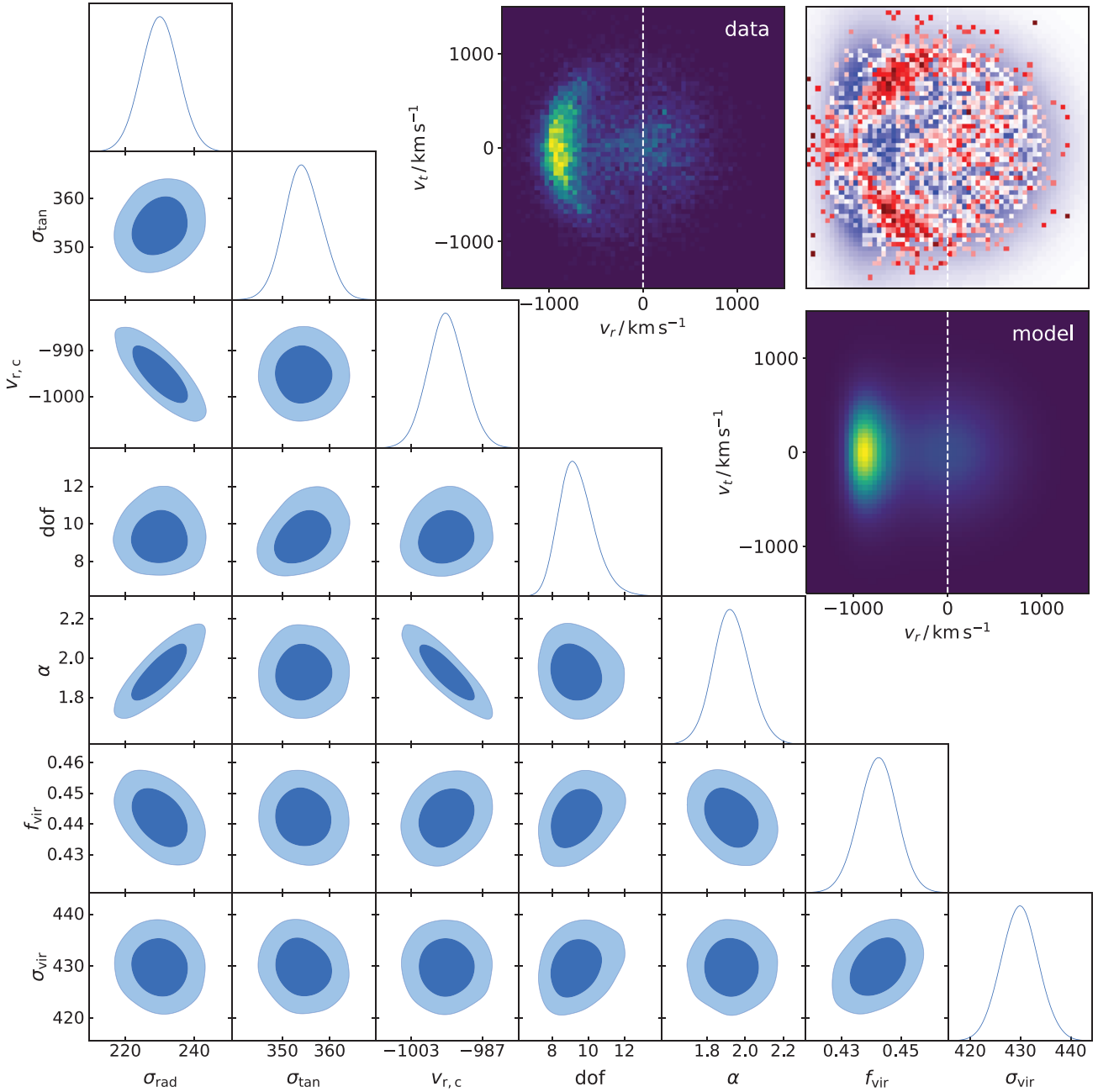


Figure 2. A corner plot showing the posterior distribution for the GIK functions fit to data generated from a stack of 24 INDRA simulations. The cluster sample is all haloes with $14.1 < \log_{10} M_{200}/h^{-1} M_{\odot} < 14.2$, with galaxies defined as subhaloes with $v_{\max} > 250 \text{ km s}^{-1}$, and is for cluster–galaxy separations in the range $1 < r/h^{-1} \text{ Mpc} < 1.25$. The data and best-fitting model for $P(v_r, v_t)$ are plotted, using the same colour scale in the two panels. The normalized residuals (data minus model, divided by a Gaussian approximation to the Poisson uncertainty) are plotted in the top right, with red corresponding to positive residuals, and blue negative.

where t_2 is the density function of a 2D t -variate with dof degrees of freedom,³ T_1 is the cumulative distribution function for a scalar t -variate with $\text{dof} + 2$ degrees of freedom,⁴ $\mathbf{v}_{r,c} = (v_{r,c}, 0)$ is a typical infall velocity (equal to the mean infall velocity for the case of zero

skewness), and $Q_v = (\mathbf{v} - \mathbf{v}_{r,c})^T \Sigma^{-1} (\mathbf{v} - \mathbf{v}_{r,c})$, where

$$\Sigma = \begin{pmatrix} \sigma_{\text{rad}}^2 & 0 \\ 0 & \sigma_{\text{tan}}^2 \end{pmatrix}. \quad (3)$$

The basic idea is that $t_2(\mathbf{v} - \mathbf{v}_{r,c}; \text{dof}, \Sigma)$ is a 2D t -distribution, which looks Gaussian in the limit $\text{dof} \rightarrow \infty$, but is leptokurtic (i.e. with fatter tails than a Gaussian with the same variance) for finite dof . This 2D t -distribution has variances of σ_{rad}^2 and σ_{tan}^2 in the radial and tangential directions, respectively. The t_2

³Implemented using the `pdf` method of a `scipy.stats.multivariate_t` object.

⁴Implemented using the `cdf` method of a `scipy.stats.t` object.

distribution is symmetric about the typical infall velocity, $v_{r,c}$, with the skewness in $\mathcal{T}(\mathbf{v})$ introduced through multiplying t_2 by the cumulative distribution function T_1 .

The probability density function $dT_1(x; \text{dof} + 2)/dx$ is symmetric about $x = 0$, such that $T_1(x; \text{dof} + 2)$ is 0 as $x \rightarrow -\infty$, 1 as $x \rightarrow \infty$ and $1/2$ at $x = 0$. α controls the skewness of the velocity distribution (in the v_r direction), with $\alpha = 0$ producing a distribution with no skewness, and positive α producing a distribution with a lengthened tail towards positive v_r .

The complete velocity distribution reads

$$P(\mathbf{v}) = f_{\text{vir}} \mathcal{G}(\mathbf{v}) + (1 - f_{\text{vir}}) \mathcal{T}(\mathbf{v}), \quad (4)$$

where $\mathcal{G}(\mathbf{v})$ is a 2D isotropic Gaussian centred on $\mathbf{v} = 0$. In summary, the seven GIK functions are:

- (i) σ_{rad} – the velocity dispersion of the infalling component in the radial direction.
- (ii) σ_{tan} – the velocity dispersion of the infalling component in the tangential direction.
- (iii) $v_{r,c}$ – the typical radial velocity of the infalling component.
- (iv) dof – the degrees of freedom associated with the infalling component’s 2D t distribution.
- (v) α – the skewness in the radial direction of the infalling component’s velocity distribution.
- (vi) f_{vir} – the fraction of galaxies in the virialized component.
- (vii) σ_{vir} – the 1D velocity dispersion of the virialized component.

In Fig. 2, we show the posterior distribution for the seven GIK functions fit to the data shown in the same figure. For a given radial bin, we measure the galaxy velocity distribution by counting galaxies in velocity bins, with 60×60 equal-sized bins covering v_r and v_t in the range $-1500 \text{ km s}^{-1} \leq v_r, v_t \leq 1500 \text{ km s}^{-1}$. We label the galaxy count in a given pixel n_{ij} , where i and j index the pixels along the v_r and v_t directions respectively. To evaluate the likelihood for a given set of GIK function values, we calculate the probability of getting the measured set of n_{ij} assuming that the number of galaxies in a pixel is a Poisson process with an expectation given by the model probability density at the centre of the pixel multiplied by the pixel area and the total number of cluster–galaxy pairs in this radial bin. Specifically, the likelihood is

$$\mathcal{L} = \prod_{i,j} \lambda_{ij}^{n_{ij}} \exp(-\lambda_{ij}) / n_{ij}^{\text{sim}!} \quad (5)$$

where $\lambda_{ij} = P(v_{r,i}, v_{t,j})(\delta v)^2 N$, with $\delta v = (v_{r,i+1} - v_{r,i}) = (v_{t,j+1} - v_{t,j}) = 50 \text{ km s}^{-1}$ the side length of a velocity-space pixel, and N the number of cluster–galaxy pairs in this radial shell. Combining this likelihood with flat priors on the seven GIK functions covering a range considerably broader than the resulting posteriors, we use the Markov chain Monte Carlo (MCMC) sampler EMCEE (Foreman-Mackey et al. 2013) to find the posterior distribution for the GIK functions.

We note that in Fig. 2 the model does a reasonable job of capturing the general features of the measured $P(v_r, v_t)$, however it clearly does not fit the data down to the noise level (associated with a stack of 24 $(1 h^{-1} \text{ Gpc})^3$ simulation volumes), with there being clear structure in the map of the residuals. In particular, the infalling component in the data has curvature in the v_r – v_t plane, such that galaxies falling in with larger (negative) radial velocities tend to have lower tangential velocities, which is not something that can be captured by the functional form of $\mathcal{T}(\mathbf{v})$ in equation (2).

3.2 The GIK parameters

If we perform a fit similar to that shown in Fig. 2 for a set of r bins, we can then see how the GIK functions vary with radius. An example of this is shown by the markers in Fig. 3, where now the results are for a single INDRA simulation (as opposed to the stack used for Fig. 2, to highlight subtle deficiencies in the model), and we plot the results for three different cluster mass bins, all with the same galaxy definition of any subhalo with $v_{\text{max}} > 250 \text{ km s}^{-1}$.

As well as introducing the form for the velocity distribution in terms of the GIK functions (equation 4), ZW13 presented parametric models for how the GIK functions vary with radius. For example, in ZW13 the functional form for $\sigma_{\text{rad}}(r)$, $\sigma_{\text{tan}}(r)$, $\alpha(r)$, and $\text{dof}(r)$ are all the same:

$$f(r) = q - p \frac{r}{(r + r_i)^\beta}, \quad (6)$$

where q , p , r_i , and β are free parameters (which take on different values for each of the four GIK functions to which this functional form is applied). We note that, written in this way, this particular functional form has the undesirable property that the dimensions of p depend on β . For this, as well as some other reasons discussed below, we choose to reparametrize this equation. In particular, we write equation (6) as

$$f(r) = q - h \frac{r}{r_{\text{min}}} \left(\frac{\beta r_{\text{min}}}{r + (\beta - 1) r_{\text{min}}} \right)^\beta, \quad (7)$$

where $r_{\text{min}} = r_i / (\beta - 1)$ is the radius at which $f(r)$ is minimized, and h can be expressed in terms of p , r_i , and β . While this new expression appears more complex than the one it replaces, it leads to free parameters with better characteristics than those of equation (6). In particular, q and h have the same dimensions as f [e.g. they are velocities when $f(r)$ is $\sigma_{\text{tan}}(r)$], and the minimum of $f(r)$ occurs at $r = r_{\text{min}}$, with $f(0) = q$ and $f(r_{\text{min}}) = q - h$. Aside from the reparametrization being useful for one’s intuition (r_{min} has a clear meaning, while r_i did not), it also decreases the covariance between the model parameters in the fit. As an example, the location of the minimum of $\sigma_{\text{rad}}(r)$ in Fig. 3 is clearly well constrained by the data, such that we get a tight constraint on r_{min} , whereas this would correspond to some degenerate set of solutions in terms of r_i and β .

For similar reasons, we reparametrized the functional forms for the majority of GIK functions from those that appear in ZW13. We describe this in Appendix A. In total, we end up with 24 parameters that describe the variation of the seven GIK functions with radius; we call these 24 parameters the GIK parameters.

Note that if one simply fits the seven GIK functions to $P(v_r, v_t)$ at large radii ($\gtrsim 10 h^{-1} \text{ Mpc}$) the values of f_{vir} tend to be fairly large ($\gtrsim 0.3$). Physically, these clearly do not correspond to galaxies ‘virialized’ within the cluster, but instead result from the fact that sufficiently far from the cluster the infalling component has a small mean radial velocity, such that it can be approximately described by the $\mathcal{G}(\mathbf{v})$ function centred on $v_r = 0$. For the purpose of plotting the GIK function posteriors (points with error bars) in Fig. 3, we circumvent this problem by first fitting the full seven parameter model for $P(v_r, v_t)$ in our radial bins and finding the radius at which f_{vir} is smallest, which we label r_f . We then fit just a five parameter model (with $f_{\text{vir}} = 0$) to all radial bins with $r > r_f$.

3.2.1 Fitting for the GIK parameters

While the process visualized in Fig. 3 – of measuring the seven GIK functions in different r bins – is useful conceptually, in practice

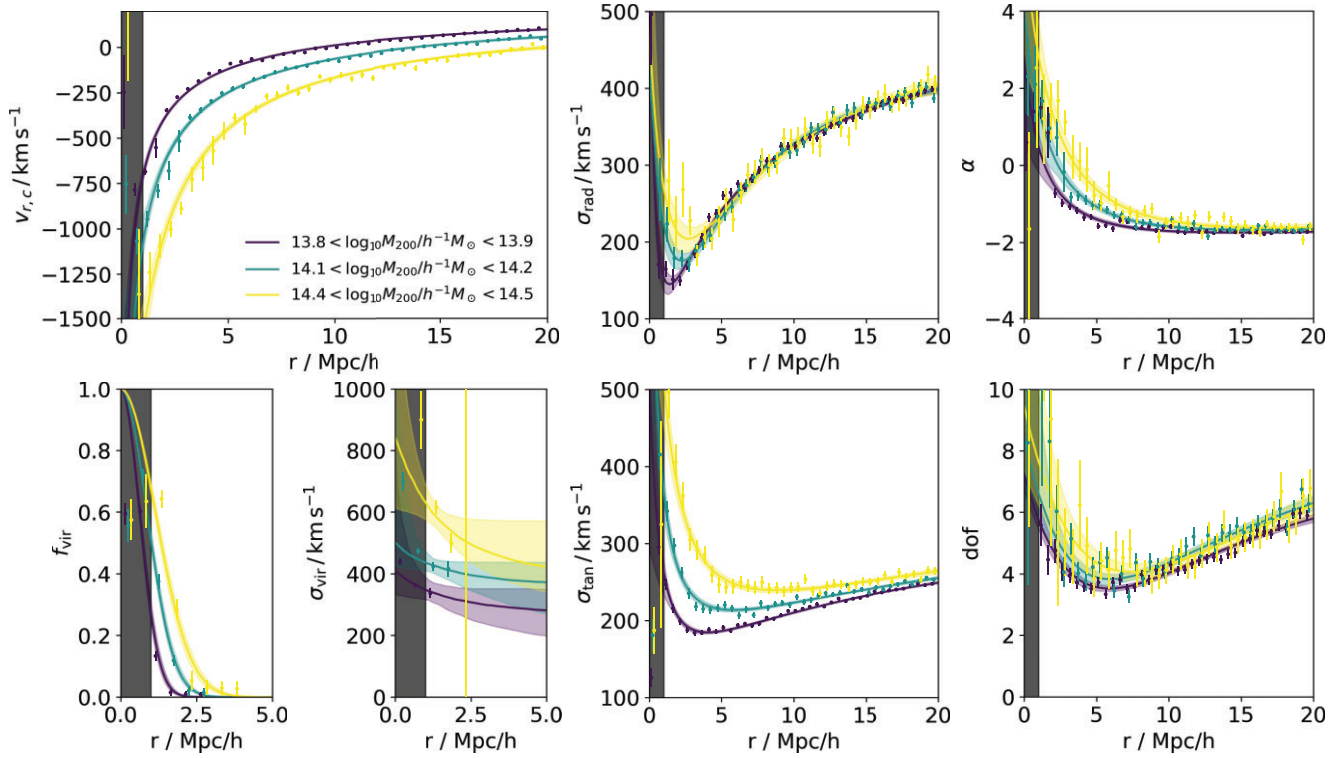


Figure 3. The variation of the GIK functions with radius. The points with error bars represent the mean \pm one standard deviation of the marginalized posteriors from fitting the GIK functions – separately at each radius – to $P(v_r, v_t)$ (e.g. as shown in Fig. 2). Generating posterior samples for the GIK parameters (using MCMC) and then calculating the corresponding GIK functions produces the solid lines with shaded regions. For each GIK parameter sample, we calculate $v_{r,c}(r)$, $f_{\text{vir}}(r)$, etc., and then the shaded region covers the 2.5th–97.5th percentiles of the GIK function at each radius, with the solid line being the median (50th percentile). The functional forms of the GIK functions are detailed in Appendix A. We note that σ_{vir} is poorly constrained at radii where $f_{\text{vir}} \approx 0$, which leads to the vertical line (an error bar extending off the figure) in the σ_{vir} panel.

we fit the 24 GIK parameters directly to the $P(v_r, v_t|r)$ data, simultaneously fitting to all radial bins. As a default we use 40 r bins, with the bin edges uniformly spaced from 0 to $20 h^{-1}$ Mpc. The likelihood we use is the product over the different radial bins of the likelihood defined in equation (5), with the model $P(v_r, v_t)$ at all radii calculated directly from a common set of GIK parameters.

This direct fitting procedure has a number of advantages over a two-step process of finding the GIK functions at different radii and then fitting the GIK parameters to these (which would amount to fitting the curves described by the GIK parameters to the points with error bars in Fig. 3). In particular, we can naturally account for the covariance between the different GIK parameters, even those relating to different GIK functions; we avoid the need to fix $f_{\text{vir}} = 0$ at large r ; we avoid the need to define priors on the GIK functions, instead only requiring priors on the GIK parameters; and we also avoid taking the (potentially) non-Gaussian GIK function posteriors, and treating them as Gaussian for the purpose of then fitting the GIK parameters.

3.2.2 Fitting the GIK parameters without binning

An alternative to the likelihood defined in equation (5) is just a product over all cluster–galaxy pairs of the GIK model’s probability density for each pair’s velocity,

$$\mathcal{L} = \prod_i P(v_i|r_i, \mathbf{g}), \quad (8)$$

where $P(v_i)$ is defined in equation (4) and depends on the set of GIK parameters (\mathbf{g}) and the separation of cluster–galaxy pair i (r_i). The

fact that we could define a likelihood that did not require either radial or velocity-space binning only occurred to us late in this project, and so the work throughout this paper uses the likelihood from equation (5). Nevertheless, we mention the alternative in case it is useful for other people trying to implement a similar modelling method. We tested that the likelihoods in equations (5) and (8) produced consistent measurements of the GIK parameters for a fiducial INDRA simulation volume, finding that the differences in best-fitting parameters were negligible, with the no-binning likelihood leading to slightly tighter constraints on some GIK parameters, presumably due to information loss when binning. This test is described further in Appendix B.

3.3 Convergence of the GIK parameters

As described later in this paper, we will ultimately use N -body simulations run with different cosmologies to predict how the GIK parameters depend upon cosmology. It is therefore important to understand how robust the GIK parameters measured from an N -body simulation are. To address this, we use publicly available simulations from the IllustrisTNG project (Nelson et al. 2019). In particular, we use variants of the TNG300 simulations, to assess how robust the GIK parameters are to changing simulation resolution, as well as whether the GIK parameters change between DMO simulations and simulations including baryonic physics.

For this section, we use the TNG300-1 simulation (which we call ‘TNG300’) as our fiducial simulation. This simulation includes both dark matter and baryons, with dark matter particle masses of $5.9 \times$

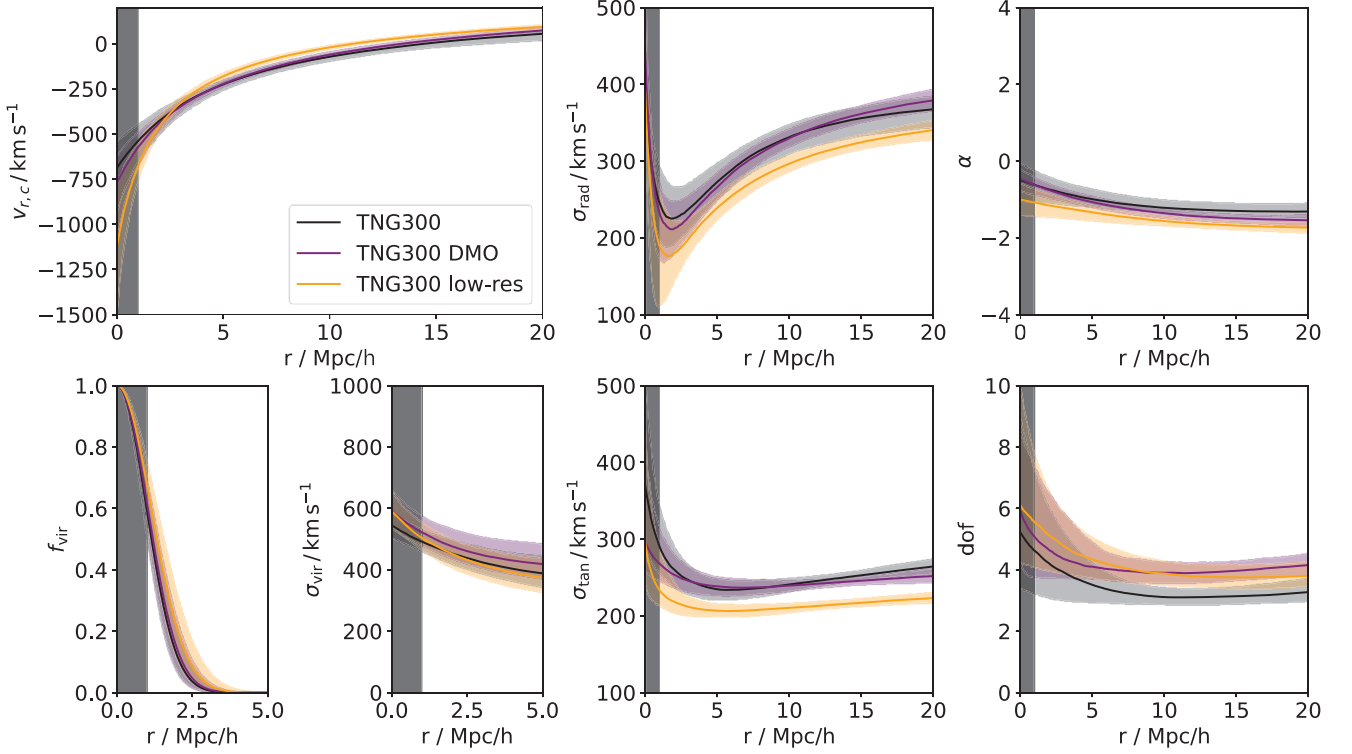


Figure 4. The GIK functions measured from three different TNG300 simulations. The inclusion of baryonic physics or not seems to have little impact on the GIK functions, while the lower resolution simulation has slight differences, including a suppression of the radial and tangential velocity dispersion of the infalling component.

$10^7 M_{\odot}$ and initial gas masses of $1.1 \times 10^7 M_{\odot}$. The TNG300-3 simulation (which we call ‘TNG300 low-res’) employs the same physics, but with 64 times fewer particles than TNG300-1, leading to DM masses of $3.8 \times 10^9 M_{\odot}$ and gas masses of $7.0 \times 10^8 M_{\odot}$. Finally, TNG300-1-Dark (which we call ‘TNG300 DMO’) is a DMO version of TNG300-1, with a dark matter particle mass of $7.0 \times 10^7 M_{\odot}$.

For each of these simulations we used the method described in Section 3.2.1 to fit for the GIK parameters. We ran an MCMC to generate samples from the posterior distribution for the GIK parameters. By calculating the corresponding GIK functions for each MCMC sample, we generate draws of the GIK functions from this posterior. These are plotted in Fig. 4, where the solid lines show the median values of the GIK functions, while the shaded regions cover the 16th–84th percentiles of the GIK function posteriors.

In Fig. 4, the differences between the fiducial and DMO simulations are negligible. The lower resolution simulation does appear to have somewhat different GIK functions, although the differences (except for σ_{tan} at large radii) are typically less than the uncertainty, where the size of the uncertainty comes from the relatively small box size of the TNG300 simulations.

In Fig. 5, we plot the real-space clustering from these three different TNG300 simulations. As described in detail in Section 4, the combination of the real-space clustering and the GIK functions (or some other description of the pairwise velocity distribution) are all that is required to calculate ξ_{cg}^s . We see that, as was the case for the GIK functions, TNG300 and TNG300 DMO produce similar results, with TNG300 low-res being more discrepant.

We note that the GIK parameter differences between DMO versus hydro simulations, or high-resolution versus low-resolution,

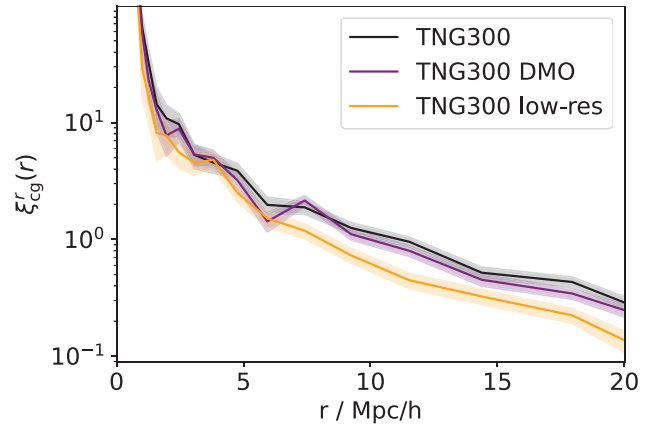


Figure 5. The real-space clustering measured from three different TNG300 simulations. The inclusion of baryonic physics has little impact, while the lower resolution simulation has suppressed clustering at large separations. The correlation function was measured by counting cluster–galaxy pairs in bins of separation and comparing with the expectation if clusters and galaxies were randomly distributed. The shaded region is an indication of the uncertainty on the correlation function measurement, from assuming that the pair counts follow Poisson statistics.

are significantly less important than the differences in real-space clustering between those same simulations for evaluating a model ξ_{cg}^s . This is quantified in Appendix C, and suggests that relying on simulations to predict the GIK parameters – while using some other method for the real-space clustering – is a sensible approach. This is what we do throughout the rest of the paper, where (because we

are dealing with simulations) we know the real-space clustering and so use the true real-space clustering in our models for ξ_{cg}^s . With real observational data, we would instead need a model for the real-space clustering, or to obtain it from observations of the projected clustering. These possibilities are discussed in Section 7.1.2.

4 CALCULATING ξ_{cg}^s WITH THE GIK MODEL

In the previous section, we described the GIK model from ZW13, and presented a method to fit the GIK parameters to the positions and velocities of clusters and galaxies extracted from a simulation. We will now show how, with the GIK parameters determined, we can predict the clustering in redshift-space given the real-space clustering.

We begin with the expression for the redshift-space clustering in terms of the real-space clustering and the probability distribution for $r_\pi - y$ (i.e. for how a cluster–galaxy pair’s separation is shifted when moving from real to redshift space). This usually goes by the name of the streaming model, and can be written as (Peebles 1980; Scoccimarro 2004; Kuruvilla & Porciani 2018)

$$1 + \xi_{\text{cg}}^s(r_p, r_\pi) = \int_{-\infty}^{\infty} [1 + \xi_{\text{cg}}^r(r)] p(r_\pi - y | r_p, y) dy, \quad (9)$$

where $r = \sqrt{r_p^2 + y^2}$, and $p(r_\pi - y | r_p, y) = p(v_y | r_p, y) H(z) / (1 + z)$ is the probability density associated with a given pairwise velocity along the line of sight, where the factor of $H(z)/(1 + z)$ comes from dr_π/dv_y .

The distribution of radial and tangential velocities at a given separation, $p(v_r, v_t | r)$, is what is specified by the GIK model. The conversion from (v_r, v_t) to v_y depends on the angle between the line of sight and the cluster–galaxy separation, and so depends on r_p and y . Mathematically, we have $v_y = v_r \sin \theta + v_t \cos \theta$, where $\theta = \tan^{-1} y/r_p$ is the angle between the plane-of-the-sky and the cluster–galaxy separation. This leads to $p(v_y | r_p, y)$ being expressible as an integral:

$$p(v_y | r_p, y) = \int_{-\infty}^{\infty} p\left(v_r, v_t = \frac{v_y - v_r \sin \theta}{\cos \theta} \mid r\right) \frac{dv_r}{\cos \theta}. \quad (10)$$

Combining equations (9) and (10), and integrating only out to finite limits, we have

$$1 + \xi_{\text{cg}}^s(r_p, r_\pi) \approx \frac{H(z)}{1 + z} \times \int_{y_{\min}}^{y_{\max}} \frac{1 + \xi_{\text{cg}}^r(r)}{\cos \theta} \int_{v_{\min}}^{v_{\max}} p(v_r, v_t | r) dv_r dy, \quad (11)$$

where again $v_t = (v_y - v_r \sin \theta) / \cos \theta$ and $v_y = H(z)(r_\pi - y) / (1 + z)$. In order for equation (11) to be exact, the integration limits should cover an infinite range of y and v_r . However, for the purpose of calculating $\xi_{\text{cg}}^s(r_p, r_\pi)$ for $r_p, r_\pi < 20 h^{-1}$ Mpc (which we do here) we found that $y_{\max} = -y_{\min} = 40 h^{-1}$ Mpc and $v_{\max} = -v_{\min} = 2500 \text{ km s}^{-1}$ were sufficient. We note that the reason for using finite limits was that to speed up the calculation, we did not use a generic numerical integration routine (which could potentially handle infinite limits), instead using fixed v_r, v_t , and y grids and evaluating equation (11) for a large number of r_π (but fixed value of r_p) simultaneously, by summing up the integrand evaluated over the grids. This process is much faster because multiple evaluations of $p(v_r, v_t | r)$ with a particular r can be made simultaneously.⁵ We use

⁵Due to how the pdf method of the `scipy.stats.multivariate_t` class is written, it is much faster to evaluate $p(v_r, v_t | r)$ with thousands of

1000 equally spaced v bins (with 5 km s^{-1} spacing) and an adaptive number of y bins, that we double until successive iterations agree on all values of $\xi_{\text{cg}}^s(r_p, r_\pi)$ to within a specified tolerance, which by default we set to be a fractional agreement of better than 1 per cent.

5 BUILDING AN EMULATOR FOR THE GIK PARAMETERS

One might hope that the GIK functions and their dependence on cosmological parameters can be predicted from theory (i.e. ‘with pen and paper’). However, even predicting the mean infall velocity onto clusters (closely related to $v_{r,c}$) from first principles is not easy, with simple spherical infall models failing to correctly predict the results of N -body simulations for the mean infall velocity of dark matter particles as a function of their distance from the centres of clusters (e.g. fig. 3 of Villumsen & Davis 1986). However, the fact that the GIK functions at fixed cosmology seem not to be too sensitive to the physics employed within the simulation, or to the numerical resolution at which the simulation was run (see Fig. 4), suggests that the GIK functions can be inferred from N -body simulations in a reasonably reliable manner.

This motivates using suites of simulations (run with different cosmological parameters) to build an emulator that can predict how the GIK parameters depend upon cosmology. Combining such an emulator with knowledge of the real-space clustering of galaxies and clusters, $\xi_{\text{cg}}^r(r)$, allows us to predict the redshift-space clustering as a function of cosmology, and therefore to fit the cosmological parameters to an observed ξ_{cg}^s data vector.

5.1 Comparison with directly emulating the ξ_{cg}^s data vector

There have been a number of recent attempts to model redshift-space galaxy clustering on small scales using emulators trained on N -body simulations (Zhai et al. 2019; Yuan et al. 2022; Lange et al. 2023; Kwan et al. 2023). Aside from the fact that these have focused on galaxy–galaxy rather than cluster–galaxy correlations, our approach is also conceptually quite different. In particular, these other works directly measure observables (e.g. ξ_{gg}^s on a grid of r_p, r_π , or multipoles of ξ_{gg}^s) from simulations, and then build emulators for these observables as a function of the cosmological parameters,⁶ while we break up the calculation into two distinct parts (obtaining the real-space clustering, and then modelling the RSD effects on this), with an emulator used to do the second part.⁷

5.1.1 Reducing noise with the GIK model

One considerable benefit of our approach is that assuming a functional form for the pairwise velocity distribution of cluster–galaxy pairs acts to reduce the effects of noise in each simulation used to build the emulator. This enables the use of smaller simulations than

different (v_r, v_t) combinations simultaneously, than to make thousands of separate calls to $p(v_r, v_t | r)$.

⁶For brevity, we refer only to the cosmological parameters, but the emulators also include parameters for the so-called *galaxy–halo connection* (Wechsler & Tinker 2018), which describes how galaxies populate dark matter haloes.

⁷We note that this approach is similar to that described in Cuesta-Lazaro et al. (2023), where they present an emulator for real-space clustering, that they intend to combine with an emulator for a parametrized velocity distribution similar to the ‘infalling’ part of the GIK model (equation 2), described in Cuesta-Lazaro et al. (2020).

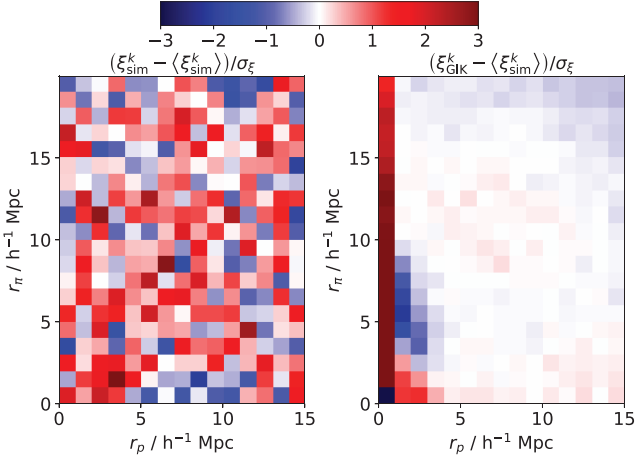


Figure 6. Maps of the normalized residuals, comparing ξ_{cg}^s from one example INDRA simulation, to the average over all INDRA simulations. In the left-hand panel, ξ_{cg}^s from the single simulation is calculated directly from the cluster and galaxy positions in redshift space in the simulation, while in the right-hand panel, the GIK parameters are fit to the simulation data, which is then combined with the real-space clustering to produce ξ_{cg}^s . This second procedure dramatically decreases the amount of noise on ξ_{cg}^s measured from a single simulation, though model mis-specification leads to systematic residuals at low r_p .

would be required if one simply ‘observes’ each simulation in a similar manner to how the real observations will be made.

To demonstrate the benefit of this quantitatively, we begin by defining $\xi_{\text{sim}}^{s,k}$, the ξ_{cg}^s data vector calculated directly from the k th INDRA simulation (with an example plotted in the right-hand panel of Fig. 1). We additionally define the data vector calculated by combining the real-space clustering from the k th INDRA simulation, $\xi^{r,k}(r)$, with the maximum-likelihood GIK parameters from the k th INDRA simulation, as $\xi_{\text{GIK}}^{s,k}$. We also define the INDRA-derived covariance matrix for the elements of ξ_{cg}^s as C_{sim} . This is calculated from the set of $\xi_{\text{sim}}^{s,k}$ (with some Jackknifing, described in Section 6.1). The inverse of C_{sim} is the precision matrix, $\Psi_{\text{sim}} = C_{\text{sim}}^{-1}$, with the elements of Ψ_{sim} that relate to the first 20 elements of our flattened data vector (which is all those with $r_p = 0.5 h^{-1}$ Mpc) set to zero. This means that we give no weight to the lowest r_p column of the ξ_{cg}^s maps, because the parametric GIK functions do not agree with the measured GIK functions on scales below $1 h^{-1}$ Mpc (see Fig. 3). In addition, we define the values along the diagonal of C_{sim} to be σ_{ξ}^2 .

With these definitions, we can assess how precisely we can use a single INDRA simulation to estimate the true ξ_{cg}^s for the INDRA cosmology, which we assume is well approximated by $\langle \xi_{\text{sim}}^{s,k} \rangle$, where the expectation value is an average over all of the INDRA simulations. In particular, we define

$$\begin{aligned} \chi_{\text{sim}}^{2,k} &= (\xi_{\text{sim}}^{s,k} - \langle \xi_{\text{sim}}^{s,k} \rangle)^{\top} \Psi_{\text{sim}} (\xi_{\text{sim}}^{s,k} - \langle \xi_{\text{sim}}^{s,k} \rangle) \\ \chi_{\text{GIK}}^{2,k} &= (\xi_{\text{GIK}}^{s,k} - \langle \xi_{\text{sim}}^{s,k} \rangle)^{\top} \Psi_{\text{sim}} (\xi_{\text{GIK}}^{s,k} - \langle \xi_{\text{sim}}^{s,k} \rangle). \end{aligned} \quad (12)$$

Calculating these χ^2 values for each INDRA simulation, we find that $\langle \chi_{\text{sim}}^{2,k} \rangle = 301$, while $\langle \chi_{\text{GIK}}^{2,k} \rangle = 27$. This means that employing the GIK model, we can calculate a model ξ_{cg}^s data vector from a single INDRA simulation that is much closer (by a factor of ~ 10 in χ^2) to the average ξ_{cg}^s directly measured from all INDRA simulations, than ξ_{cg}^s directly measured from any one INDRA simulation is to the INDRA average.

For some intuition as to how this works, in Fig. 6 we plot the

normalized residuals for one example INDRA simulation, which we define as the difference between a $\xi_{\text{cg}}^{s,k}$ predicted from one simulation and the average $\xi_{\text{sim}}^{s,k}$ over all INDRA simulations, divided by the simulation-to-simulations noise (σ_{ξ}). In the left-hand panel, we do this for a case where the one-simulation prediction is measured directly from the simulation ($\xi_{\text{sim}}^{s,k}$), while in the right-hand panel, the one-simulation prediction uses the GIK model ($\xi_{\text{GIK}}^{s,k}$). As expected, the map on the left consists of many fluctuations of order $1\sigma_{\xi}$, while employing the GIK model smooths over this noise in a physically motivated way.

5.1.2 Model mis-specification error

The main drawback of our approach is also highlighted in Fig. 6, and is the possibility for model mis-specification error. In particular, if the GIK model does not adequately describe the pairwise velocity distribution of cluster–galaxy pairs, then using this model to map from real- to redshift-space clustering can introduce systematic errors. The residuals seen at $r_p \lesssim 3 h^{-1}$ Mpc in the right-hand panel of Fig. 6 demonstrate some deficiency of the GIK model, and could arise for two primary reasons. One is if the seven GIK functions do not adequately describe $P(v_r, v_t|r)$ at fixed radius, while the other is if the 24 GIK parameters do not adequately describe the radial dependence of the GIK functions.

We have already observed that with a sufficiently large simulated volume, deviations between $P(v_r, v_t|r)$ from simulations, and the seven-parameter velocity distribution function at the heart of the GIK model (equation 4) can be detected, as demonstrated in the residuals panel of Fig. 2. In Appendix D, we investigate the relative importance of these two effects, finding that the GIK parameters’ inability to perfectly describe the GIK functions explains about half (in a χ^2 sense) of the residuals, with the remaining half coming from the seven-parameter model for $P(v_r, v_t|r)$.

Realistic surveys are likely to have significantly greater uncertainties on these small scales than implied by σ_{ξ} , because spectroscopic surveys face significant observational challenges when dealing with close pairs of galaxies (e.g. Bianchi & Percival 2017). This suggests that the model mis-specification errors that appear highly salient in Fig. 6 may not matter so much in a real survey. Also, the model mis-specification errors are confined to a restricted region of the r_p – r_{π} plane, so an analysis using the GIK model could cut out the unreliable regions without much difficulty. Depending on the accuracy requirements for the model ξ_{cg}^s when applied to real observational data, it may be that alternative functional forms that can better describe the velocity distribution are required, which we leave to future work.

5.1.3 Separating out real-space clustering from RSD effects

Our approach of building a model for the mapping from real- to redshift-space clustering and then combining this with some model/measurement of the real-space clustering to get ξ_{cg}^s is conceptually appealing, because physically the real-space clustering and the RSD are separate processes. This makes our model more interpretable than an emulator that directly predicts ξ_{cg}^s , and aids when trying to understand intuitively the effects of different cosmological parameters/models on ξ_{cg}^s .

Additionally, there are observables that depend on the real-space clustering (such as cosmic shear, Kilbinger 2015), as opposed to the redshift-space clustering. Our approach means that when doing a joint analysis of different cosmological observables, consistent

real-space clustering can be used in the modelling of the different observables, whereas when directly emulating ξ_{cg}^s , the underlying real-space clustering may be inconsistent with that used elsewhere in the analysis.

5.1.4 Simpler cosmological parameter dependence of emulated quantities

While we do not attempt to quantify it here, another potential advantage of emulating the GIK functions (rather than ξ_{cg}^s directly) is that they are simpler physical quantities than ξ_{cg}^s in a particular (r_p, r_π) pixel. This means that one would expect the GIK parameters describing the GIK functions to vary in a more systematic way with variations to the cosmological parameters than the redshift-space correlation function does. For example, if increasing some cosmological parameter leads to an increase in the infall velocities, $|v_{r,c}|$, this would lead to systematic shifts in the relevant GIK parameters. But in terms of values of ξ_{cg}^s at particular (r_p, r_π) , this could behave in a non-monotonic manner, with (e.g.) ‘Kaiser squashing’ enhancing the clustering, before ‘fingers of god’ decrease it. While Gaussian processes are flexible enough to model more complex functions, emulation of simpler functions can be done with less thought given to the Gaussian process kernels, and should require fewer training data locations to produce a robust emulator.

5.2 The FORGE simulations

One of the major goals of modern cosmology is to understand what drives the accelerated expansion of the Universe at late times (Weinberg et al. 2013), with modified gravity theories being a popular alternative to GR plus a cosmological constant. $f(R)$ gravity (first proposed in Buchdahl 1970) is one popular modified gravity theory, which we investigate here as an example of how one would use the GIK model to constrain the parameters of a particular theory of gravity. We note that of the many modified gravity theories that have been proposed, we do not hold $f(R)$ in any special regard, but choose to focus on $f(R)$ in the first instance due to the fact that the GIK parameters have already been shown to differ in $f(R)$ and GR (Zu et al. 2014), and due to the existence of FORGE (Arnold et al. 2022), a suite of cosmological simulations run with different $f(R)$ models.

The FORGE simulations were run with 50 different combinations of cosmological parameters in Hu–Sawicki $f(R)$ gravity (Hu & Sawicki 2007), including various values for $\bar{f}_{\text{R}0}$,⁸ as well as Ω_m , σ_8 , and h . For each simulated cosmology, there are both high- and low-resolution simulations. We only use the high-resolution simulations, which are of cubic volumes with a comoving side length of $500 h^{-1}$ Mpc. The particle mass used in these simulations is $9.1 \times 10^9 h^{-1} M_\odot$, with a gravitational softening length of $15 h^{-1}$ kpc. We do not use the large volume, low-resolution, simulations because the comparatively poor mass resolution ($m_{\text{DM}} = 1.5 \times 10^{12} h^{-1} M_\odot$) means that the (sub-)structures capable of hosting galaxies of interest to us are not all resolved. As such, we would need a different method of populating our simulations with galaxies in order to calculate the GIK parameters.⁹

⁸ $\bar{f}_{\text{R}0}$ is the background value of the scalar degree of freedom at $z = 0$ and controls the potential depth threshold at which the chameleon screening becomes active and GR-like forces are recovered. Larger values of $|\bar{f}_{\text{R}0}|$ correspond to larger deviations away from GR.

⁹ These larger volume simulations could, for example, be used in conjunction with a halo occupation distribution (HOD) model (e.g. Peacock & Smith

The FORGE cosmologies are primarily arranged in a latin hypercube, with 49 of the 50 simulations uniformly covering the following parameter ranges:

- (i) Ω_m from 0.11 to 0.55;
- (ii) $S_8 \equiv \sigma_8 \times (\Omega_m/0.3)^{0.5}$ from 0.60 to 0.90;
- (iii) h from 0.60 to 0.82;
- (iv) $\log_{10} |\bar{f}_{\text{R}0}|$ from -6.2 to -4.5 ;

the remaining simulation is a Λ CDM one, that is, it has $|\bar{f}_{\text{R}0}| = 0$. Because the majority of the FORGE simulations are evenly distributed in $\log_{10} |\bar{f}_{\text{R}0}|$ rather than $|\bar{f}_{\text{R}0}|$, it makes sense to treat $\log_{10} |\bar{f}_{\text{R}0}|$ as the $f(R)$ parameter in our emulator. However, this then leads to the value for the Λ CDM simulation being undefined. To avoid this, we instead use $\log_{10} (|\bar{f}_{\text{R}0}| + 10^{-6})$ as the $f(R)$ parameter in our emulator.

For each of the 50 FORGE simulations, we calculate a posterior distribution for the GIK parameters following the method described in Section 3.2.1. The priors on the GIK parameters are given in Appendix A. We continue to define galaxies as haloes/subhaloes with $v_{\text{max}} > 250 \text{ km s}^{-1}$, and we evaluate the GIK parameters for seven different cluster mass bins for each cosmology. Each mass bin spans 0.1 dex in mass, covering the range $13.8 < \log_{10} M_{200}/h^{-1} M_\odot < 14.5$. For each cluster mass bin, we consider the cluster mass associated with that bin to be the geometric mean of the mass of the upper and lower edges of the bin, such that the bin containing haloes with $14.1 < \log_{10} M_{200}/h^{-1} M_\odot < 14.2$ is considered to correspond to clusters of mass $10^{14.15} h^{-1} M_\odot$.

5.3 Emulation with Gaussian processes

The aim of our emulator is to predict the cluster–galaxy pairwise velocity distribution (and its dependence on cluster–galaxy separation) as a function of the cosmological parameters plus the cluster mass (going forward we will refer to cluster mass as an additional ‘cosmological parameter’ for brevity, such that we have 350 different simulated ‘cosmologies’). Within the GIK model, the velocity distribution at all radii is specified by the 24 GIK parameters, and so we emulate the velocity distribution by separately emulating each of the 24 GIK parameters.

For the purpose of emulating each GIK parameter, g_i , we use a Gaussian process, using the PYTHON package GEORGE (Ambikasaran et al. 2015). A Gaussian process is a prior over possible functions, which, when conditioned on a set of (potentially uncertain) observations of the function (which we will call the training data), can produce a posterior distribution for the target function. In our case, the functions for which we would like to infer the posteriors are the $g_i(\theta)$, with i (ranging from 1 to 24) being the index for a particular GIK parameter, and θ being a vector of cosmological parameters:

$$\theta = \{ \Omega_m, S_8, h, \log_{10} (|\bar{f}_{\text{R}0}| + 10^{-6}), \log_{10} M_{200}/h^{-1} M_\odot \}. \quad (13)$$

A detailed description of Gaussian processes can be found in Rasmussen & Williams (2006).¹⁰ In general, when combined with training data (in our case a mean and Gaussian uncertainty on each

2000), although doing this in the context of extracting the GIK parameters – particularly those important for the velocity distribution at small cluster–galaxy separations – would be somewhat circular, as HOD galaxies are typically assigned velocities drawn from simple Gaussian velocity distributions (e.g. Guo et al. 2015).

¹⁰ For intuition on how and why Gaussian processes work, we recommend an excellent lecture by Richard Turner, available on [YouTube](https://www.youtube.com/watch?v=...).

of the $g_i(\theta_j)$, where the set of θ_j are the 350 input cosmologies at which we have fit for the GIK parameters) a Gaussian process can make a prediction for how the output of a function depends on the input parameters [i.e. $g_i(\theta)$ for arbitrary θ].

Specifying a Gaussian process requires that we specify an a priori most-likely mean function, $m(\theta)$, as well as a covariance function, often referred to as the kernel. This kernel, $K(\theta_1, \theta_2)$, describes how correlated we expect $g_i(\theta_1)$ and $g_i(\theta_2)$ to be, as a function of the separation between θ_1 and θ_2 . The kernel is used to specify the notion that we expect $g_i(\theta)$ to vary smoothly with the cosmological parameters, such that two similar values of θ will have similar values for $g_i(\theta)$. We use a squared-exponential kernel, which can be written as

$$K(\theta_1, \theta_2) = \sigma_K^2 \exp\left(-\frac{1}{2}(\theta_1 - \theta_2)^\top \Sigma_K^{-1}(\theta_1 - \theta_2)\right). \quad (14)$$

We assume that Σ_K is a diagonal matrix, with elements

$$\Sigma_K = \text{diag}\left(l_{\Omega_m}^2, l_{S_8}^2, l_h^2, l_{f_{\text{R0}}}^2, l_{M_{200}}^2\right), \quad (15)$$

where (e.g.) l_{S_8} is a length-scale in terms of the parameter S_8 . The intuition one should have is that our prior will substantially downweight functions where g_i fluctuates as S_8 is varied, if the length-scale of the fluctuations is shorter than l_{S_8} . The value of σ_K determines the prior expectation on the amplitude of variations in g_i as one varies θ .

We build a separate Gaussian process to emulate each of the 24 GIK parameters. For a single GIK parameter, g_i , the inputs to our Gaussian process are:

- (i) The estimates of g_i evaluated for each of our seven cluster mass bins, at each of the 50 FORGE cosmologies ($\hat{g}_i(\theta_j)$), for which we use the mean of the marginalized posterior on this particular GIK parameter.
- (ii) The Gaussian uncertainty on each $\hat{g}_i(\theta_j)$, which we calculate as the square root of the variance of the relevant marginalized posterior.
- (iii) A kernel function (equation 14), with its associated hyperparameters.
- (iv) A mean function, which we assume to be a constant (i.e. independent of θ), and which we set equal to $\langle g_i(\theta_j) \rangle_j$, the mean value of g_i over the 350 ‘cosmologies’ where the GIK parameters have been fit.

5.4 Selecting kernel hyperparameters

For each of our GIK parameters, we have six kernel hyperparameters (σ_K in equation 14, and the five length-scales in equation 15). In order to set these hyperparameters in an objective manner, we find (separately for each i) the set of hyperparameters that maximize the marginal likelihood of the input data (see equation 5.8 of Rasmussen & Williams 2006). Considering the combination of our kernel function (with a specific set of hyperparameters) and mean function as defining a prior on the function $g_i(\theta)$, the marginal likelihood is the probability density associated with having drawn our input data from this prior, and can be calculated by GEORGE.

We note that some work in the literature alternatively determines the optimal hyperparameters using leave one out tests (called ‘cross-validation’ in Rasmussen & Williams 2006). This involves building one Gaussian process per piece of training data, where each Gaussian process is trained on all but one, left out, piece of training data. Each Gaussian process is then used to predict the value of the relevant left out training data. This can be done for a variety of different

kernel hyperparameters, with the predicted and true values of the left out data compared, and the hyperparameters that lead to the most accurate predictions for the left out data chosen. We implemented such a method, and found best-fitting hyperparameters that were similar to those from maximizing the marginal likelihood. As this method is slow (it requires building a separate Gaussian process for each piece of training data) we used the marginal-likelihood approach for this work.

6 FITTING THE EMULATOR MODEL TO MOCK ξ_{cg}^s DATA

Combining an emulator for the GIK parameters as a function of the cosmological parameters, with knowledge of the real-space cluster–galaxy correlation function, we are able to predict ξ_{cg}^s as a function of cosmology. By comparing these model predictions with some observed ξ_{cg}^s data vector, we can then constrain the cosmological parameters. In order to make this comparison, we need to define a likelihood of having obtained a particular observed ξ_{cg}^s , given a model ξ_{cg}^s . We assume this likelihood is Gaussian, with

$$\mathcal{L}(\theta) \propto \exp\left(-\frac{1}{2}\left(\xi_{\text{cg}}^{s,d} - \xi_{\text{cg}}^s(\theta)\right)^\top \mathbf{C}^{-1}\left(\xi_{\text{cg}}^{s,d} - \xi_{\text{cg}}^s(\theta)\right)\right), \quad (16)$$

where $\xi_{\text{cg}}^{s,d}$ is the observed data vector, $\xi_{\text{cg}}^s(\theta)$ is the model-predicted data vector for the cosmological parameter vector θ , and \mathbf{C} is the data covariance matrix.

6.1 Covariance matrix for ξ_{cg}^s

Even if adopting the correct cosmological parameters, with a perfect cosmological model, we would not expect the model data vector to match the observed one. Instead, we would expect the observed data vector to look like a random draw from a multivariate Gaussian distribution that has a mean value equal to the model data vector and some covariance matrix.¹¹ This covariance matrix describes the amount of noise on each element of the observed data vector, as well as correlations between the noise on different elements of the data vector. If we could generate an infinite number of independent realizations of a Gaussian distributed data vector, \mathbf{d} , then the covariance matrix for \mathbf{d} would be $C_{ij} \equiv \langle (d_i - \langle d_i \rangle)(d_j - \langle d_j \rangle) \rangle$, where $\langle \mathbf{d} \rangle$ is the mean of the data vectors, and d_i is the i th element of \mathbf{d} .

To estimate a covariance matrix suitable for our purposes, we use the INDRA simulations that were described in Section 2.1. The data vector we use when fitting for the cosmological parameters is $\xi_{\text{cg}}^s(r_p, r_\pi)$ on a 15×20 grid of (r_p, r_π) , with $(1 h^{-1} \text{ Mpc})^2$ pixels, meaning that we have a data vector with $n_d = 300$ elements. For the purposes of discussing the covariance matrix, it is simpler to consider our data vector to be 1D. In order to facilitate this, we flatten our 2D data vector, such that elements 1–20 run from small to large r_π with $r_p = 0.5 h^{-1} \text{ Mpc}$, elements 21–40 are the same at $r_p = 1.5 h^{-1} \text{ Mpc}$, etc. We label this flattened version of the data vector ξ , with ξ_i the i th element, with i running from 1 to 300.

In total there are $n_s = 384$ INDRA simulations, each a $1 h^{-1} \text{ Gpc}$ on-a-side box, all run with the same cosmological parameters. We can use each INDRA simulation to calculate a redshift-space cluster–galaxy correlation function data vector, ξ^k , with k indicating the index of the INDRA simulation. These can then be used to produce an

¹¹This is what is meant by assuming that our ‘likelihood is Gaussian’.

unbiased estimate for the data covariance matrix

$$\widehat{C}_{ij} = \frac{1}{n_s - 1} \sum_{k=1}^{n_s} (\xi_i^k - \langle \xi_i \rangle) (\xi_j^k - \langle \xi_j \rangle), \quad (17)$$

where the $n_s - 1$ factor in the denominator (as opposed to simply n_s) accounts for the fact that we have estimated the mean ξ from the data itself.

While \widehat{C} is an unbiased estimate of C , \widehat{C}^{-1} is not an unbiased estimate of the precision matrix $\Psi \equiv C^{-1}$. However, we can achieve an unbiased estimate of Ψ as (Hartlap, Simon & Schneider 2007)

$$\widehat{\Psi} = \frac{n_s - n_d - 2}{n_s - 1} \widehat{C}^{-1}. \quad (18)$$

Note that at a minimum, one requires $n_s > n_d + 2$ in order for \widehat{C} to be invertible. However, if n_s is only slightly larger than $n_d + 2$, while equation (18) gives an unbiased estimate of the precision matrix, the uncertainty on this estimate will be large (Taylor, Joachimi & Kitching 2013). One generally requires that n_s is considerably larger than n_d in order for Ψ to be precisely determined. In our case, we have a length-300 data vector, with only 384 simulations, which would lead to significant uncertainty on the covariance matrix.

Without access to a larger number of simulations, or analytic estimates for the covariance matrix, we can instead turn to resampling techniques. These techniques include the subsample, bootstrap, and Jackknife methods (Norberg et al. 2009). They are usually applied when fitting to an observed data vector, and involve using the data itself (split up into subregions) to estimate the data covariance. Similar techniques can also be used to decrease the number of independent simulations required to compute a covariance matrix to some prescribed level of accuracy. For example, Escoffier et al. (2016) performed Jackknife resampling over a set of independent simulations, finding that this reduced the requirement on the number of simulations (in their case) by a factor of seven.

We follow Escoffier et al. (2016) and calculate a covariance matrix from each of our individual simulations, using the deleted Jackknife scheme (Shao & Wu 1989).¹² We split each simulation into $N_s = 8$ subregions, by splitting the box in half along each of the three Cartesian axes. Each Jackknife configuration, c , then leaves out $N_d = 2$ of the N_s regions, and we evaluate the corresponding correlation function, ξ^c , by adopting the simulation z -axis as the line-of-sight direction, and using the cluster and galaxy definitions given in Section 2.1. Note that whether a particular cluster–galaxy pair was included in a Jackknife region was based upon the position of the cluster, such that clusters close to the boundary of a ‘deleted’ subregion, can appear in pairs with galaxies from the deleted region. The covariance matrix estimate from each simulation is

$$\widehat{C}_{ij}^{\text{1sim}} = \frac{N_s - N_d}{N_d N_{\text{JK}}} \sum_{c=1}^{N_{\text{JK}}} (\xi_i^c - \langle \xi_i \rangle) (\xi_j^c - \langle \xi_j \rangle), \quad (19)$$

where $N_{\text{JK}} = \binom{N_s}{N_d}$ is the number of possible Jackknife configurations.

Having evaluated a separate $\widehat{C}^{\text{1sim}}$ from each INDRA simulation, our estimate for the data covariance matrix is the average of all of these, and our estimate for the inverse data covariance matrix is just the inverse of this average (without any factor like that in equation 18, following Escoffier et al. 2016).

The covariance matrix evaluated in this way is appropriate for the $(1 h^{-1} \text{Gpc})^3$ volume of an INDRA simulation. Where necessary, we

¹²We note that a similar strategy was used in Hang et al. (2022).

assume that the covariance matrix scales inversely with volume (e.g. Howlett & Percival 2017), in order to calculate a covariance matrix appropriate for other volumes.

In Fig. 7, we plot the correlation matrix, which is closely related to the covariance matrix just described. The correlation matrix measures the Pearson correlation coefficient between all pairs of elements, and is defined as $\text{Corr}(\xi_i, \xi_j) \equiv \widehat{C}_{ij} / \sqrt{\widehat{C}_{ii} \widehat{C}_{jj}}$. In the right-hand panel of Fig. 7, one can see that neighbouring pixels of $\xi_{\text{cg}}^s(r_p, r_\pi)$ are quite correlated,¹³ suggesting that this particular data vector may not be the most efficient compression of the set of small-scale (r_p, r_π) separations for cluster–galaxy pairs. One could instead consider multipoles of the correlation function (e.g. Zhai et al. 2019), clustering wedges (e.g. Kazin, Sánchez & Blanton 2012), or different numbers of pixels and/or different spacing of bin edges (such as the logarithmically spaced r_p bin edges in Yuan et al. 2022). An advantage of these alternatives is that they generally lead to a smaller number of data vector elements, decreasing the requirements on the number of simulations required to evaluate the covariance matrix. However, we leave investigating these alternatives to future work.

In the left-hand panel of Fig. 7, we plot a map of the signal-to-noise ratio (SNR) from the ensemble of INDRA simulations. The signal in this case is the mean value of ξ_{cg}^s over the different INDR

A subvolumes, while the noise is the square root of the diagonal elements of \widehat{C} , which we label $\sigma_{\xi_{\text{cg}}^s}$. Both the signal and noise increase as one approaches the origin (i.e. $r_p = r_\pi = 0$), but the SNR peak is at low r_π , with $r_p \sim 4 h^{-1} \text{Mpc}$.

6.2 Including model uncertainty in the covariance matrix

The covariance matrix calculated from the INDRA simulations, which we here label C_{INDRA} , describes how large the variations typically are between ξ_{cg}^s calculated from any one particular INDRA simulation, and the average of all the INDRA simulations (which, given the large number of INDRA simulations, we can think of as being a good approximation to the ‘noise-free’ prediction for ξ_{cg}^s at the INDRA cosmology).

This means that if our model for $\xi_{\text{cg}}^s(\theta)$ could produce the ‘noise-free’ prediction for ξ_{cg}^s as a function of cosmological parameters, we could set C in equation (16) equal to $C_{\text{data}} = C_{\text{INDRA}} / (V_{\text{data}} / V_{\text{INDRA}})$, with V_{data} the effective volume of our survey, and $V_{\text{INDRA}} = (1 h^{-1} \text{Gpc})^3$. However, our emulator for the GIK parameters is built from simulations with a finite volume, which leads to uncertainty on the GIK parameters (which can be visualized, e.g. by the shaded regions around the GIK function lines in Fig. 3).

Recall that we have a separate Gaussian process emulator for predicting each element, g_i , of the GIK parameter vector, \mathbf{g} , as a function of θ , and that these predictions take the form of a Gaussian probability distribution, specified by a mean prediction $\langle g_i(\theta) \rangle$, and a standard deviation $\sigma_{g_i}(\theta)$. This can also be expressed as having a predicted mean GIK parameter vector, $\langle \mathbf{g}(\theta) \rangle$, and a covariance matrix for $\mathbf{g}(\theta)$, which we label Σ_g . For now Σ_g is diagonal (with the diagonal elements equal to the $\sigma_{g_i}^2(\theta)$), because we have built a separate emulator for each element of \mathbf{g} , and therefore cannot account for any covariance between the different elements of \mathbf{g} .

¹³Given the order in which we flatten $\xi_{\text{cg}}^s(r_p, r_\pi)$ to get ξ , neighbouring indices correspond to neighbouring pixels in the r_π direction, while indices separated by 20 correspond to neighbouring pixels in the r_p direction.

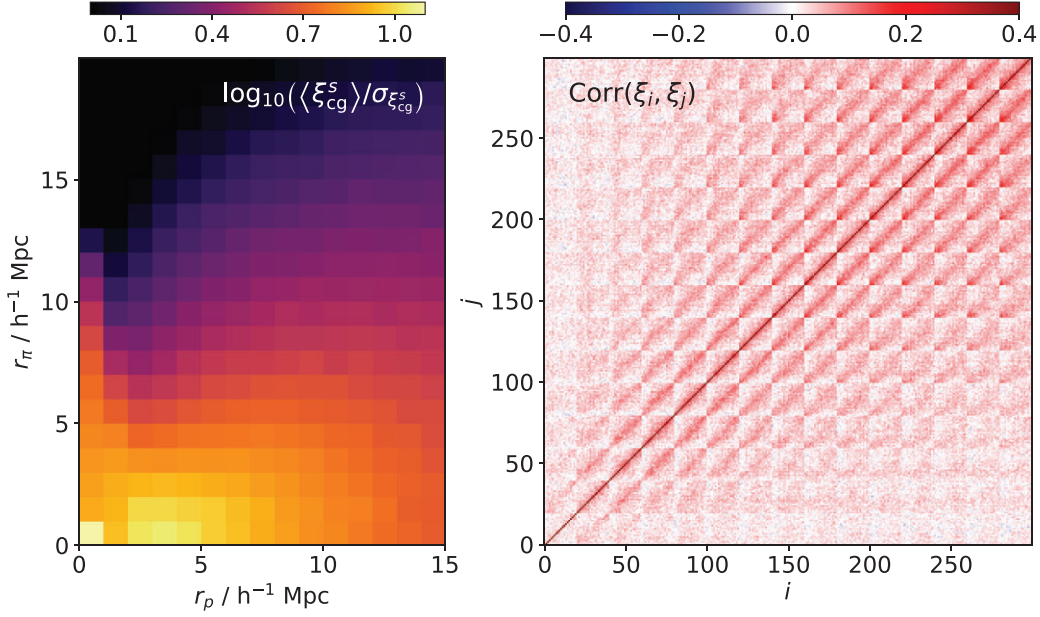


Figure 7. Left: a map of the SNR in the INDRAsigma_cg^s maps. This is calculated by dividing (pixel-by-pixel) the mean value of xi_cg^s from the various INDRAsigma_cg^s subvolumes by the standard deviation of xi_cg^s over the subvolumes. Note that the values of sigma_cg^s are equal to the diagonals of the covariance matrix, C_INDRAsigma_cg^s. Right: the correlation matrix, corresponding to the covariance matrix, C_INDRAsigma_cg^s. The values along the diagonal are (by definition) all equal to 1, but we use a smaller range for the colour map to highlight the correlations between many of the pixel values.

6.2.1 ‘Diagonalized’ GIK parameters

While there has been work on how to formulate multi-output Gaussian processes (e.g Wang & Chen 2015; Parra & Tobar 2017), we are not aware of any available Gaussian process software that can take vector-valued training data such as the values of $\mathbf{g}(\theta_j)$ for a set of θ_j , along with correlated uncertainties on each element of $\mathbf{g}(\theta_j)$, and then make predictions for $\mathbf{g}(\theta)$ along with $\Sigma_g(\theta)$. By building a separate emulator for each element of \mathbf{g} , we implicitly ignore the off-diagonal elements of the uncertainty on the measurements of the $\mathbf{g}(\theta_j)$, and force Σ_g to be diagonal.

Given that the posterior distributions on the $\mathbf{g}(\theta_j)$ do have quite considerable degeneracies between the different elements of \mathbf{g} , it would be better not to ignore them. We can circumvent the requirement for implementing a multi-output Gaussian process, by finding a linear transformation, $\mathbf{g}' = \mathbf{L}\mathbf{g}$, such that the posterior distribution for \mathbf{g}' is approximately a separable function of each element of \mathbf{g}' , which is to say that the posterior for each \mathbf{g}' in our set of training data has little covariance between the different elements.

We define the covariance matrix for a GIK parameter posterior distribution as

$$\mathbf{M} = \langle (\mathbf{g} - \langle \mathbf{g} \rangle)(\mathbf{g} - \langle \mathbf{g} \rangle)^T \rangle, \quad (20)$$

where the expectation values can be estimated from an MCMC chain of GIK parameter vectors. Then, defining \mathbf{V} as a matrix whose columns are the unit-length eigenvectors of \mathbf{M} , we have that $\mathbf{g}' = \mathbf{V}^T \mathbf{g}$.

The idea then, is that because each piece of training data (in terms of \mathbf{g}') has only minimal covariance between the different elements, that building a separate Gaussian process emulator for each element of \mathbf{g}' (rather than \mathbf{g}), and propagating the uncertainties on the training data \mathbf{g}' s through the Gaussian process, allows us to more faithfully include the covariance in our training data between the different elements of \mathbf{g} . The Gaussian processes then return an estimate of \mathbf{g}'

along with a diagonal covariance matrix, $\Sigma_{g'}$, and from this we can evaluate $\mathbf{g} = \mathbf{V}\mathbf{g}'$ and $\Sigma_g = \mathbf{V}\Sigma_{g'}\mathbf{V}^T$ (which can now have non-zero off-diagonal elements, representing the covariance between the different elements of \mathbf{g}).

6.2.2 Mapping GIK parameter uncertainties into uncertainties on xi_cg^s

In order to evaluate the likelihood for a particular θ (equation 16), we need to find the GIK model-predicted xi_cg^s(theta), which is done following the procedure described in Section 4. What we would then like to know is how uncertainty on $\mathbf{g}(\theta)$ (described by the covariance Σ_g), maps in to uncertainty on xi_cg^s(theta), such that this uncertainty on the model prediction for xi_cg^s(theta) can be accounted for in our covariance matrix. To do this, we consider small perturbations to the GIK parameter vector about some reference value, \mathbf{g}_0 . We assume that the correlation function in such a case can be well approximated by

$$\xi \approx \mathbf{A}(\mathbf{g} - \mathbf{g}_0) + \xi_0, \quad (21)$$

where ξ_0 is the data vector evaluated for the GIK parameter vector \mathbf{g}_0 . The matrix \mathbf{A} is the gradient of ξ with respect to \mathbf{g} , $A_{ij} = \partial \xi_i / \partial g_j$. This can be calculated using a finite-difference approach, by evaluating $\xi(\mathbf{g})$ over a grid of \mathbf{g} -points around \mathbf{g}_0 . The covariance matrix for ξ due to uncertainties on \mathbf{g} is then $\mathbf{C}_g = \mathbf{A}\Sigma_g\mathbf{A}^T$, where we evaluate \mathbf{A} at a fiducial cosmology (and treat it as independent of cosmology), while $\Sigma_g(\theta)$ is returned by our Gaussian processes, alongside $\mathbf{g}(\theta)$.

We note that not doing our diagonalization procedure would be a conservative choice. This is because the GIK parameter posteriors can have quite tight parameter degeneracies. If we consider two highly degenerate parameters, then their individual marginal distributions can be quite broad, while some combination of them is well constrained. If we ignore the fact that there is a well-constrained combination, and instead set their joint distribution equal to the

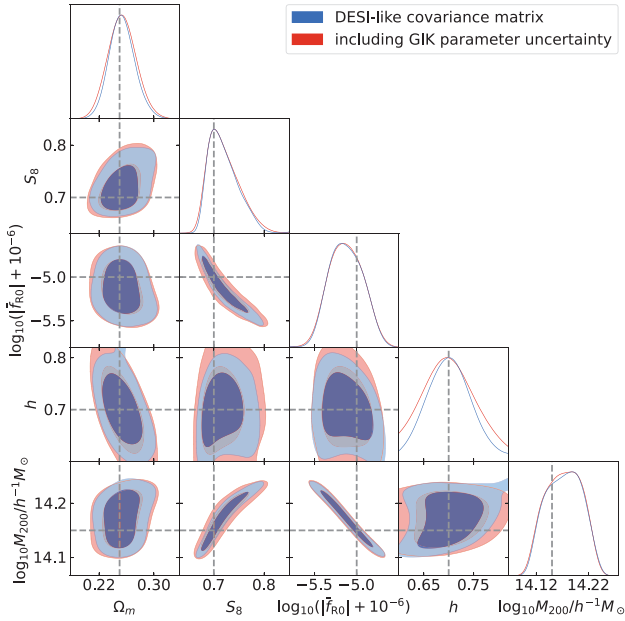


Figure 8. The posterior distribution for the cosmological parameters when fitting to a data vector generated by our emulator. The input cosmology is marked by the dashed lines, and the “DESI-like covariance matrix” contours contain 68 per cent and 95 per cent of the posterior distribution when using a covariance matrix (within our likelihood) that corresponds to the INDRA data covariance rescaled for an effective volume of $20 (h^{-1} \text{Gpc})^3$. The “including GIK parameter uncertainty” contours show the results when adding terms to the covariance matrix to account for the fact that we only know the GIK parameters as a function of cosmology to some finite level of precision, which is described in Section 6.2.

product of the two marginals, then this creates more freedom for the parameters to take different values, and therefore more uncertainty on the GIK functions, and a resulting increased model uncertainty on ξ_{cg}^s .

To demonstrate this quantitatively, we calculated a χ^2 between a fiducial ξ_{cg}^s (from $\mathbf{g}(\boldsymbol{\theta})$ at the mean FORGE cosmology), and the ξ_{cg}^s evaluated for the best-fitting GIK parameters for each of the 50 FORGE simulations. Doing this with and without ‘diagonalization’, we found that the mean χ^2 without diagonalization is ~ 70 per cent of that with diagonalization.

6.3 Fitting to mock data generated by the emulator

As a first demonstration of using our emulator to fit to a ξ_{cg}^s data vector, we fit to a data vector generated by the emulator itself. Such an exercise only partially tests our emulation procedure, but has the advantage that the data vector being fit to can be generated ‘noise free’, or with some other level of noise that we specify. This differs from a data vector constructed from an N -body simulation, which will necessarily have a level of noise set by the finite volume of the simulation.

In Fig. 8, we show the cosmological constraints from such an exercise. We build an emulator using the 50 FORGE simulations, and then use the emulator to predict a model data vector at a particular point in our cosmological parameter space (marked by the vertical and horizontal dashed lines in Fig. 8). We then use MCMC to sample from the posterior distribution for $\boldsymbol{\theta}$, with flat priors on each element of $\boldsymbol{\theta}$ (see equation 13) covering the range of $\boldsymbol{\theta}$ over which FORGE simulations were run, and using a data covariance matrix that is

equal to that calculated for $(1 h^{-1} \text{Gpc})^3$ from the INDRA simulations, but scaled down by a factor of 20, such that it is approximately appropriate for the effective volume of a survey such as DESI (DESI Collaboration 2016), which we take to have an effective volume of $20 (h^{-1} \text{Gpc})^3$ following Grove et al. (2022).

Fig. 8 reveals some interesting degeneracies between the different cosmological parameters. Perhaps the most striking is the degeneracy between the cluster mass associated with our cluster sample and \bar{f}_{R0} . In particular, a large cluster mass with a low value for $\log |\bar{f}_{\text{R0}}|$, produces a similar ξ_{cg}^s to a smaller cluster mass with a higher value for $\log |\bar{f}_{\text{R0}}|$. Given that we expect the gravitational attraction of more massive clusters to be stronger, and that a larger $\log |\bar{f}_{\text{R0}}|$ means an increased enhancement in the gravitational forces, this degeneracy makes intuitive sense, in that an enhancement to gravity can make clusters appear more massive than they actually are. It also suggests that combining the modelling of ξ_{cg}^s with some independent measurement of cluster masses will be a fruitful avenue for constraining the nature of gravity. For example, weak lensing can measure the mass of clusters in a manner that is unaffected by \bar{f}_{R0} (because the modifications to gravity do not affect the propagation of photons in $f(R)$ gravity). Such a measurement would break the $M_{200} - \bar{f}_{\text{R0}}$ degeneracy seen in Fig. 8, leading to an improved constraint on \bar{f}_{R0} .

Another feature of Fig. 8 is that we can see the impact that including the GIK parameter uncertainty into the covariance matrix (following Section 6.2) has on the cosmological constraints. In particular, while not dominant over the data covariance, the constraints are broadened somewhat when accounting for the GIK parameter uncertainty. This means that for a DESI-like survey one would ideally want larger volume simulations than the FORGE simulations we used here, in order for this model uncertainty to be negligible for the derived constraints. We stress that the ability for $(500 h^{-1} \text{Mpc})^3$ volume simulations to be used in the modelling of a $20 (h^{-1} \text{Gpc})^3$ data vector, without the model uncertainty being dominant over the data covariance, is possible only because of employing the GIK model (as opposed to directly emulating the data vector extracted from the set of simulations) and the resulting advantages described in Section 5.1.

6.4 Fitting to mock data extracted from simulations

As described above, while fitting to emulator-generated data provides some tests of our analysis pipeline, and also acts as a forecast for the constraining power expected with a particular experiment (which enters through the covariance matrix), it does not provide an exhaustive test of our modelling procedure. For example, we have seen in Fig. 2 that the functional form that we use for $P(v_r, v_t|r)$ does not perfectly describe the galaxy–cluster pairwise velocity distribution. To assess the extent to which this mismatch could lead to biased results, we also want to test fitting to data vectors measured directly from simulations (as in, to ‘observe’ our simulations much like we observe the real Universe).

6.4.1 Stacked $\text{INDRA}_{\text{cg}}^s$

One problem with testing on data vectors measured from simulations, is that simulations with sufficient resolution that we can populate them with galaxies using an SHAM-like scheme are typically not of comparable volume to future surveys. This means that while we can test that the model gives reasonable results with a modest volume

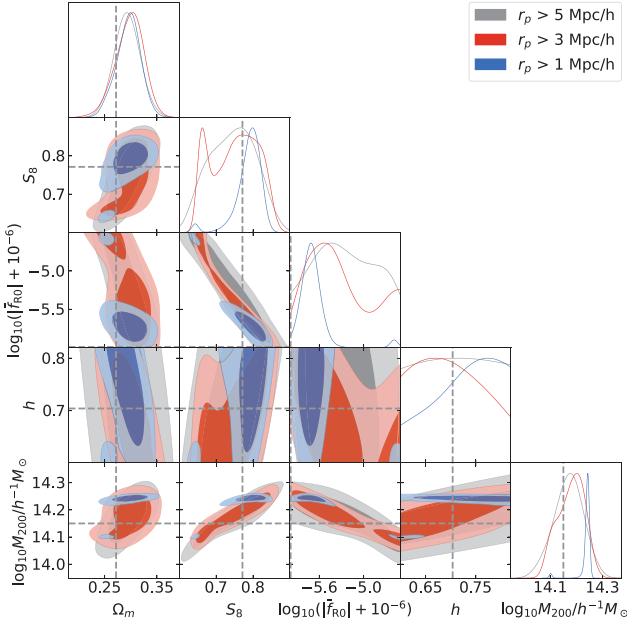


Figure 9. The posterior distribution for the cosmological parameters when fitting to a data vector calculated from a stack of the INDRA simulations. The different contours correspond to fitting the ξ_{cg}^s data vector down to different minimum r_p . The INDRA cosmology is marked by the dashed lines. Because the INDRA simulations were run with GR, they have $\bar{f}_{\text{R}0} = 0$, corresponding to $\log_{10} |\bar{f}_{\text{R}0} + 10^{-6}| = -6$, which is on the edge of our cosmological parameter prior.

of data, we cannot test that it returns results that are unbiased at the level of precision that will be afforded by future data sets.

To get around this, we return to the INDRA suite of simulations, which comprises many simulations, each of which has a $(1 h^{-1} \text{Gpc})^3$ volume. These simulations were all run with the same cosmology, and so we can combine the $\xi_{\text{cg}}^s(r_p, r_\pi)$ from many of them to get a simulated data vector, at the INDRA cosmology, from a large simulated volume.¹⁴ Specifically, we calculate $\xi_{\text{cg}}^s(r_p, r_\pi)$ from each INDRA simulation (using our fiducial cluster and galaxy definitions), and then take the mean of this over the different INDRA simulations to get our stacked INDRA data vector (this is the same as $\langle \xi_{\text{sim}}^{s,k} \rangle$ from Section 5.1).

We fit to this stacked INDRA data vector using MCMC, with the model $\xi_{\text{cg}}^s(\theta)$ calculated from our FORGE-trained emulator for the GIK parameters, along with the true real-space clustering (for which we use the mean over the different INDRA simulations of $\xi_{\text{cg}}^r(r)$). The resulting posterior distribution is shown in Fig. 9, where the different colours correspond to different minimum r_p values down to which we fit the data.

A key takeaway from Fig. 9 is that if we make our fiducial choice of fitting to ξ_{cg}^s over the range $1 \leq r_p/h^{-1} \text{Mpc} \leq 15$, $0 \leq r_\pi/h^{-1} \text{Mpc} \leq 20$, then we confidently infer a cosmology that is inconsistent with the true cosmology used to run the INDRA simulations. This inconsistency goes down – both due to a broadening of the posterior, and due to a shifting of the mean of the posterior – if we increase the minimum value of r_p down to which we fit

¹⁴We note that this is not quite equivalent to having run a single large volume simulation, primarily due to the absence of density fluctuations with wavelengths larger than the boxsize in each of the INDRA simulations.

the data. The fact that using the GIK model to predict ξ_{cg}^s on scales $\lesssim 3 h^{-1} \text{Mpc}$ can lead to biases in the inferred cosmology should not come as a surprise given the evidence for model mis-specification that we saw on these scales in Fig. 6.

It is interesting to consider why the constraints are noticeably tighter when one includes scales down to $1 h^{-1} \text{Mpc}$. This suggests that there is considerable cosmological information contained in the velocities of galaxies $1 - 3 h^{-1} \text{Mpc}$ away from clusters. We hypothesize that this is because the radius at which $f(R)$'s Chameleon screening turns on (and so inside of which, the usual GR forces are obtained) is of order an Mpc. For example, at $z = 0$, and in a halo slightly less massive than the clusters we consider here, the screening radius is between 1 and 2 Mpc when $\log_{10} |\bar{f}_{\text{R}0}| = -6$ (see fig. 7 of Moran, Teyssier & Li 2015). If some enhancement to gravity happens everywhere, then it is hard to distinguish between a more massive cluster and enhanced gravity. However, with data that spans regions that – with $f(R)$ – would be both screened and unscreened, it is no longer the case that enhancing gravity (in only the unscreened regions) is equivalent to increasing the cluster mass.

We should stress that the GIK parameter emulator used with these different minimum r_p values was subtly different. We had originally fit the GIK parameters for each FORGE simulation over the radial range $1 < r/h^{-1} \text{Mpc} < 20$, and used these GIK parameter posteriors when training our GIK parameter emulator. However, we found that when only fitting to the ξ_{cg}^s data vector at $r_p > 3 h^{-1} \text{Mpc}$, we got a less biased result if our GIK parameter emulator was trained using GIK parameters fit to the radial range $3 < r/h^{-1} \text{Mpc} < 20$ rather than $1 < r/h^{-1} \text{Mpc} < 20$. This makes sense, because if the GIK parameters returned by fitting to $r > 1$ and $> 3 h^{-1} \text{Mpc}$ are different, the $r > 3 h^{-1} \text{Mpc}$ case must be a better description of the cluster–galaxy pairwise velocity distribution at large radii. Forcing the same set of GIK parameters to simultaneously explain the velocity distribution within $1 < r/h^{-1} \text{Mpc} < 3$ (where we know the model is not an accurate description of the velocity distribution) then degrades how well the GIK model describes the velocity distribution at large radii. Given that ξ_{cg}^s at some r_p is affected only by the clustering and velocity distribution at 3D radii $r \geq r_p$, if we only need to model the data vector at $r_p > r_{\text{min}}$, then it makes sense to only fit the GIK parameters to radii $r > r_{\text{min}}$.

An avenue for future work would be to improve the GIK model such that it provides a better description of $P(v_r, v_t|r)$ at small r . This could involve both improving upon the seven-parameter model for the velocity distribution at a single radius (equation 4), which we have seen in Fig. 2 is not a perfect description of the velocity distribution found in N -body simulations, as well as finding improved functional forms for how these seven GIK functions vary with radius (e.g. the form of equation 7).

For now, we note that while our GIK model-based emulator allows for the extraction of cosmological information from RSD on scales below that at which linear theory predictions break down, it does not appear that we can use it to model cluster–galaxy clustering, all the way down to scales below the virial radii of the clusters. This is somewhat disappointing given that one of the key features of the GIK model is that the velocity distribution has two components, one of which describes galaxies that are virialized within galaxy cluster potentials (and which is only relevant on scales around and below cluster virial radii).

Another potential explanation for the biased cosmological parameter inference seen in Fig. 9 is differences between the FORGE and INDRA simulations. If the INDRA and FORGE simulations predicted

different GIK parameters for the same cosmology, then using a FORGE-trained emulator to fit to INDRA mock data would lead to such a bias.¹⁵ While this effect is almost certainly present at some level, the reasonable convergence of the GIK functions with respect to numerical resolution seen in Fig. 4 suggests that the differences between INDRA and FORGE (most notably that INDRA has eight times worse mass resolution) should not lead to dramatic differences.

To be more quantitative about this, we inspected the GIK parameters predicted by our FORGE-trained emulator, at the INDRA cosmology (there is no FORGE simulation run at the INDRA cosmology). We define the following:

- (i) $\mathbf{g}_{\text{FORGE}}(\boldsymbol{\theta}_{\text{INDRA}})$, the GIK parameters predicted by the FORGE-trained emulator, at the cosmology of the INDRA simulations.
- (ii) $\mathbf{g}_{\text{INDRA}}^k$, the GIK parameters measured for the k th INDRA simulation.
- (iii) $\langle \mathbf{g}_{\text{INDRA}}^k \rangle$, the mean of the GIK parameters from the different INDRA simulations.
- (iv) $\boldsymbol{\theta}_{\xi_{\text{cg}}^s}$, the maximum-likelihood cosmological parameters when fitting to the stacked-INDRA ξ_{cg}^s data vector (the ‘ $r_p > 1 h^{-1} \text{Mpc}$ ’ fit in Fig. 9).
- (v) $\mathbf{g}_{\text{FORGE}}(\boldsymbol{\theta}_{\xi_{\text{cg}}^s})$, the GIK parameters predicted by the FORGE-trained emulator, at $\boldsymbol{\theta}_{\xi_{\text{cg}}^s}$. These are the GIK parameters being used in the best-fitting model to the stacked-INDRA ξ_{cg}^s .
- (vi) $\boldsymbol{\Sigma}_{\mathbf{g}}^{\text{INDRA}}$, the covariance matrix for \mathbf{g} calculated from the set of $\mathbf{g}_{\text{INDRA}}^k$.

Then, defining $\chi^2(\mathbf{g}_a, \mathbf{g}_b)$ as $(\mathbf{g}_a - \mathbf{g}_b)^\top (\boldsymbol{\Sigma}_{\mathbf{g}}^{\text{INDRA}})^{-1} (\mathbf{g}_a - \mathbf{g}_b)$, we find that $\chi^2(\mathbf{g}_{\text{FORGE}}(\boldsymbol{\theta}_{\text{INDRA}}), \langle \mathbf{g}_{\text{INDRA}}^k \rangle) = 177$, while $\chi^2(\mathbf{g}_{\text{FORGE}}(\boldsymbol{\theta}_{\xi_{\text{cg}}^s}), \langle \mathbf{g}_{\text{INDRA}}^k \rangle) = 571$. So the difference between $\boldsymbol{\theta}_{\xi_{\text{cg}}^s}$ and $\boldsymbol{\theta}_{\text{INDRA}}$ is not because $\mathbf{g}_{\text{FORGE}}(\boldsymbol{\theta}_{\xi_{\text{cg}}^s})$ does a better job than $\mathbf{g}_{\text{FORGE}}(\boldsymbol{\theta}_{\text{INDRA}})$ of producing GIK functions that look like those found in INDRA. This points towards the dominant effect being model misspecification, that results in the optimal set of GIK parameters to describe ξ_{cg}^s not being the same as those one gets by fitting the GIK parameters directly to the position/velocity data from simulations.

Nevertheless, $\chi^2(\mathbf{g}_{\text{FORGE}}(\boldsymbol{\theta}_{\text{INDRA}}), \langle \mathbf{g}_{\text{INDRA}}^k \rangle) = 177$ (compared with an expectation of ~ 24 if the FORGE prediction at the INDRA cosmology looked like the GIK parameters from a random INDRA volume) suggests there are systematic differences between INDRA and FORGE, and/or errors induced by the emulation process. In order to get a sense of how robust the GIK parameters are to the details of the simulations, in the future it would be good to compare the GIK parameters fit to different simulations run with the same cosmology, and also to build emulators with different suites of simulations and compare their results.

7 DISCUSSION

The procedure that we have presented, combining an emulator for the cluster–galaxy pairwise velocity distribution with some external knowledge (or model) for the real-space clustering, is an interesting alternative to more direct emulation of redshift-space clustering data vectors from simulations. As described in Section 5.1 it has distinct advantages and disadvantages, and so is worth considering as the community tackles how to maximally extract small-scale information from the next generation of redshift surveys.

¹⁵This would also suggest that the GIK parameters are not robustly predicted by simulations, which would be bad for the prospects of applying this model to observational data.

7.1 Required developments

Before a method like the one we have described here can be applied to real observational data, there is some additional work that is required. We outline some of these developments now.

7.1.1 Variations in the galaxy–halo connection

On sufficiently large scales, galaxies are expected to be almost unbiased tracers of the cosmic velocity field (Chen et al. 2018). However, effects like dynamical friction (which affects dark matter haloes, but not individual dark matter particles) can lead to haloes (and hence galaxies) having velocity distributions that differ from that of the matter distribution on smaller scales (e.g. Armitage et al. 2018).

In this work we used (sub)haloes as our galaxies, which should produce a more realistic velocity bias than had we simply used dark matter particles. However, exactly which haloes get populated with galaxies could potentially alter the velocity bias, leading to different GIK parameters for different galaxy samples. This means that in order to apply our method to real data, it would be important to populate the simulations – used to train the GIK parameter emulator – with galaxies in a manner that is close to how the real Universe populated its dark matter haloes with galaxies.

Exactly which galaxies are selected in some survey (and therefore how those galaxies populate the underlying dark matter structures) is expected to lead to differences in their kinematics. For example, Orsi & Angulo (2018) found that populating a DMO simulation with galaxies using a semi-analytic model lead to different kinematics for stellar mass selected galaxies and galaxies with prominent emission lines. As a simple example, if part of what quenches star formation in galaxies and leads them to turn from blue to red is environmental effects, one expects some blue galaxies that fall into a cluster will be made red once inside. Within the GIK model this would produce quite different $f_{\text{vir}}(r)$ for red and blue galaxies, with (at fixed radius) blue galaxies more likely to be infalling, and red galaxies more likely to be virialized within the cluster potential.

Potentially better than trying to put galaxies into the simulations in the ‘correct’ way, would be to use a parametrized model to populate the simulations with galaxies, populating each simulation numerous times using different galaxy–halo connection parameters. The GIK parameter emulator could then be built to cover variations in both cosmological and galaxy–halo connection parameters. Then, the galaxy–halo connection parameters could be left free in the fit, and marginalized over when drawing conclusions about cosmological parameters. This sort of approach is what is done when directly emulating redshift-space clustering data vectors from simulations (Zhai et al. 2019; Yuan et al. 2022; Lange et al. 2023; Kwan et al. 2023). In our case, it would be non-trivial, because fitting for the GIK parameters is a reasonably computationally expensive step, so doing this many times over with different galaxy–halo connection parameters for each simulated cosmology would require a considerable amount of computing time. Nevertheless, such an endeavour would be important to properly incorporate how uncertainty on the way galaxies trace dark matter leads to uncertainties on the GIK parameters for a given cosmology.

7.1.2 Prescription for the real-space clustering

Throughout this paper, we have assumed that the real-space clustering is known, and that we only need to model the effects of RSD on this to get ξ_{cg}^s . When applying this method to real data,

the real-space clustering will not be known a priori, and must either be measured or modelled. Measuring the real-space clustering could be done by measuring the projected correlation function (usually denoted $w_{\text{cg}}(\theta)$, which expresses the cross-correlation between the positions-on-the-sky of clusters and galaxies) and applying an inverse Abel transform (e.g. Hickstein et al. 2019) to get $\xi_{\text{cg}}^r(r)$. Modelling $\xi_{\text{cg}}^r(r)$ could involve using an emulator for the real-space clustering’s dependence on cosmology (e.g. Cuesta-Lazaro et al. 2023), or just using a flexible parametric model for $\xi_{\text{cg}}^r(r)$ and adding $w_{\text{cg}}(\theta)$ to the data vector being fit to, in order to help constrain this model’s parameters.

We note that inferring the real-space clustering has some associated challenges. For example, the projected clustering

$$w_{\text{cg}}(r_p) = \int_{-r_{\pi,\text{max}}}^{r_{\pi,\text{max}}} \xi_{\text{cg}}^s(r_p, r_\pi) \quad (22)$$

integrates over the line of sight, removing any dependence on RSD. However, this is only exactly true for infinite $r_{\pi,\text{max}}$, and Hang et al. (2022) (when fitting models to the galaxy–group cross-correlation from the Galaxy and Mass Assembly (GAMA) survey) found that a large enough $r_{\pi,\text{max}}$ to achieve results that were independent of RSD effects, lead to measurements that were too noisy to be useful.

In addition, the projected clustering faces observational challenges associated with how galaxy spectra are obtained. For example, both the Sloan Digital Sky Survey (SDSS) and now DESI cannot obtain spectra for sufficiently nearby pairs of galaxies without multiple observations of the same field (Hahn et al. 2017; Smith et al. 2019). The bias to the measured correlation function that would come from preferentially missing galaxy pairs with small separations can be corrected for (e.g. Bianchi & Percival 2017), but these observational systematic effects nevertheless increase the uncertainties on small-scale clustering. This means that our inferred real-space clustering may have significant uncertainty, which would translate into increased uncertainty on the velocity distribution (and hence the cosmological parameters) when fitting the GIK model to ξ_{cg}^s .

7.1.3 Distribution of cluster masses

At present, we measure the GIK parameters from simulations for multiple narrow cluster mass bins, and use this to build an emulator for $p(v_r, v_t|r, M)$ (the cluster–galaxy pairwise velocity distribution, for a given cluster–galaxy separation and cluster mass). A realistic cluster sample, selected on a property like richness or X-ray luminosity, would be expected to have a sizeable range of cluster masses. If we have a sample of clusters with a number density per unit cluster mass of dn/dM , then one might imagine that we can find the GIK parameters appropriate for this sample of clusters as some sort of weighted average of the GIK parameters for different cluster masses. If we take equation (11) and make explicit the dependence of various quantities on the mass of the galaxy clusters (M), we have at a particular mass that

$$1 + \xi_{\text{cg}}^s(r_p, r_\pi|M) = \frac{H(z)}{1+z} \times \int \frac{1 + \xi_{\text{cg}}^r(r|M)}{\cos \theta} \int p(v_r, v_t|r, M) dv_r dy. \quad (23)$$

The average redshift-space cross-correlation with galaxies for our sample of clusters is

$$\langle \xi_{\text{cg}}^s(r_p, r_\pi) \rangle = \int \xi_{\text{cg}}^s(r_p, r_\pi|M) p(M) dM, \quad (24)$$

where $p(M) = \frac{dn}{dM} / \int \frac{dn}{dM} dM$ is the normalized probability density associated with the cluster mass distribution. Similarly, the real-space clustering of our cluster sample is

$$\langle \xi_{\text{cg}}^r(r) \rangle = \int \xi_{\text{cg}}^r(r|M) p(M) dM. \quad (25)$$

Putting equation (23) into equation (24), we get

$$1 + \langle \xi_{\text{cg}}^s(r_p, r_\pi) \rangle = \frac{H(z)}{1+z} \times \int p(M) \left[\int \frac{1 + \xi_{\text{cg}}^r(r|M)}{\cos \theta} \int p(v_r, v_t|r, M) dv_r dy \right] dM. \quad (26)$$

The form of this integral means that we cannot simply calculate $\langle \xi_{\text{cg}}^s(r_p, r_\pi) \rangle$ using $\langle \xi_{\text{cg}}^r(r) \rangle$ and some appropriately averaged $\langle p(v_r, v_t|r) \rangle$. To properly calculate a model $\langle \xi_{\text{cg}}^s(r_p, r_\pi) \rangle$ for a broad cluster mass distribution, we would need to have a model for $\xi_{\text{cg}}^r(r|M)$, a model for $p(v_r, v_t|r, M)$ (which our emulator for the GIK parameters provides), and a model for $p(M)$.

This is problematic for the approach (described in Section 7.1.2) of using the observed projected clustering to infer $\xi_{\text{cg}}^r(r)$, as this would produce a measurement of $\langle \xi_{\text{cg}}^r(r) \rangle$ – averaged over the cluster sample – without knowing about its mass dependence. For cluster mass distributions that are reasonably narrow, it may be sufficiently accurate to combine some sort of mass-averaged $p(v_r, v_t|r)$ with $\langle \xi_{\text{cg}}^r(r) \rangle$. Generating cluster samples with suitably narrow mass distributions will be helped by the ongoing development of low-scatter cluster mass proxies (e.g. Eckert et al. 2020), but we leave an investigation of how well this would work, and how one would best average $p(v_r, v_t|r, M)$, to future work.

If one instead takes the approach of having a parametrized model for $\xi_{\text{cg}}^r(r)$ and including $w(\theta)$ in the data vector being fit to, then this could be extended to have a parametrized model for $\xi_{\text{cg}}^r(r|M)$ as well as $p(M)$. Combining these with $p(v_r, v_t|r, M)$ from the GIK parameter emulator we can then calculate both the redshift-space and projected correlation functions, allowing the parameters of the $\xi_{\text{cg}}^r(r|M)$ and $p(M)$ models to be constrained by the clustering data.

7.1.4 Extension to higher redshifts

In this work, our GIK parameter emulator was built only using simulation data from $z = 0$. Extending this to work at different redshifts would simply require using simulation outputs at different redshifts. We could either build an emulator for the specific effective redshift of some target survey, or we could build the emulator with redshift being an additional input parameter along with cosmology and cluster mass, and use simulation outputs at many different redshifts in the GIK parameter training data. This emulator could then be applied to data from any redshift (including clustering data in different redshift bins).

7.1.5 Selection effects

Our model for redshift-space clustering assumes that the real-space clustering is isotropic, and that RSD due to cluster/galaxy velocities breaks this. However, the selection of our clusters and/or galaxies may break this assumption of isotropy. For example, the processes involved in optical cluster selection typically lead to the overdensities selected as clusters being preferentially elongated along the line of sight (Dietrich et al. 2014). If the cluster shape is correlated with the

distribution of galaxies around it, then this would lead to the real-space clustering of galaxies around clusters already being anisotropic (with respect to the line of sight to the cluster) before accounting for RSD.

Assessing the potential impact of this, as well as testing any mitigation strategies, would most-likely require realistic mock data to which cluster selection algorithms could be applied and a mock analyses performed. For example, the Cardinal mock galaxy catalogues (To et al. 2023) accurately reproduce the abundance of galaxy clusters, and could be used to assess how selection effects (especially orientation-dependent ones) might lead to biases in the inferred cosmological parameters.

7.1.6 Artificial disruption in simulations

Our approach relies on N -body simulations faithfully reproducing the cluster–galaxy pairwise velocity distribution. There has been recent literature about artificial disruption/destruction of subhaloes in simulations (e.g. van den Bosch et al. 2018; van den Bosch & Ogiya 2018; Green, van den Bosch & Jiang 2021; Diemer, Behroozi & Mansfield 2023), with claims that a significant fraction of subhaloes that should survive are destroyed due to numerical effects. If the subhaloes artificially destroyed preferentially have certain orbits, their destruction would bias the inferred GIK parameters, although this effect should primarily be on small scales (within approximately the cluster virial radii) where the GIK model cannot currently be applied to infer unbiased cosmological constraints anyway.

Fully understanding the potential impact of this would require a dedicated simulation study. Here, we note that the relatively good agreement in Fig. 4 between the GIK functions for TNG300, and a simulation with 64 times worse mass resolution, is a cause for optimism. That said, van den Bosch & Ogiya (2018) warn that because of the way in which both softening and particle mass are typically varied as simulation resolution is changed, it is possible for results to appear converged while not having converged to the true (infinite resolution) result.

7.1.7 The effects of simulation box size

As previously stated, a key advantage of our method is that it does not suffer too much from the noise associated with relatively small simulation volumes (see Section 5.1), enabling the use of small simulations to build an emulator for the pairwise velocities. However, small volume simulations do not just provide noisier estimates of quantities compared with their larger box-size counterparts, but can produce biased estimates. This arises if large-scale density fluctuations (not present in smaller volume simulations) impact the quantities of interest.

Large-scale power is certainly important for the distribution of the peculiar velocities of dark matter haloes (see e.g. fig. 9 of Cole 1997). However, it is not clear whether the pairwise velocities of haloes/subhaloes with separations much smaller than the box-size are appreciably impacted by the finite box-size and the associated absence of large-scale power. Maleubre et al. (2023) recently presented results for the pairwise velocities of the matter field, from ‘scale-free’ simulations, which have power-law power spectra. Box size effects were found to be strongly dependent on the spectral index of the power spectrum in these simulations, making it hard to extrapolate to the case of Λ CDM-like universes, in which the power spectrum is not scale free. We are not aware of work looking at this in

a Λ CDM context, but a future measurement of the pairwise velocity distribution as a function of box size is one of the steps that should be taken before a method like ours is applied to real data, in order to demonstrate the level at which the GIK parameters inferred from simulations might be biased by the box size of the simulations used.

7.2 Null tests of general relativity

Here, we have built an emulator for a specific theory of modified gravity, $f(R)$. This is just one of many alternatives to GR, and tying our modelling procedure to a particular modified gravity theory is somewhat undesirable. To understand alternative strategies to test theories of gravity, it is instructive to consider the case of RSD on larger scales. On large scales, we can make use of linear theory, for which the matter power spectrum in redshift space, $P_m^s(k, \mu)$, is given by (Kaiser 1987)

$$P_m^s(k, \mu) = (1 + f\mu^2)^2 P_m^r(k), \quad (27)$$

where $P_m^r(k)$ is the (isotropic) real-space power spectrum, $\mu \equiv \hat{\mathbf{k}} \cdot \hat{\mathbf{z}}$ is the cosine of the angle between wave vector \mathbf{k} and the line-of-sight direction, and f is the linear growth rate (for a definition, see e.g. Hamilton 2001). Within a Λ CDM + GR universe, the value of f is closely approximated by $f \approx \Omega_m^{0.55}$ (Linder 2005). Nevertheless, when fitting to observational data, f and Ω_m can be treated as independent parameters. The inferred values of f and Ω_m can then be compared, to see if the relationship between them is that expected in a Λ CDM + GR universe, or if they appear inconsistent with this. Treating the standard cosmological model as some sort of null hypothesis, and then seeing if there is evidence for deviations to this, is attractive because it does not tie the analysis to any particular alternative, but just seeks to investigate whether there is evidence for some departure from Λ CDM + GR.

On the face of it, our method for modelling ξ_{cg}^s is not so amenable to this style of analysis, because the emulator is essentially interpolating between a set of self-consistent simulations, and so it cannot produce predictions that are not self-consistent (such as a Λ CDM + GR universe with $f \not\approx \Omega_m^{0.55}$). It is interesting to consider how we could extend our modelling procedure to be more analogous to the case described above (where f and Ω_m are treated as being independent when modelling RSD on linear scales). One possibility would be to treat our RSD model primarily as a way to measure the mass of the cluster sample, assuming GR within the model. We could build an emulator using Λ CDM + GR simulations (which, no longer having a variable \bar{f}_{R0} , would no longer suffer the $M_{200} - \bar{f}_{R0}$ degeneracy that we see in our results). We could then use this emulator to fit to an observed data vector. Comparing the mass inferred from fitting to ξ_{cg}^s with that inferred using an independent method (with weak lensing being an obvious choice), we could then see if these two masses agree or not, with a disagreement hinting towards non-GR gravity.

Another alternative, would be to again build an emulator for the GIK parameters using Λ CDM + GR simulations, but then introduce one or more rescaling parameters that would shift the GIK function curves. For example, one could imagine a single rescaling parameter, f_{vel} , such that $v_{r,c} \rightarrow f_{\text{vel}} v_{r,c}$, $\sigma_{\text{rad}} \rightarrow f_{\text{vel}} \sigma_{\text{rad}}$, and $\sigma_{\text{tan}} \rightarrow f_{\text{vel}} \sigma_{\text{tan}}$. Fitting for the cosmological parameters alongside f_{vel} , evidence that $f_{\text{vel}} \neq 1$ would point towards some sort of modification to gravity.

8 CONCLUSIONS

We have presented a method to model the cluster–galaxy cross-correlation function in redshift space, ξ_{cg}^s , based upon a slightly modified version of the GIK model from ZW13. We demonstrated

how to find the posterior distribution for the parameters of the GIK model from 3D simulation data, with the data required being both positions and velocities of clusters and galaxies extracted from N -body simulations. Repeating this procedure for simulations run with different cosmologies (from the FORGE simulations, that simulate different $f(R)$ modified gravity cosmologies), we built up training data for the cosmology dependence of the GIK parameters.

We used this training data to build Gaussian process emulators for how the different GIK parameters depend upon the cosmological parameters and the theory of gravity. Combining these GIK parameter emulators with knowledge of the real-space cluster–galaxy cross-correlation, allowed us to calculate ξ_{cg}^s as a function of cosmological parameters. Comparing these model predictions with an observed data vector, we can then make inferences about the cosmological parameters.

The finite volume of the simulations, leads to uncertainties on our training data. We presented a method to propagate uncertainty on the GIK parameters from the training data, through to uncertainty on the model data vector. This included a procedure that we call ‘GIK parameter diagonalization’, a linear transform of the GIK parameter vector that allows separate Gaussian Processes (one for each ‘diagonalized’ GIK parameter) to properly account for the covariance between uncertainties on the different GIK parameters in the training data.

Fitting our model to mock data generated by the model itself, we showed forecasts for the cosmological constraining power that RSD in the cluster–galaxy correlation function on small scales will provide with data from a survey like DESI. We found that a combination of the typical mass of our cluster sample and the strength of modifications to gravity was tightly constrained, such that an independent measurement of cluster mass (which could be provided by weak gravitational lensing) will enable RSD to tightly constrain theories of gravity.

Finally, we applied our method to mock data extracted directly from N -body simulations. We found that our procedure leads to a biased inference on the cosmological parameters if fitting the correlation function all the way down to a scale of $1 h^{-1}$ Mpc. However, these biases are strongly reduced when cutting out the smallest scales ($< 3 h^{-1}$ Mpc), at the expense of broadened constraints on the cosmological parameters. We attribute these biases to deficiencies in the model on a scale of $1 - 3 h^{-1}$ Mpc, with these deficiencies visible in both Figs 2 and 6.

A large fraction of the cosmological constraining power from current and future redshift surveys comes from small scales, which will most likely require modelling methods based around the use of N -body simulations. Our approach – of using simulations to build an emulator for the RSD kernel, which can be combined with some other model for the real-space clustering – is quite different from other approaches in the literature that directly measure the redshift-space correlation function from simulations (Zhai et al. 2019; Yuan et al. 2022; Lange et al. 2023; Kwan et al. 2023). Advantages of our approach include:

(i) It separates out the two conceptually different components that produce the redshift-space clustering: the real-space clustering and the RSD kernel. This allows developments to be made to either component independently, and can also help with understanding and intuition.

(ii) It allows simulations with substantially smaller volumes than observational surveys to be used in the modelling of those large surveys, because imposing a functional form for the RSD kernel reduces the impact of noise in the simulations (see Fig. 6). This

means that more simulations can be run, covering a wider range of possible cosmologies.

(iii) It uses simulations to predict the ingredient of small-scale redshift-space clustering that they appear to predict more robustly. As discussed in Appendix C, for the TNG300 simulations: the RSD kernel’s contribution to ξ_{cg}^s is better converged with respect to numerical resolution than the real-space clustering’s contribution to ξ_{cg}^s is.

The main drawback of our approach is that it relies on the GIK model being a good description of the true pairwise velocities. While this generally seems to be true, there is evidence that at radii $\lesssim 3 h^{-1}$ Mpc the ‘infalling’ component of the GIK model does not adequately describe the velocities from simulations (see the ‘banana-shaped’ $P(v_r, v_t)$ data and best-fitting model in Fig. 2). An avenue for future research would be to investigate modifications to the functional form for $P(v_r, v_t)$ that allow it to better describe the simulation data, with the goal being a model that could be applied down to very small scales without leading to biased inferences of the cosmological parameters.

ACKNOWLEDGEMENTS

The authors thank Christian Arnold for providing us with access to the FORGE simulations, and thank David Weinberg, Chun-Hao To, Peter Taylor, Alkistis Pourtsidou, Benjamin Bose, and John Peacock for helpful discussions. This research was carried out at the Jet Propulsion Laboratory (JPL), California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004). The authors are grateful for funding from JPL’s R&TD 7x ‘Future of Dark Sector Cosmology: Systematic Effects and Joint Analysis’ (01STRS/R.23.312.005). The High Performance Computing resources used in this investigation were provided by funding from the JPL Information and Technology Solutions Directorate.

This work made use of the following software packages: **ASTROPY** (Astropy Collaboration 2013; Price-Whelan et al. 2018), **EMCEE** (Foreman-Mackey et al. 2013), **GETDIST** (Lewis 2019), **MATPLOTLIB** (Hunter 2007), **NUMPY** (van der Walt, Colbert & Varoquaux 2011), and **SCIPY** (Virtanen et al. 2020). This work used some of the **IllustrisTNG** simulations (Pillepich et al. 2018; Springel et al. 2018; Nelson et al. 2018; Naiman et al. 2018; Marinacci et al. 2018). It also used the **INDRA** simulations (Falck et al. 2021), accessing them via the SciServer science platform (Taghizadeh-Popp et al. 2020).

DATA AVAILABILITY

The **INDRA** and **IllustrisTNG** simulation data are publicly available, while the **FORGE** simulation data can be obtained from the authors of Arnold et al. (2022). The derived data products underlying this article (including GIK parameter posteriors for the **FORGE** simulations, data covariance matrices calculated from **INDRA**, and Gaussian process kernel hyperparameters) will be shared on reasonable request to the corresponding author.

REFERENCES

- Abbott T. M. C. et al., 2020, *Phys. Rev. D*, 102, 023509
 Ambikasaran S., Foreman-Mackey D., Greengard L., Hogg D. W., O’Neil M., 2015, *IEEE Trans. Pattern Anal. Mach. Intell.*, 38, 252
 Angulo R. E., Pontzen A., 2016, *MNRAS*, 462, L1

- Armitage T. J., Barnes D. J., Kay S. T., Bahé Y. M., Dalla Vecchia C., Crain R. A., Theuns T., 2018, *MNRAS*, 474, 3746
- Arnold C., Li B., Giblin B., Harnois-Déraps J., Cai Y.-C., 2022, *MNRAS*, 515, 4161
- Astropy Collaboration, 2013, *A&A*, 558, A33
- Aung H., Nagai D., Rozo E., Wolfe B., Adhikari S., 2023, *MNRAS*, 521, 3981
- Azzalini A., Capitanio A., 2009, preprint (arXiv:0911.2342)
- Bianchi D., Percival W. J., 2017, *MNRAS*, 472, 1106
- Brax P., 2013, *Class. Quantum Gravity*, 30, 214005
- Buchdahl H. A., 1970, *MNRAS*, 150, 1
- Chen J., Zhang P., Zheng Y., Yu Y., Jing Y., 2018, *ApJ*, 861, 58
- Cole S., 1997, *MNRAS*, 286, 38
- Cuesta-Lazaro C., Li B., Eggemeier A., Zarrouk P., Baugh C. M., Nishimichi T., Takada M., 2020, *MNRAS*, 498, 1175
- Cuesta-Lazaro C. et al., 2023, *MNRAS*, 523, 3219
- Davis M., Efstathiou G., Frenk C. S., White S. D. M., 1985, *ApJ*, 292, 371
- DESI Collaboration, 2016, preprint (arXiv:1611.00036)
- Diemer B., Behroozi P., Mansfield P., 2023, preprint (arXiv:2305.00993)
- Dietrich J. P. et al., 2014, *MNRAS*, 443, 1713
- Eckert D., Finoguenov A., Ghirardini V., Grandis S., Kaefer F., Sanders J., Ramos-Ceja M., 2020, *Open J. Astrophys.*, 3, 12
- Escoffier S. et al., 2016, preprint (arXiv:1606.00233)
- Euclid Collaboration, 2019, *MNRAS*, 484, 5509
- Falck B., Wang J., Jenkins A., Lemson G., Medvedev D., Neyrinck M. C., Szalay A. S., 2021, *MNRAS*, 506, 2659
- Foreman-Mackey D., Hogg D. W., Lang D., Goodman J., 2013, *PASP*, 125, 306
- Frieman J. A., Turner M. S., Huterer D., 2008, *ARA&A*, 46, 385
- García R., Salazar E., Rozo E., Adhikari S., Aung H., Diemer B., Nagai D., Wolfe B., 2023, *MNRAS*, 521, 2464
- Green S. B., van den Bosch F. C., Jiang F., 2021, *MNRAS*, 503, 4075
- Grove C. et al., 2022, *MNRAS*, 515, 1854
- Guo H. et al., 2015, *MNRAS*, 446, 578
- Hahn C., Scoccimarro R., Blanton M. R., Tinker J. L., Rodríguez-Torres S. A., 2017, *MNRAS*, 467, 1940
- Hamilton A. J. S., 2001, *MNRAS*, 322, 419
- Hang Q. et al., 2022, *MNRAS*, 517, 374
- Hartlap J., Simon P., Schneider P., 2007, *A&A*, 464, 399
- Hickstein D. D., Gibson S. T., Yurchak R., Das D. D., Ryazanov M., 2019, *Rev. Sci. Instrum.*, 90, 065115
- Howlett C., Percival W. J., 2017, *MNRAS*, 472, 4935
- Hu W., Sawicki I., 2007, *Phys. Rev. D*, 76, 064004
- Hunter J. D., 2007, *Comput Sci Eng*, 9, 90
- Joyce A., Lombriser L., Schmidt F., 2016, *Annu. Rev. Nucl. Part. Sci.*, 66, 95
- Kaiser N., 1987, *MNRAS*, 227, 1
- Kazin E. A., Sánchez A. G., Blanton M. R., 2012, *MNRAS*, 419, 3223
- Kilbinger M., 2015, *Rep. Prog. Phys.*, 78, 086901
- Kokron N., Chen S.-F., White M., DeRose J., Maus M., 2022, *J. Cosmol. Astropart. Phys.*, 2022, 059
- Kuruwilla J., Porciani C., 2018, *MNRAS*, 479, 2256
- Kwan J. et al., 2023, *ApJ*, 952, 27
- Lange J. U., Hearin A. P., Leauthaud A., van den Bosch F. C., Xhakaj E., Guo H., Wechsler R. H., DeRose J., 2023, *MNRAS*, 520, 5373
- Lewis A., 2019, preprint (arXiv:1910.13970)
- Linder E. V., 2005, *Phys. Rev. D*, 72, 043529
- Maion F., Angulo R. E., Zennaro M., 2022, *J. Cosmol. Astropart. Phys.*, 2022, 036
- Maleubre S., Eisenstein D. J., Garrison L. H., Joyce M., 2023, *MNRAS*, 525, 1039
- Marinacci F. et al., 2018, *MNRAS*, 480, 5113
- Moran C. C., Teyssier R., Li B., 2015, *MNRAS*, 448, 307
- More S., Diemer B., Kravtsov A. V., 2015, *ApJ*, 810, 36
- Naiman J. P. et al., 2018, *MNRAS*, 477, 1206
- Nelson D. et al., 2018, *MNRAS*, 475, 624
- Nelson D. et al., 2019, *Comput. Astrophys. Cosmol.*, 6, 2
- Norberg P., Baugh C. M., Gaztañaga E., Croton D. J., 2009, *MNRAS*, 396, 19
- Orsi Á. A., Angulo R. E., 2018, *MNRAS*, 475, 2530
- Parra G., Tobar F., 2017, preprint (arXiv:1709.01298)
- Peacock J. A., Smith R. E., 2000, *MNRAS*, 318, 1144
- Peebles P. J. E., 1980, *The Large-scale Structure of the Universe*. Princeton University Press, Princeton, NJ
- Pillepich A. et al., 2018, *MNRAS*, 475, 648
- Price-Whelan A. M. et al., 2018, *AJ*, 156, 123
- Rasmussen C. E., Williams C. K. I., 2006, *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA
- Reddick R. M., Wechsler R. H., Tinker J. L., Behroozi P. S., 2013, *ApJ*, 771, 30
- Scoccimarro R., 2004, *Phys. Rev. D*, 70, 083007
- Shao J., Wu C. F. J., 1989, *Ann. Stat.*, 17, 1176
- Smith A. et al., 2019, *MNRAS*, 484, 1285
- Springel V., White S. D. M., Tormen G., Kauffmann G., 2001, *MNRAS*, 328, 726
- Springel V. et al., 2018, *MNRAS*, 475, 676
- Taghizadeh-Popp M. et al., 2020, *Astron. Comput.*, 33, 100412
- Taylor A., Joachimi B., Kitching T., 2013, *MNRAS*, 432, 1928
- To C.-H. et al., 2024, *ApJ*, 961, 59
- van den Bosch F. C., Ogiya G., 2018, *MNRAS*, 475, 4066
- van den Bosch F. C., Ogiya G., Hahn O., Burkert A., 2018, *MNRAS*, 474, 3043
- van der Walt S., Colbert S. C., Varoquaux G., 2011, *Comput. Sci. Eng.*, 13, 22
- Villumsen J. V., Davis M., 1986, *ApJ*, 308, 499
- Virtanen P. et al., 2020, *Nat. Methods.*, 17, 261
- Wang B., Chen T., 2015, *Chemometr. Intell. Lab. Syst.*, 142, 159
- Wechsler R. H., Tinker J. L., 2018, *ARA&A*, 56, 435
- Weinberg D. H., Mortonson M. J., Eisenstein D. J., Hirata C., Riess A. G., Rozo E., 2013, *Phys. Rep.*, 530, 87
- Yuan S., Garrison L. H., Eisenstein D. J., Wechsler R. H., 2022, *MNRAS*, 515, 871
- Zhai Z. et al., 2019, *ApJ*, 874, 95
- Zhou R. et al., 2023, *AJ*, 165, 58
- Zu Y., Weinberg D. H., 2013, *MNRAS*, 431, 3319
- Zu Y., Weinberg D. H., Jennings E., Li B., Wyman M., 2014, *MNRAS*, 445, 1885

APPENDIX A: FUNCTIONAL FORM OF THE GIK FUNCTIONS

As discussed in Section 3.2, the variation of the GIK functions with radius is described by the set of GIK parameters. In Table A1, we provide the parametric forms for the GIK functions. We note that:

(i) The functional forms for σ_{rad} , σ_{tan} , dof , and α are all the same as one another (and the same as that used by ZW13 for these GIK functions), but we parametrize them slightly differently based on our experience of parameter degeneracies when fitting these functions to simulation data.

(ii) For f_{vir} we use the same functional form as in ZW13, but we fix $\gamma = 2$ rather than $\gamma = 3$.

(iii) We use a different functional form for σ_{vir} from that in ZW13. This was because the form listed in ZW13, equation (9) did not seem to agree with what was plotted in ZW13, fig. 6. The functional form we used was inspired by Aung et al. (2023).

Table A1. The functional forms for the various GIK functions, with the corresponding free parameters (the ‘GIK parameters’) listed. A GIK parameter vector, \mathbf{g} , is the set of 24 dimensionless numbers listed in the ‘GIK parameters’ column, working from the top down. Our prior on \mathbf{g} (used when measuring the GIK parameters directly from position/velocity data from simulations) is a separable function of each GIK parameter, and is a uniform prior in each case, with the upper and lower limits enclosed in square brackets in the ‘Prior’ column.

GIK function	GIK parameters	Prior	Notes on GIK function
$\sigma_{\text{rad}}(r) = q - qf \frac{r}{r_{\text{min}}} \left(\frac{\beta r_{\text{min}}}{r + (\beta - 1)r_{\text{min}}} \right)^\beta$	$\log_{10} q / \text{km s}^{-1}$ f $\log_{10} \beta$ $\log_{10} r_{\text{min}} / h^{-1} \text{ Mpc}$	[2,3.5] [0,0.99] [0.04,1.7] [-0.5, 2]	
$\sigma_{\text{tan}}(r) = q - qf \frac{r}{r_{\text{min}}} \left(\frac{\beta r_{\text{min}}}{r + (\beta - 1)r_{\text{min}}} \right)^\beta$	$\log_{10} A / \text{km s}^{-1}$ f $\log_{10} \beta$ $\log_{10} r_{\text{min}} / h^{-1} \text{ Mpc}$	[1.5,3] [0,0.99] [0.04,1.7] [-0.5, 2]	A is the minimum value of $\sigma_{\text{tan}}(r)$, given by $A = q(1 - f)$
$v_{r,c}(r) = q - t \left(\frac{r_i}{r + r_i} \right)^\beta$	$q / \text{km s}^{-1}$ $\log_{10} t / \text{km s}^{-1}$ β $\log_{10} r_i / h^{-1} \text{ Mpc}$	[-1000, 1000] [1,4] [0.1,4] [-1.5, 2]	
$\text{dof}(r) = 1 + q - qf \frac{r}{r_{\text{min}}} \left(\frac{\beta r_{\text{min}}}{r + (\beta - 1)r_{\text{min}}} \right)^\beta$	A f $\log_{10} \beta$ $\log_{10} r_{\text{min}} / h^{-1} \text{ Mpc}$	[0,10] [0,1] [0.1,4] [-0.5, 2]	$A = q(1 - f)$ $\text{dof} \geq 1$
$\alpha = q - qf \frac{r}{r_{\text{min}}} \left(\frac{\beta r_{\text{min}}}{r + (\beta - 1)r_{\text{min}}} \right)^\beta$	q h $\log_{10} \beta$ $\log_{10} r_{\text{min}} / h^{-1} \text{ Mpc}$	[-3, 10] [-10, 10] [0.04,1.7] [-0.5, 2]	$h = f q$
$f_{\text{vir}}(r) = \exp(-(r/r_0)^\gamma)$	$r_0 / h^{-1} \text{ Mpc}$	[0,10]	We fix $\gamma = 2$
$\sigma_{\text{vir}}(r) = \sigma_0 \left(1 + q \exp\left(\frac{-r}{r_{\text{vir}}}\right) \right)$	$\sigma_0 / \text{km s}^{-1}$ q $r_{\text{vir}} / h^{-1} \text{ Mpc}$	[100,2000] [0,2] [0,5]	Inspired by Aung et al. (2022)

APPENDIX B: ALTERNATIVE LIKELIHOOD FOR FITTING GIK PARAMETERS TO A SIMULATION

For our fiducial INDRA simulation volume (for which ξ_{cg}^s is plotted in Fig. 1), we tested whether our inferred GIK parameter vector, \mathbf{g} , was different if we used the likelihoods defined in equation (5) or (8). For the first of these, the data that we fit to is calculated by binning cluster–galaxy pairs into radial shells and then within each radial shell further binning them in velocity (v_r , v_t) space. We then used MCMC to fit \mathbf{g} to these binned counts, assuming that the counts obey Poisson statistics, with an expectation value for each pixel that depends on \mathbf{g} . For the second likelihood, the data are simply the set of r , v_r , and v_t for each cluster–galaxy pair, with the likelihood being the product over all cluster–galaxy pairs of the probability density $P(v_r, v_t | r)$.

The results of these MCMCs are chains of samples drawn from the posteriors on the GIK parameters, which we can map into draws from

the posteriors on the GIK functions by evaluating the functions at the locations of the GIK parameter samples. In Fig. B1, we plot these GIK function posteriors, showing the median GIK function values as well as the 2σ uncertainties. The results are virtually indistinguishable between using the two different likelihoods.

In terms of how important the small differences in Fig. B1 are to our use case of using the GIK model to make model predictions for ξ_{cg}^s , we performed the following test. Keeping the real-space clustering fixed, we used the maximum-likelihood GIK parameters from both the binned and no-binned likelihood approaches to calculate ξ_{cg}^s . The difference between these two ξ_{cg}^s model data vectors, when expressed as a χ^2 using the data covariance matrix described in Section 6.1, is 0.16, very small compared with the differences one gets when calculating ξ_{cg}^s through the GIK model using INDRA simulations run with the same cosmology but with different random initial conditions (see Section 5.1).

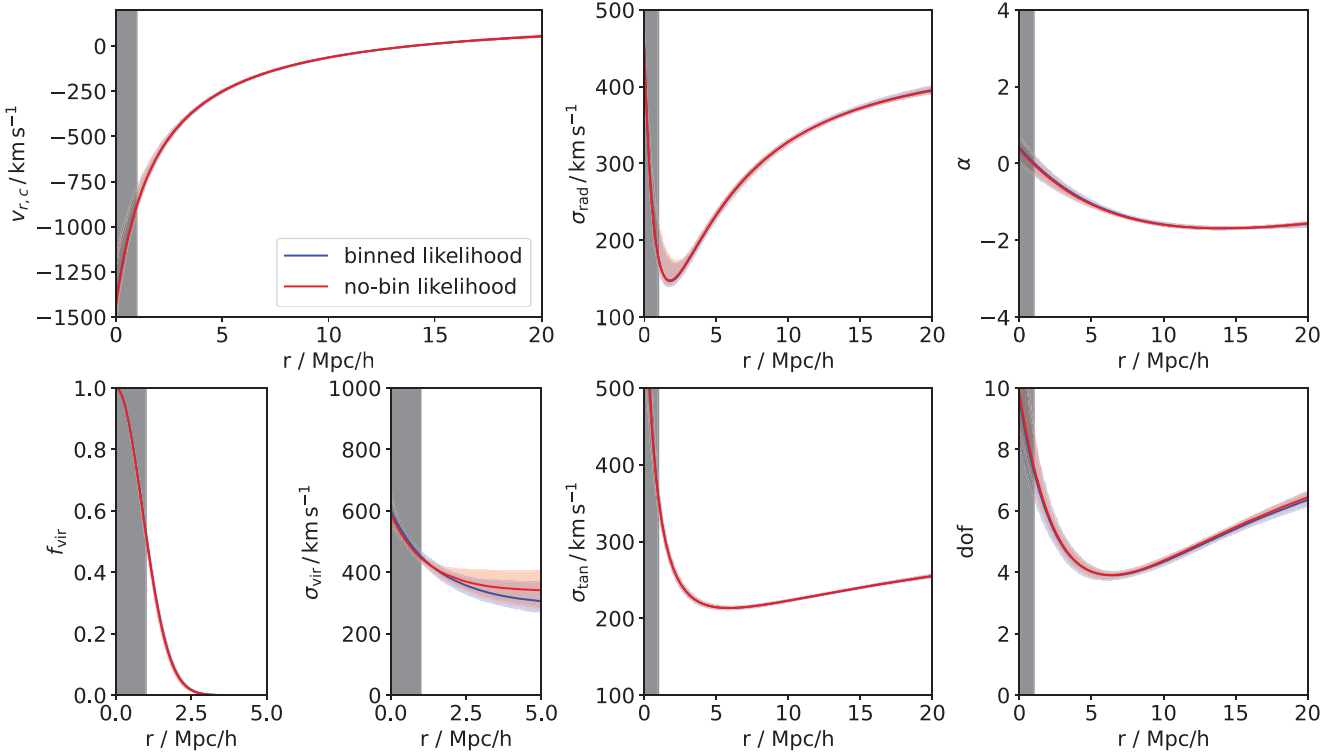


Figure B1. The GIK function posteriors, from fitting for the GIK parameters using the binned and no-binned likelihoods described in Appendix B. The solid lines show the median GIK functions as a function of radius, while the shaded regions show the 2σ uncertainty (the 2.5th–97.5th percentiles of the GIK functions at each radius).

APPENDIX C: THE IMPACT OF SIMULATION RESOLUTION AND THE INCLUSION OF BARYONIC PHYSICS ON THE PREDICTED ξ_{cg}^s

In Section 3.3, we presented the results for both the GIK functions and the real-space clustering of different variants of the TNG300 simulation. Here, we show the relative importance of these differences for the calculation of a model ξ_{cg}^s .

For each simulation, we measured the maximum-likelihood GIK parameters, \mathbf{g} , as well as the real-space clustering, $\xi_{\text{cg}}^r(r)$. We then used combinations of \mathbf{g} and ξ_{cg}^r measured from different simulations to calculate various model ξ_{cg}^s . The similarity of two different ξ_{cg}^s , say $\xi_{\text{cg},1}^s$ and $\xi_{\text{cg},2}^s$, was then assessed in terms of $\chi^2 = \xi_{\text{cg},1}^s \mathbf{C}^{-1} \xi_{\text{cg},2}^s$, using the data covariance matrix, \mathbf{C} , described in Section 6.1 (rescaled for the $\sim (300 \text{ Mpc})^3$ volume of a TNG300 simulation). The results of various combinations were as follows:

(a) When keeping the real-space clustering fixed to ξ_{cg}^r from TNG300, but using \mathbf{g} from TNG300 or TNG300 low-res, the χ^2 difference was 1.0.

(b) When keeping \mathbf{g} fixed to that from TNG300, but using ξ_{cg}^r from TNG300 or TNG300 low-res, the χ^2 difference was 18.

(c) When keeping the real-space clustering fixed to ξ_{cg}^r from TNG300, but using \mathbf{g} from TNG300 or \mathbf{g} from TNG300 DMO, the χ^2 difference was 0.12.

(d) When keeping \mathbf{g} fixed to that from TNG300, but using ξ_{cg}^r from TNG300 or TNG300 DMO, the χ^2 difference was 1.7.

Comparison of (a) and (b) with (c) and (d) shows that ξ_{cg}^s is more affected by a factor 64 change in resolution than baryons versus no baryons. Comparing (a) with (b) we see that using low-resolution

simulation GIK parameters leads to a better estimate of the high-resolution ξ_{cg}^s , than if using low-resolution simulation real-space clustering. And comparing (c) with (d) tells us that the same is true when considering a hydrodynamical simulation versus a DMO one.

APPENDIX D: WHAT IS THE PRIMARY SOURCE OF MODEL MIS-SPECIFICATION ERROR?

The right-hand panel of Fig. 6 made clear that there is substantial model mis-specification on small scales ($\lesssim 3 h^{-1} \text{ Mpc}$). This means that the 24 GIK parameters are not adequately describing the velocity distribution at small radii. There are two obvious culprits for this, one is that the seven GIK functions do not adequately describe the pairwise velocity distribution at fixed radius, and the second is that the GIK parameters do not adequately describe the radial dependence of the GIK functions. Knowing the relevant importance of these sources of model mis-specification should aid with future efforts to improve the GIK model.

To assess this, we calculated model ξ_{cg}^s data vectors that did not make use of the GIK parameter description of how the GIK functions vary with radius. Instead, having fit for the GIK functions in many radial shells, we directly used the GIK functions to describe the velocity distribution at radii within the relevant shell. Specifically, the GIK functions used to calculate a model ξ_{cg}^s were (over the radial range $[r_i, r_{i+1})$) the maximum-likelihood set of GIK functions fit to the $\{v_r, v_t\}$ data for cluster–galaxy pairs with separation in the range $[r_i, r_{i+1}]$. We used radial shells that were $1 h^{-1} \text{ Mpc}$ thick.

We generated model ξ_{cg}^s using both our fiducial approach (‘GIK parameters’) and the approach just described (‘GIK functions’) for

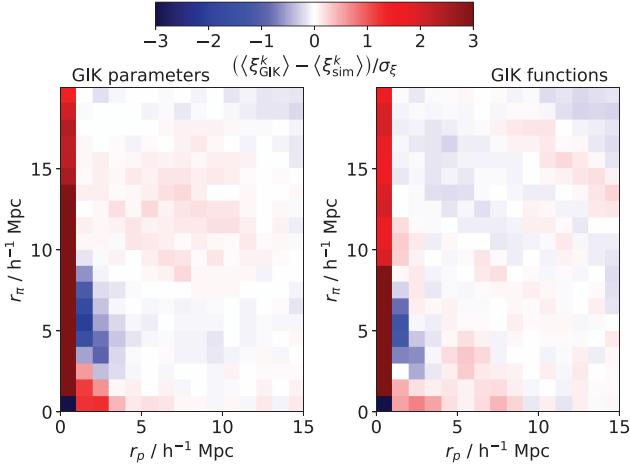


Figure D1. Maps of the normalized residuals, similar to those in Fig. 6. Each panel shows the mean result over 24 INDRA simulations. The mean ξ_{cg}^s from the GIK models is compared with the mean ξ_{cg}^s measured directly from the INDRA simulations. In the left-hand panel, the model ξ_{cg}^s is calculated in the usual way (where the velocity distribution at all radii is described by the 24 ‘GIK parameters’). In the right-hand panel, the pairwise velocity distribution used for the model ξ_{cg}^s treats each GIK function as constant within $1h^{-1}$ Mpc thick radial shells, with the values of the GIK functions within each radial shell obtained as the maximum-likelihood values from fitting to the $\{v_r, v_t\}$ data in that shell.

each of 24 INDRA simulations for which we had fit the GIK functions to the velocity data in spherical shells. Fig. D1 shows the residuals between the mean of these 24 model ξ_{cg}^s , and the mean ξ_{cg}^s measured directly from the INDRA simulations, normalized by an error map, σ_ξ (defined in Section 5.1.1). The ‘GIK functions’ residuals plot shows

similar structure on small scales to the ‘GIK parameters’ one, which means that a deficiency in the GIK functions is at least partly to blame for the mismatch between model and data ξ_{cg}^s on small scales. Nevertheless, the residuals are less pronounced in the ‘GIK functions’ case than the ‘GIK parameters’ one, which means that the model could be improved somewhat without altering the seven parameter model describing $P(v_r, v_t)$ at fixed radius, but simply by improving the description of how this varies with radius. Quantitatively, we found that the corresponding χ^2 (using the covariance matrix from Section 6.1, which ignores pixels in the lowest r_p bin) for the residuals plotted in Fig. D1 were 23 and 11, for the ‘GIK parameters’ and ‘GIK functions’ cases, respectively.

Achieving an improvement to the radial dependence of the GIK functions could be a case of devising better functional forms, that may then require more GIK parameters. An alternative, would be to use the Gaussian processes that currently model the cosmology dependence of the GIK parameters, to also model the radial dependence of the GIK functions. In essence, one would use a suite of simulations to measure the GIK functions at different points in cosmological parameter space, as well as for different cluster masses and different cluster–galaxy separations. Then, a Gaussian process could be constructed for each of the GIK functions, that predicts the value of this GIK function as a function of cosmology, halo mass, and separation. Such a procedure could only improve the model so far though, given that the right-hand panel of Fig. D1 still shows residuals, that would require an improvement to the GIK functions themselves to remove. As noted in Section 5.1.2, the comparison between the data and model $P(v_r, v_t)$ in Fig. 6 provides some guidance as to how the current GIK functions are deficient at small scales.

This paper has been typeset from a $\text{\TeX}/\text{\LaTeX}$ file prepared by the author.