

ON SOFT CLUSTERING FOR CORRELATION ESTIMATORS

EDWARD BERMAN^{1,2,18 *}, SNEH PANDYA^{1,3}, JACQUELINE MCCLEARY^{1 *}, MARKO SHUNTOV^{4,5}, CAITLIN CASEY^{4,6,17}, NICOLE DRAKOS⁷, ANDREAS FAISST⁸, STEVEN GILLMAN^{4,9}, GHASSEM GOZALIASL^{10,11}, NATALIE B. HOGG¹², JEYHAN KARTALTEPE¹³, ANTON KOEKEMOER¹⁴, WILFRIED MERCIER¹⁵, DIANA SCOGNAMIGLIO¹⁶, AND COSMOS-WEB: THE JWST COSMIC ORIGINS SURVEY

¹Department of Physics, Northeastern University, Boston, MA 02115, USA

²Department of Mathematics, Northeastern University, Boston, MA 02115, USA

³NSF AI Institute for Artificial Intelligence and Fundamental Interactions (IAIFI)

⁴Cosmic Dawn Center (DAWN), Denmark

⁵Niels Bohr Institute, University of Copenhagen, Jagtvej 128, DK-2200, Copenhagen, Denmark

⁶Department of Astronomy, The University of Texas at Austin, Austin, TX, USA

⁷Department of Physics and Astronomy, University of Hawaii, Hilo, 200 W Kawili St, Hilo, HI 96720, USA

⁸Caltech/IPAC, 1200 E. California Blvd., Pasadena, CA 91125, USA

⁹DTU-Space, Technical University of Denmark, Elektrovej 327, DK-2800 Kgs. Lyngby, Denmark

¹⁰Department of Computer Science, Aalto University, P.O. Box 15400, FI-00076 Espoo, Finland

¹¹Department of Physics, University of Helsinki, P.O. Box 64, FI-00014 Helsinki, Finland

¹²Laboratoire Univers et Particules de Montpellier, CNRS & Université de Montpellier, Parvis Alexander Grothendieck, Montpellier, France 34090

¹³Laboratory for Multiwavelength Astrophysics, School of Physics and Astronomy, Rochester Institute of Technology, 84 Lomb Memorial Drive, Rochester, NY 14623, USA

¹⁴Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218, USA

¹⁵Aix Marseille University, CNRS, CNES, LAM, Marseille, France

¹⁶Jet Propulsion Laboratory, California Institute of Technology, 4800, Oak Grove Drive - Pasadena, CA 91109, USA

¹⁷Department of Physics, University of California Santa Barbara, CA, 93106, CA and

¹⁸Visiting Scientist, Center for Astrophysics Harvard and Smithsonian, AstroAI

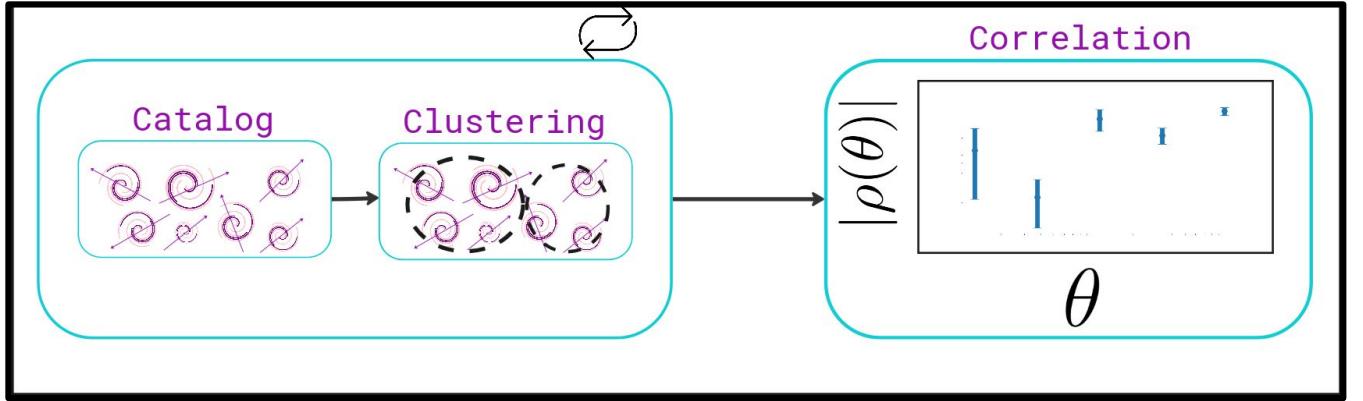
Version September 13, 2025

ABSTRACT

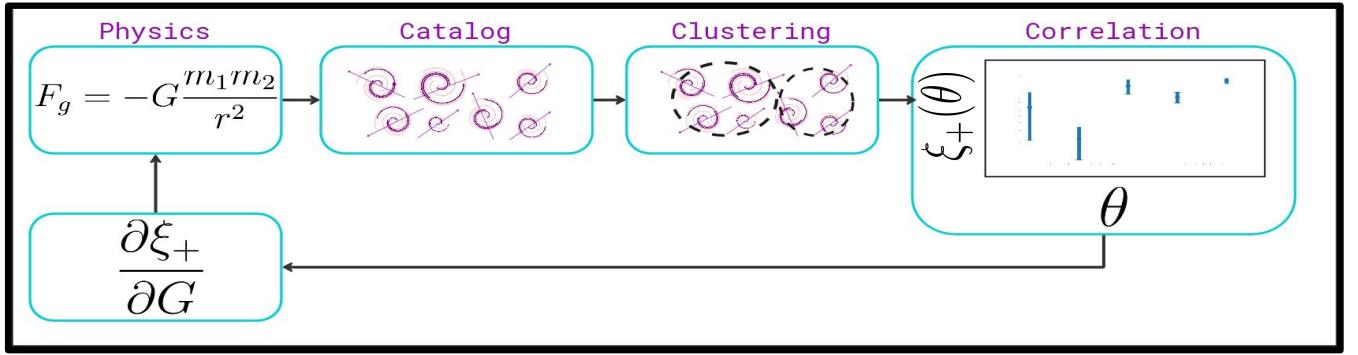
Properly estimating correlations between objects at different spatial scales necessitates $\mathcal{O}(n^2)$ distance calculations. For this reason, most widely adopted packages for estimating correlations use clustering algorithms to approximate local trends. However, methods for quantifying the error introduced by this clustering have been understudied. In response, we present an algorithm for estimating correlations that is probabilistic in the way that it clusters objects, enabling us to quantify the uncertainty caused by clustering simply through model inference. These soft clustering assignments enable correlation estimators that are theoretically differentiable with respect to their input catalogs. Thus, we also build a theoretical framework for differentiable correlation functions and describe their utility in comparison to existing surrogate models. Notably, we find that repeated normalization and distance function calls slow gradient calculations and that sparse Jacobians destabilize precision, pointing towards either approximate or surrogate methods as a necessary solution to exact gradients from correlation functions. To that end, we close with a discussion of surrogate models as proxies for correlation functions. We provide an example that demonstrates the efficacy of surrogate models to enable gradient-based optimization of astrophysical model parameters, successfully minimizing a correlation function output. Our numerical experiments cover science cases across cosmology, from point spread function (PSF) modeling efforts to gravitational simulations to galaxy intrinsic alignment (IA). 

Subject headings: Cosmology, Instrumentation and Methods, Weak Gravitational Lensing, Uncertainty Quantification, Differentiable Programming

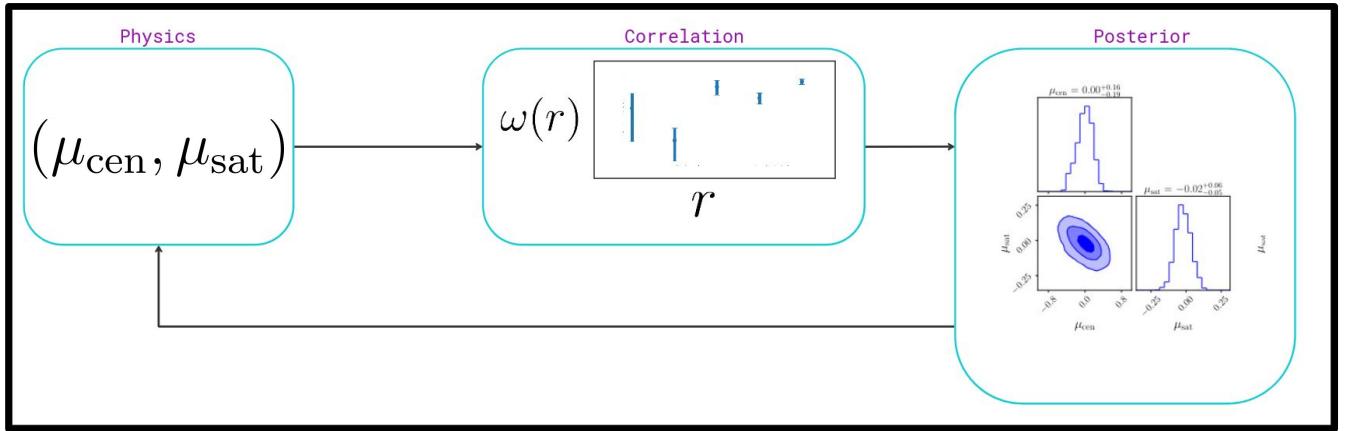
* E-mail: {berman.ed,j.mccleary}@northeastern.edu



(a) An overview of our model uncertainty experiment. We repeatedly cluster using a probabilistic algorithm and look at the resulting standard deviation of the resulting correlations.



(b) An overview of our differentiability experiment. We forward model a gravitational simulation and show that we can differentiate through the estimator even with the clustering approximation.



(c) An overview of our surrogates experiment. We exploit the differentiability of neural networks to enable Hamiltonian Monte Carlo for fast posterior sampling of the IA parameters most likely to minimize a correlation function.

FIG. 1.— Outline of the three experiments conducted in this paper. The top outlines our model uncertainty experiment (§5), the middle outlines our differentiability experiment (§6), and the bottom outlines our surrogate experiment (§7).

1. INTRODUCTION

Two- and three-point correlation functions (2/3PCF) are widely used in cosmology. Algorithms to compute these correlation functions first cluster their objects into representative points to ease the computational load of computing $\mathcal{O}(n^2)$ pairwise distances. To numerically compute the correlation function, the pairwise distance between all of the points is then computed, and an estimator is computed on all of the points in each distance bin.

A long-standing goal in the computational sciences is to reconcile epistemic (model) uncertainties from aleatoric (data) uncertainties (Hüllermeier & Waegeman 2021; Osband et al. 2023). Aleatoric uncertainties describe uncertainty that is inherent to a statistic. This can be due to data quality limitations or inherent stochasticity of the quantity one is trying to model. On the other hand, epistemic uncertainties reflect the inability to generalize from a finite data set due to inherent limitations of the model. In the case of 2PCFs and 3PCFs, the epistemic uncertainty is often overlooked: the best assignment of an object to a given cluster is often unclear, as data points that are “in-between” two or more learned cluster centers can reasonably be chosen to fall in any of the clusters. This ambiguity introduces an epistemic uncertainty, as the model will struggle to appropriately assign objects to representative clusters without enough data. With more data points, it is easier to identify the most appropriate way to cluster the objects and thus epistemic uncertainty should decrease in the main, though individual data points will still face ambiguity. Prior works usually assume this clustering uncertainty to be trivial, arguing that for a significantly large sample, the number of overestimated and underestimated distances are roughly equal, and so the errors tend to cancel out (Jarvis et al. 2004). For example, this was assumed for the Dark Energy Survey (DES) point spread function (PSF) modeling efforts (Jarvis et al. 2021). A PSF is an impulse response of an optical system to light, and the quality of a PSF model is often assessed with ρ statistics (Rowe 2010; Vogelsberger et al. 2016), which describe correlations in size and shape residuals of a PSF model relative to observed stars. These empirical PSF models are often trained using point source catalogs that are generated by SourceExtractor++ (Bertin & Arnouts 1996) and filtered to be within a given size/magnitude range. DES covered a vast survey area and included millions of objects which were used to be fit and evaluate PSF models (Dark Energy Survey Collaboration 2024; Jarvis et al. 2021). With so much data at hand, it was fair to assume that the number of overestimated and underestimated distances were roughly equal, and that downstream modeling errors from clustering were negligible. Surveys such as DES therefore look exclusively at uncertainties quantified via bootstrapping or jackknife (Jarvis et al. 2021; Jarvis & collaborators 2024), which captures only the model’s robustness to individual data points. This is closer in nature to being an aleatoric uncertainty, and we will refer to it as aleatoric uncertainty throughout.

This work presents a method to quantify epistemic uncertainty in regimes where one cannot make the same assumptions as Jarvis et al. (2004) due to the small number of data points available. For instance, clustering errors cannot be assumed to be negligible for the NIRCam PSF modeling efforts with COSMOS-Web (Casey et al. 2023), where the number of point sources is on the order of a few 100 across the entire field of view (Berman et al. 2024). Even through the TreeCorr algorithm (Jarvis et al. 2004) makes sure distance offsets are less than a bin width, results can still be heavily biased. A toy example of this with the position-position autocorrelation is presented in Figure 2. The unbiased estimator for this correlation is the Landy–Szalay estimator (Landy & Szalay 1993) shown in Equation 1.

$$\omega(\theta) = \frac{DD(\theta) - 2DR(\theta) + RR(\theta)}{RR(\theta)} \quad (1)$$

Equation 1 assumes access to one real observation of galaxies as well as an additional catalog of galaxies randomly distributed on the sky. The terms in Equation 1 are $DD(\theta)$, the number of real galaxy pairs separated by an angular distance θ , $RR(\theta)$, the number of random galaxy pairs separated by a distance θ , and $DR(\theta)$, the number of pairs between one real galaxy and one random galaxy separated by a distance θ . Since the estimator is directly proportional to the number of galaxies that are measured to be θ arcmins apart through $DD(\theta)$, even a single bin offset can heavily skew a measurement. While ρ statistics themselves are not directly related to the number of objects in a single distance bin, the example indicates that binning offsets can constructively bias the result in a given bin.

The core method of this work is to recast object-cluster assignment probabilistically. In this way, we study how the uncertainty related to the clustering of objects can propagate through to the output. Our findings suggest that there are equal amounts of data and model uncertainty in the case of computing ρ statistics. This further implies that one should skip the clustering step to mitigate the clustering uncertainty.

We also observe that the soft or probabilistic assignment of galaxies to cluster centers makes our estimator theoretically differentiable with respect to its input catalog, and by extension the parameters of any differentiable forward model that produces an input catalog. Differentiability in this context is highly desirable, as it allows one to relate

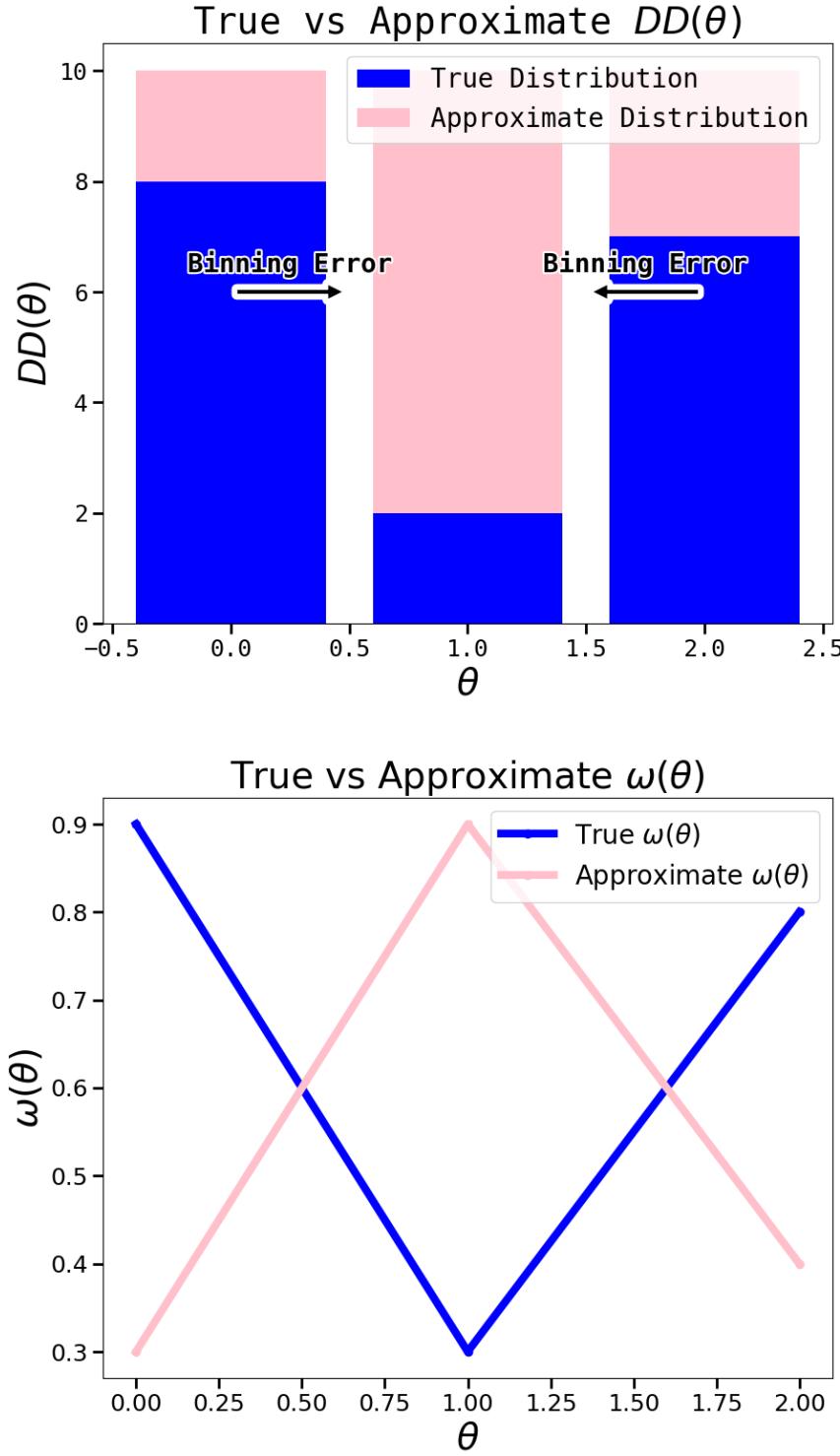


FIG. 2.— A toy example of a clustering approximation causing a downstream error in the estimation of the $\omega(\theta)$ position-position auto-correlation function.

correlations to astrophysical model parameters through gradient-based optimization (for related work, see Lanzieri et al. 2023; Cuesta-Lazaro & Mishra-Sharma 2024; Lanusse et al. 2023; Campagne et al. 2023; Jagvaral et al. 2024; Berman & McCleary 2025; Chantada et al. 2023; Halverson & Pandya 2024). Halverson & Pandya (2024) in particular describes a pipeline analogous to what we would like to achieve. They optimize cosmological model parameters to maximize the number of extended cosmological epochs where the relative abundances of different Ω_i remain unchanged, referred to as periods of cosmological stasis (Dienes et al. 2022). In our case, we are interested in what kind of implicit

cosmological parameters maximize or minimize different types of two and three-point correlations (cf. Figure 1). Motivated by this desire to maximize or minimize correlation functions, we provide a discussion of the gradient related properties of our algorithm. In contrast to our method, clustering schemes that use hard assignments (i.e. every object has one assignment with unit probability) often depend on functions like MEDIAN or ARGMIN, which results in derivatives that are either identically zero or ill-defined. We further explore the differentiability of our clustering scheme with an experiment. In our experiment, we use a gravitational simulation as a forward model with the goal of differentiating a correlation function with respect to the gravitational constant, G . Our findings motivate the use of surrogate models: learned functions that approximate the relationship between astrophysical model parameters and observable data. Recent studies demonstrate their ability to directly model two-point statistics from astrophysical model parameters (Pandya et al. 2024, 2025), making them ideal for our purpose. Accordingly, we extend the work of Pandya et al. (2024, 2025), using the Intrinsic Alignment Emulator (IAEmu) surrogate model to optimize seven halo-based modeling parameters to minimize a correlation function. In doing so, we illustrate the effectiveness of surrogate models to enable gradient-based optimization of astrophysical model parameters, which in turn allows us to study what physical circumstances cause correlations to be maximized or minimized.

Our paper executes three experiments that are designed to explore three themes: model uncertainty, differentiability through an estimator, and differentiability through a surrogate.

1. **Model Uncertainty and PSFs (§5):** For this experiment, we use ShOpt.jl (Berman et al. 2024; Berman & McCleary 2024) to produce PSF models for the COSMOS-Web NIRCam images (Casey et al. 2023). Using our proposed algorithm, we then compute ρ statistics. Finally, we assess the epistemic uncertainty associated with our model and compare it to the uncertainty we would get with traditional bootstrapping techniques. Our PSF modeling experiment is geared towards observers, who do not have access to differentiable forward models of PSF residuals.
2. **Differentiability and Gravity Simulations (§6):** For this experiment, we forward model gravitational interactions using a series of ordinary differential equations. We then compute correlation functions on the resulting catalog and test different techniques for differentiating the correlation function outputs with respect to the cosmological parameters that determined the simulation. Our gravitational simulation experiment is designed with theorists in mind. Theorists can produce multiple realizations of their model through simulations that are derived from different cosmological parameters. Here, we assess the feasibility of differentiating a correlation function with respect to G .
3. **Surrogates and Galaxy Intrinsic Alignment (§7):** For this experiment, we use an existing surrogate model to relate astrophysical model parameters to IA (Pandya et al. 2024, 2025). Using this surrogate, we use Hamiltonian Monte Carlo (HMC) to find posterior distributions over astrophysical model parameters that are likely to correspond to correlations of zero. Our IA experiment shows how to compensate for forward models and estimators that are not easily differentiated, and are again geared towards theorists who are able to realize many instances of their forward model through inference in order to train a surrogate. Our usage of HMC leverages the differentiability of neural networks to accelerate the sampling procedure in comparison to traditional algorithms like Metropolis–Hastings (Betancourt 2017; Robert et al. 2004).

The remainder of the paper is organized as follows: §2 discusses the relevant notation and theory for ρ statistics and IA, §3 follows with an outline of the data products used in our study, §4 lays out additional notation as well as details for the algorithm used in this paper, and §5 -7 presents the results of the experiments described above followed by our conclusions in §8.

2. NOTATION AND THEORY

In this section we discuss the relevant theory and notation for correlations of PSF size and shape residuals and of IA. We also expand on the theory we use to compute $\sigma_{\text{epistemic}}$ and $\sigma_{\text{aleatoric}}$.

2.1. PSF Modeling

A PSF describes the impulse response of an optical system to light. Effects like diffraction, optical aberrations, atmospheric turbulence (if applicable), and telescope jitter are summarized in the telescope’s PSF. For the James Webb Space Telescope (JWST), the PSF provides an obstacle for science goals including the highest resolution dark

matter mass maps (Scognamiglio 2024, Scognamiglio et al., in prep) and the characterization of over 100 strong lens systems (Nightingale et al., in prep). ρ statistics are a suite of 2PCFs introduced in Rowe (2010) and expanded upon in Vogelsberger et al. (2016) for characterizing biases in size and shape measurement that arise from PSF modeling.

We adopt the same notation for describing ρ statistics as Jarvis et al. (2021) and McCleary et al. (2023):

$$\rho_1(\theta) \equiv \langle \delta e_{\text{PSF}}^*(\mathbf{x}) \delta e_{\text{PSF}}(\mathbf{x} + \boldsymbol{\theta}) \rangle \quad (2)$$

$$\rho_2(\theta) \equiv \langle e_{\text{PSF}}^*(\mathbf{x}) \delta e_{\text{PSF}}(\mathbf{x} + \boldsymbol{\theta}) \rangle \quad (3)$$

$$\rho_3(\theta) \equiv \left\langle \left(e_{\text{PSF}}^* \frac{\delta T_{\text{PSF}}}{T_{\text{PSF}}} \right) (\mathbf{x}) \left(e_{\text{PSF}} \frac{\delta T_{\text{PSF}}}{T_{\text{PSF}}} \right) (\mathbf{x} + \boldsymbol{\theta}) \right\rangle \quad (4)$$

$$\rho_4(\theta) \equiv \left\langle \delta e_{\text{PSF}}^*(\mathbf{x}) \left(e_{\text{PSF}} \frac{\delta T_{\text{PSF}}}{T_{\text{PSF}}} \right) (\mathbf{x} + \boldsymbol{\theta}) \right\rangle \quad (5)$$

$$\rho_5(\theta) \equiv \left\langle e_{\text{PSF}}^*(\mathbf{x}) \left(e_{\text{PSF}} \frac{\delta T_{\text{PSF}}}{T_{\text{PSF}}} \right) (\mathbf{x} + \boldsymbol{\theta}) \right\rangle \quad (6)$$

where e_{PSF} is the ellipticity of the real PSF, i.e., the star ellipticity, T_{PSF} is the size of the real PSF, δe_{PSF} is the difference between the ellipticity of the real and model PSFs at position \mathbf{x} , and δT_{PSF} is the difference between the sizes of the real and model PSFs at position \mathbf{x} . Brackets denote averages over all pairs within a separation $\boldsymbol{\theta}$, and asterisks denote complex conjugates. T_{PSF} is nominally defined as $2\sigma^2$, where σ^2 is the variance of the best fit elliptical Gaussian to the PSF.

ρ statistics specify the ξ_+ weak lensing shear correlations, which are given by

$$\xi_+(\theta) = \langle \epsilon_t(x + \theta) \epsilon_t(x) \rangle + \langle \epsilon_{\times}(x + \theta) \epsilon_{\times}(x) \rangle \quad (7)$$

and estimated by

$$\hat{\xi}_+(\theta) = \frac{\sum_{i,j} w_i w_j (\epsilon_{t,i} \epsilon_{t,j} + \epsilon_{\times,i} \epsilon_{\times,j})}{\sum_{i,j} w_i w_j}. \quad (8)$$

where $w_{i,j}$ represents the weight uncertainties of object (i/j) , $\epsilon_{t,i,j}$ represents tangential shear of object (i/j) and $\epsilon_{\times,i,j}$ represents the cross shear of object (i/j) (Kilbinger 2015; Schneider et al. 2002). The complex components of the products in Equations 2 - 6 are known to average out to zero due to parity symmetry (Rowe 2010; Schneider et al. 2006), and are thus ignored in the estimator.

2.2. Galaxy Intrinsic Alignment

Intrinsic alignment describe correlations between galaxy orientations and the underlying distribution of dark matter where they are embedded; see Lamman et al. (2023) for a review. The three relevant IA correlation functions are the galaxy position-position $\xi(r)$, position-orientation $\omega(r)$, and orientation-orientation $\eta(r)$ correlation functions, and are estimated according to Equations 9-11.

$$\xi(r) = \left\langle \frac{n(r) - \bar{n}(r)}{\bar{n}(r)} \right\rangle \quad (9)$$

$$\omega(r) = \langle |\hat{e}(\mathbf{x}) \cdot \hat{r}|^2 \rangle - \frac{1}{3} \quad (10)$$

$$\eta(r) = \langle |\hat{e}(\mathbf{x}) \cdot \hat{e}(\mathbf{x} + \mathbf{r})|^2 \rangle - \frac{1}{3} \quad (11)$$

where $n(r)$ is the number of galaxies separated by distance r , $\bar{n}(r)$ is the expected number of galaxies separated by distance r for a random distribution, x is the position vector of a galaxy, and $\hat{e}(x)$ is a 3D orientation unit vector of a galaxy.

2.3. Uncertainty Quantification

Given a probabilistic clustering algorithm for estimating correlations, $\sigma_{\text{epistemic}}$ is computed by estimating the correlation value in each distance bin n times and then computing the standard deviation of the distribution. For computing

$\sigma_{\text{aleatoric}}$ via bootstrapping or jackknife, one must first make the probabilistic clustering algorithm deterministic. In this work, we use greedy sampling to accomplish this; that is, each object is always assigned to the cluster with its highest membership probability. In this way, the model uncertainty is isolated from the data uncertainty. To perform bootstrapping, random subsets of the input data is removed over n iterations and the correlation value is estimated in each distance bin. $\sigma_{\text{aleatoric}}$ is then computed via the standard deviation of the output in each distance bin. We choose $n = 10$ samples to obtain the error bars.

Our definitions of aleatoric and epistemic are consistent with what is used in the machine learning literature (Hüllermeier & Waegeman 2021). It is worth noting that in the cosmology literature, these uncertainties are sometimes described as statistical and systematics, e.g. in Freedman et al. (2024). Following Freedman et al. (2024), we take our uncertainties to be additive. That is, correlation functions will have some value $\mu \pm \sigma_{\text{aleatoric}} \pm \sigma_{\text{epistemic}}$ in each distance bin. μ can be obtained from the average of the either the probabilistic or deterministic algorithm samples.

3. DATA

3.1. Point Source Catalog

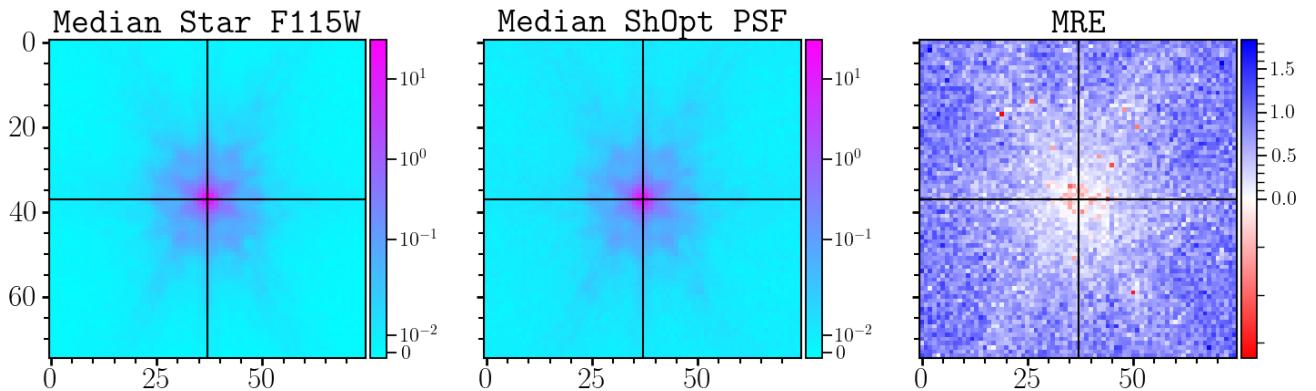


FIG. 3.— ShOpt PSF fit used in this work. The left panel shows the median of the vignettes. The center panel show the median PSF cutout. The right panel show the average relative error between the vignette cutouts and the PSF cutouts. More on these plots can be found in Berman et al. (2024).

The point sources used for generating our PSF models come from the SE++ (Bertin & Arnouts 1996) COSMOS-Web catalog (Casey et al. 2023, Shuntov et al. in prep). We restrict ourselves to imaging from the F115W NIRCam (Rieke et al. 2003, 2005; Beichman et al. 2012) wavelength filter, as the PSFs in this wavelength are the closest to being well-approximated by a Gaussian (Berman et al. 2024). ρ statistics and the related Hirata–Seljak–Mandelbaum (HSM) adaptive moments (Hirata & Seljak 2003; Mandelbaum et al. 2005) are based off of the Gaussian approximation of the PSF, and so the use of ρ statistics in our experiments is most appropriate in this wavelength. PSF models are obtained using ShOpt.jl (Berman et al. 2024; Berman & McCleary 2024). Our choice of PSF fitter is discussed in more detail in Appendix A. Our ShOpt fits can be seen in Figure 3. The ellipticities of the objects in our data set are also visualized in Figure 4. We use the same configuration file as found in the appendix of Berman et al. (2024). Our data set statistics for PSF modeling are summarized in Table 1.

Training Stars	Validation Stars	Filter	Survey Area (deg ²)
6333	610	F115W	0.59

TABLE 1
DATA SET STATISTICS FOR PSF CATALOG

3.2. Simulated Galaxy Catalog

There are plenty of widely-used physically realistic N-body simulations, using publically available codes such as Gadget-2/3 (Springel 2005), ART (Kravtsov et al. 1997), RAMSES (Teyssier 2002), and Enzo (Brummel-Smith et al.

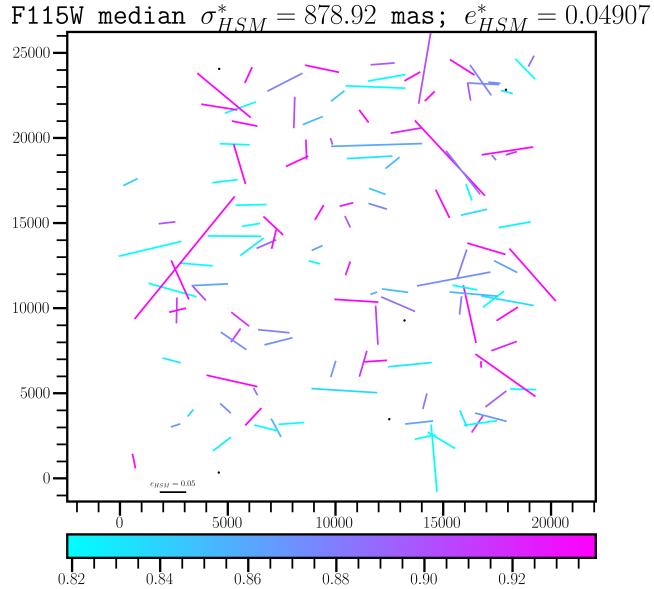


FIG. 4.— A quiverplot indicating the sizes and shape of the point sources computed via HSM adaptive moments in a single tile of the COSMOS-Web survey. The color bar at the bottom denotes the size of the source, the length of the quiver describes the magnitude of ellipticity, and the orientation and position describes its measured orientation and position on the image plane. The median size and ellipticity of point sources in terms of HSM moments is given at the top of the figure.

2019; Bryan et al. 2014), see Vogelsberger et al. (2020) for a review. However, this work aims only to have a simple differentiable forward model. Instead of using Gadget-2 or similar tools, we write our simulations from scratch in Julia and Python and time evolve objects according to ordinary differential equations from Newtonian mechanics. Our main implementation uses Jax (Bradbury et al. 2018) with Diffrax (Kidger 2021). We specify starting positions and velocities for objects $[\vec{x}, \vec{v}]$, which are then evolved via $\frac{d\vec{x}}{dt} = \vec{v}$, $\frac{d\vec{v}}{dt} = \vec{a}$, and $\vec{a} = -\frac{GM}{|\vec{x}|^3} \vec{x}$. We begin by randomly initializing galaxies with some mass M , velocity \vec{v} , and initial position \vec{r} . These galaxies exist in a gravitational field where the galaxy-galaxy gravitational attraction is assumed to be negligible compared to the attraction toward the center of the field. Thus, we can solve each ODE system in Diffrax independently and use the vmap function to efficiently time evolve each object. While our toy simulation is not very physically meaningful, this approach outputs catalogs that are differentiable with respect to the input cosmology, which in this case is just the gravitational constant G . The number of galaxies in the simulation can vary, allowing us to study the efficacy of our algorithm as a function of catalog sizes.

3.3. Intrinsic Alignment Catalog

This work uses an extensive galaxy catalog that was generated according to seven halo-based modeling parameters using the procedure of Pandya et al. (2024, 2025). The halo occupation distribution (HOD) model is used to populate existing catalogs of dark matter-only halos with galaxies (Hearin et al. 2017). The model is parameterized by five occupation components, $\log M_{\min}$, $\sigma_{\log M}$, $\log M_0$, $\log M_1$, and α , described in detail in (Zheng et al. 2007). Briefly, the occupation components parameterize the mean occupation functions given by

$$\langle N_{\text{cen}}(M) \rangle = \frac{1}{2} \left[1 + \text{erf} \left(\frac{\log M - \log M_{\min}}{\sigma_{\log M}} \right) \right] \quad (12)$$

and

$$\langle N_{\text{sat}}(M) \rangle = \frac{1}{2} \left[1 + \text{erf} \left(\frac{\log M - \log M_{\min}}{\sigma_{\log M}} \right) \right] \left(\frac{M - M_0}{M'_1} \right)^{\alpha} \quad (13)$$

where

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt. \quad (14)$$

The contributions of Van Alfen et al. (2024) introduce a model of IA within a HOD-based framework. Specifically,

the IA implementation includes a two-parameter family governing central and satellite alignment strengths (μ_{cen} and μ_{sat}). The work explores different forms of alignment strength (e.g., constant, radially dependent), and demonstrates that this parameterization can effectively model IA in both dark-matter only simulations which include galaxies via HOD modeling, and within the TNG300 suite of hydrodynamic simulations.

For each of the seven input parameters, ten realizations of the model were produced. Once the catalogs were generated, the three IA correlation functions were estimated according to Equations 9 - 11. This led to a data matrix of the shape (110, 526; 7; 100; 3; 20). In other words, we have 110, 526 tuples of seven input values that produced ten realizations of three correlation functions that each occupy twenty radial bins. This catalog is used to train the IA emulator (IAEmu) surrogate model introduced in [Pandya et al. \(2024\)](#). IAEmu is a neural network based emulator designed to predict galaxy IA statistics from a given HOD, with the alignments parameterized according to the two parameter family μ_{cen} and μ_{sat} ([Van Alfen et al. 2024](#)). These 110, 526 tuples are broken down in training, validation, and testing sets for IAEmu. This is summarized in Table 2.

Training Set	Validation Set	Testing Set	Input Parameters	Output Bins	Realizations
77,368 (70%)	11,052 (10%)	22,105 (20%)	7	20	10

TABLE 2
DATA SET STATISTICS FOR THE IA CATALOGS

4. ALGORITHM OVERVIEW

In this section, we outline our algorithms used to estimate correlations. Our algorithm has three variants, one geared toward estimating clustering uncertainties and the other two towards differentiability. Variations on the algorithm geared toward estimating clustering uncertainties are needed because estimating uncertainty requires sampling, which is incompatible with differentiation. There are a variety of ways to adapt sampling to be a differentiable process, which we explore in depth in §6. Throughout this work, we use most of the formalism laid out in [Bezdek et al. \(1984\)](#). Our formalism differs in that it uses a distance function to compute angular distances on the sky rather than using a matrix to define a norm. We also define the quantity matrix, which allows us incorporate the quantities we wish to correlate into the original formalism. For details on the fuzzy sets formalism, see [Zadeh \(1965\)](#) and (page 38, [Aluffi 2021](#)). Algorithm 1 first uses K-means++ ([Kapoor & Singhal 2017](#)) to initialize a series of cluster centers. This algorithm adds initial cluster centers sequentially by considering the distances to centers that have already been determined. This initialization has been shown to help clustering algorithms and is better than choosing initial cluster centers at random ([Kapoor & Singhal 2017](#)). We next use the fuzzy-c-means algorithm ([Bezdek et al. 1984](#)) to obtain final cluster centers and weight assignments that minimize the functional

$$J_m(U, v) = \sum_{k=1}^N \sum_{i=1}^c (U_{ik})^m \|y_k - v_i\|^2 \quad (15)$$

where

- $Y = \{y_1, \dots, y_n\}$ are the object positions,
- c is the number of clusters in Y ; $2 \leq c < n$,
- m is the weighting exponent (or fuzziness); $1 \leq m < \infty$,
- U is the fuzzy c-partition of Y ; $U \in \mathbb{R}^{c \times N}$
- $v = (v_1, \dots, v_c)$ is a vector of centers,
- $v_i = (v_{i1}, \dots, v_{in})$, is the center of cluster i .

We also define q_{ij} with $1 \leq i \leq M$ and $1 \leq j \leq N$ as the quantity matrix. Each row represents a quantity of interest that we wish to correlate and each column represents a different object. In this work, there will be two rows corresponding to two shears. Our formalism differs from [Bezdek et al. \(1984\)](#) in that the norm on $\|y_k - v_i\|^2$ is not induced by a matrix. Instead, we use the Vincenty formula given by Equation 16 to compute angular separations on the sky between two angular coordinates $(\phi_1, \lambda_1), (\phi_2, \lambda_2)$:

$$\Delta\sigma = \arctan \left(\sqrt{(\cos \phi_2 \cdot \sin \Delta\lambda)^2 + (\cos \phi_1 \cdot \sin \phi_2 - \sin \phi_1 \cdot \cos \phi_2 \cdot \cos \Delta\lambda)^2}, \sin \phi_1 \cdot \sin \phi_2 + \cos \phi_1 \cdot \cos \phi_2 \cdot \cos \Delta\lambda \right). \quad (16)$$

While this does not have a compact matrix representation, it is stable at all numerical scales (Vincenty 1975). In this way, we are sacrificing speed for stability. This design choice is discussed more in §6. The two argument arctan function is used to constrain the solution to the correct quadrant. Our usage of the Vincenty formula also assures that our distance estimates are consistent with Astropy (Robitaille et al. 2013), which also uses the Vincenty formula for this task.

Our implementation of fuzzy-c-means works as follows:

1. Initialize centers via K-means++ and weights via random initialization.

2. Update centers via

$$\hat{v}_i = \frac{\sum_{k=1}^N \left(\hat{U}_{ik} \right)^m y_k}{\sum_{k=1}^N \left(\hat{U}_{ik} \right)^m}; 1 \leq i \leq c \quad (17)$$

3. Update weights via

$$\hat{U}_{ik} = \left(\sum_{j=1}^c \left(\frac{\hat{d}_{ik}}{\hat{d}_{jk}} \right)^{\frac{2}{m-1}} \right)^{-1} \quad (18)$$

where the distances d are given by Equation 16.

4. Repeat steps 2 and 3 until either the maximum number of iterations is reached or the difference in weights between iterations is less than a user-provided tolerance and matrix norm.

After obtaining final centers v and weights U , we can assign each galaxy to a cluster center via sampling. Each row in U^T represents the normalized probability distribution that an object belongs to a cluster with center v_i . Thus, each object is assigned to a cluster via

$$M_i \sim \text{Categorical}(U_i^T). \quad (19)$$

We define the quantities of each cluster as the average of all the quantities of the objects assigned to it. Note that this is unnecessary if there are no quantities. This may arise if one is computing a position-position autocorrelation or similar. From here, we can proceed by computing all of the pairwise distances between clusters and then use a known estimator to compute the correlation in each distance bin. The whole of Algorithm 1 is recapitulated in the algorithm block below.

Algorithm 1 Probabilistic Estimation

- 1: Initialize centers v using K-Means++
- 2: Compute new centers v and weights U using fuzzy-c-means
- 3: Assign objects to clusters via $M_i \sim \text{Categorical}(U_i)$
- 4: Define new objects with centers v and quantities given by the average of all objects in the cluster
- 5: Compute all pairwise distances d between objects using Equation 16
- 6: Bin the pairwise distances by spatial separation
- 7: **for** each bin b_i **do**
- 8: Evaluate correlation estimator on binned object distance pairs
- 9: **end for**

Algorithm 2 is a differentiable analog to Algorithm 1. Algorithm 1 goes from soft bin assignments to hard assignments by sampling the rows of U . By soft assignments, we mean that for each object the assignment probabilities are spread

across the different clusters. This is in contrast to our definition of a hard assignment in §1. This transition from hard assignment to soft assignments is the first obstacle to be amended in Algorithm 2, since sampling is not a differentiable

COSMOS Field Point Sources and the Cluster Centers

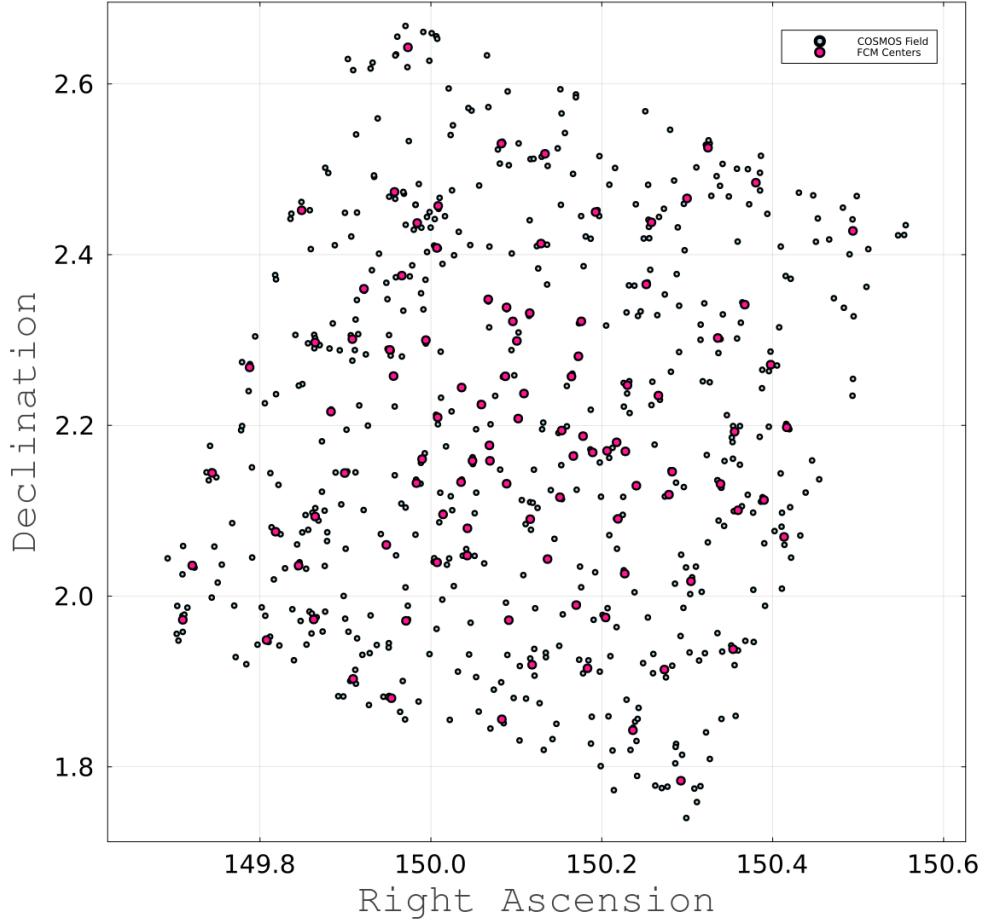


FIG. 5.— The cluster centroids found via fuzzy-c-means overlayed on the point sources of the underlying COSMOS field

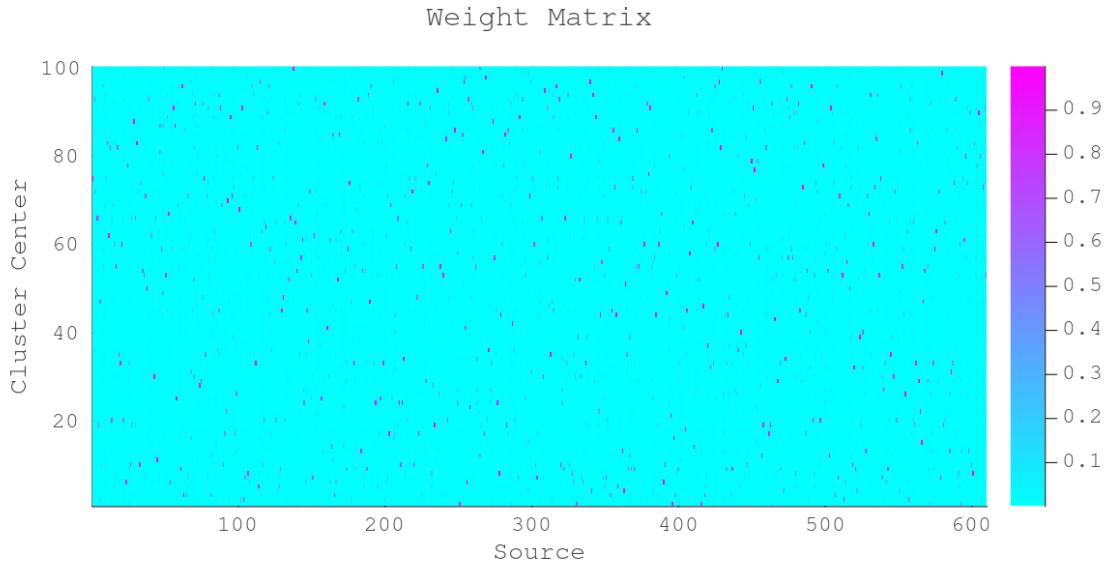


FIG. 6.— An example weight matrix U produced via fuzzy-c-means for 610 point sources in the COSMOS field and 100 cluster centers.

process. There are at least three ways to adapt this step in Algorithm 1 to become differentiable, which we will hereafter refer to as approaches a, b, and c:

- (a) Use the Gumbel Max reparameterization trick to enable differentiation through the sampling process (Maddison et al. 2016; Jang et al. 2016)¹.
- (b) Avoid sampling entirely and instead compute weighted averages to get the quantities for each cluster. The quantities are weighted by how likely they are to belong to an object in a given cluster and then averaged. This is done for each cluster.
- (c) Assign each object to a cluster with the highest membership probability via ARGMAX (effectively proceeding with hard assignments). Adapt the membership matrix such that there is a 1 in the entries U_{ij} where object j most likely belongs to cluster i . Then, make new objects by averaging the quantities in each cluster. During the gradient calculation, approximate the gradients for each object as the gradient of the new cluster object it belongs to. In other words, if object i is in cluster j , approximate the gradient of a correlation function with respect to object i as the gradient of the correlation function with respect to the new object generated from cluster j .

We explore approaches b and c because they are simple to compute via matrix multiplication. For these approaches to work, we must also normalize the rows of U . We define

$$\tilde{U}_{ij} = \frac{U_{ij}^T}{\sum_{k=1}^m U_{kj}^T}. \quad (20)$$

Now we can express our weighted average with the matrix product

$$\tilde{q} = q\tilde{U}. \quad (21)$$

We see that $\tilde{q} \in \mathbb{R}^{M \times c}$, representing the M quantities for each cluster c . Typical correlations choose M to be 2 or 3. For example, if we were computing the ξ_+ shear-shear correlation, we would have two quantities e and e^* for each cluster.

Another adaptation we need to make from algorithm 1 is the hard assignment of distance pairs to bins. Algorithm 1 takes all of the distance pairs that fall into a distance bin and runs an estimator to get the correlation value for that bin. The differentiable analog is to compute the estimator over *all* distance pairs, where the contribution of each distance pair is weighted by how likely it is to fall into a given bin with bin edges a and b . In the case of the shear-shear correlation, this extends naturally the known estimator in Equation 8. We determine the weights $w_i w_j$ using the product of two sigmoid functions, which gives high importance to distances within the bin and almost no importance to objects that are outside of it. We explore two weighting functions. The first is given by

$$W(a, x, b) = \frac{1}{1 + e^{-\alpha(x-a)}} \cdot \frac{1}{1 + e^{\alpha(x-b)}} \quad (22)$$

and the second by

$$W(a, x, b) = e^{-\alpha(x - \frac{b-a}{2})^2}. \quad (23)$$

Both weighting functions exploit the sharp decline of the exponential function to quickly decrease the weights. An example of this for the first weighting function is shown in Figure 7 for distances between 1 and 2. Algorithm 2 is summarized in the algorithm block below. Given that both weighting functions can have their decay controlled by the α parameter, we found that both weighting functions can be tuned to produce correlation outputs consistent with other packages used for estimating correlation functions.

¹ An example implementation of this can be found here: <https://github.com/cassanof/gumbel-bucket-rs>.

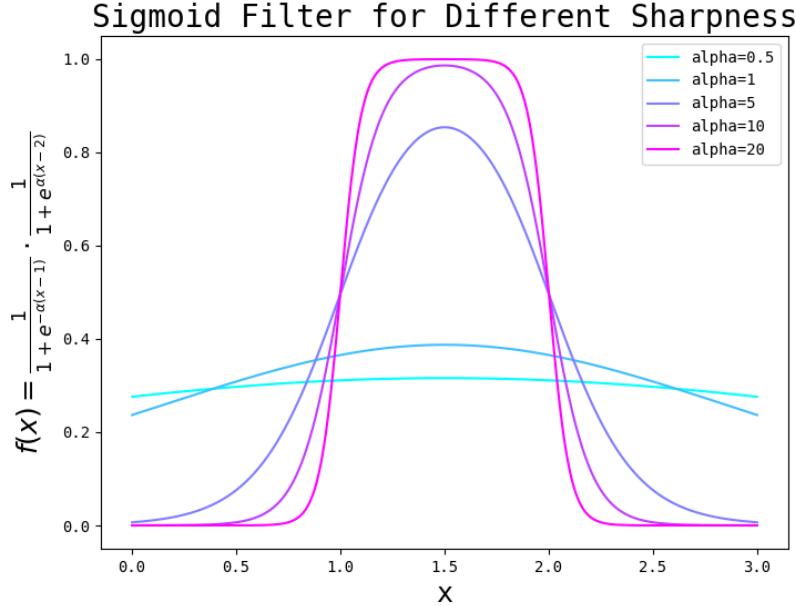


FIG. 7.— An example weighting for estimates for a distance bin bounded below by 1 and above by 2 for various sharpnesses.

Algorithm 2 Differentiable Estimation

- 1: Initialize centers v using K-Means++
- 2: Compute new centers v and weights U using fuzzy-c-means
- 3: Define new objects with centers v and quantities $q\tilde{U}$
- 4: Compute all pairwise distances d between objects using Equation 16
- 5: Compute the probability that each object pair falls in distance bin b_i by weighting by the sigmoid function in Equation 22 or the Gaussian function in Equation 23
- 6: **for** each bin b_i **do**
- 7: Evaluate correlation estimator on galaxy pairs weighted by their probability of being in bin b_i
- 8: **end for**

As we will see, there are some computational hurdles that any differentiable estimator will inevitably face that can make differentiability intractable in practice. These can be circumnavigated using a surrogate model, as outlined in Algorithm 3.

Algorithm 3 Surrogate Differentiable Estimation

- 1: Simulate realizations of a correlation function with astrophysical model parameters ϕ
- 2: Divide correlation function outputs into training, validation, and testing sets
- 3: Train a model with learnable parameters θ to learn the relationship between astrophysical model parameters and correlation functions from a (not necessarily differentiable) forward model, $\xi_\theta(\phi)$
- 4: Once trained, perform inference on parameters of choice
- 5: Optionally optimize via gradient descent the parameters ϕ that minimize some loss function $L(\xi_\theta(\phi))$
- 6: Optionally retrieve via HMC the parameter posteriors $p(\phi|\text{data})$ with priors $p(\phi)$ and a likelihood $p(\text{data}|\phi)$

5. MODEL UNCERTAINTY

In this section, we discuss the results of Algorithm 1 on our point source catalog dataset, which we compare with a bootstrapping baseline to obtain epistemic and aleatoric uncertainties.

Figure 5 shows the cluster centers overlayed with the objects on the sky. The result indicates that the cluster centers reasonably approximate the spatial variation of the sources in the underlying COSMOS field. Figure 6 shows the weight matrix after the objects are clustered with fuzzy-c-means.

Figure 9 shows the Shannon information entropy and maximum probability of each row in the weight matrix. While the majority of objects have a clear preference toward being assigned to a specific cluster, there is a non-trivial number of objects that could be reasonably assigned to multiple clusters. This indicates that the model is often not sure where the best assignment of an object to a cluster is. For this reason, we should expect the error bar associated with model uncertainty to be comparable to data uncertainty. This bears out in Figure 8 and Figures 14 - 17.

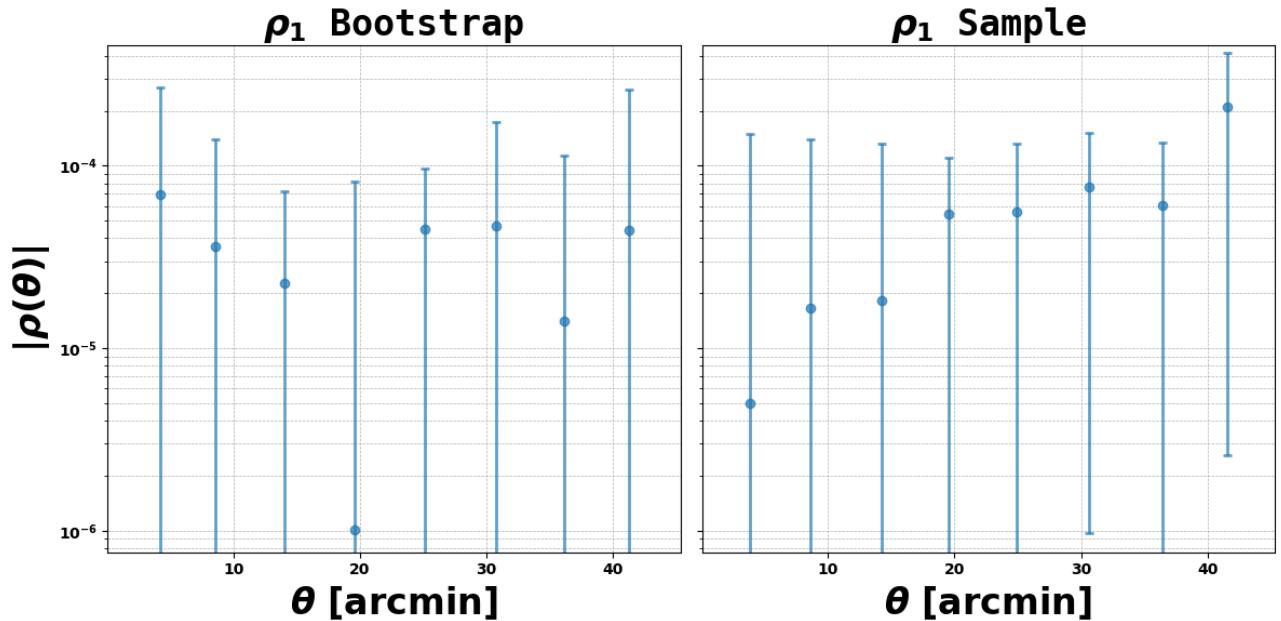


FIG. 8.— ρ_1 correlation function with data taken from the COSMOS-Web point source catalogs. Error bars drawn from bootstrap (left) and sampling (right) approaches. Negative correlations are shown in absolute value, and correlations are plotted in logarithmic scale.

We note that increasing or decreasing the number of clusters did not significantly reduce the size of the model uncertainty. This indicates that the source density in our COSMOS-Web data set is not large enough to beat down the errors associated with clustering. Recall that in §2 we said that we are treating uncertainties as additive. Figures 8 and 14 - 17 therefore indicate that in many cases the epistemic uncertainties doubles the uncertainty compared to bootstrapping alone. This is especially true for the ρ_1 , ρ_2 , and ρ_4 statistics. For this reason, we suggest a “naïve”² approach to computing ρ statistics, wherein no clusters are used and the correlation estimators are used directly on the catalogs. Given that we have on the order of ~ 500 objects, the number of pairwise distances is not too large to store and the number of distance calculations to compute is not prohibitively expensive.

6. DIFFERENTIABLE FORWARD MODEL

In this section we outline the utility of algorithm 2 for gradient-based optimization. To do this, we create a simulated catalog of galaxies moving in a gravitational field. We time evolve the objects until some terminus T_{\max} , which was found by considering an object undergoing uniform circular motion and calculating how long it would take to complete one orbit. For simplicity, we analyze correlations across three linearly spaced angular separation bins. A max separation of 160 arcmins is chosen to ensure each of the three bins have enough samples. We seek to differentiate the shear-shear correlation $\xi_+(\theta)$ with respect to the gravitational constant G at each angular separation bin θ_i , illustrating an example of enabling differentiability of a correlation with respect to an astrophysical model parameter.

² The use of the word naïve is in reference to its usage in [Jarvis et al. \(2004\)](#).

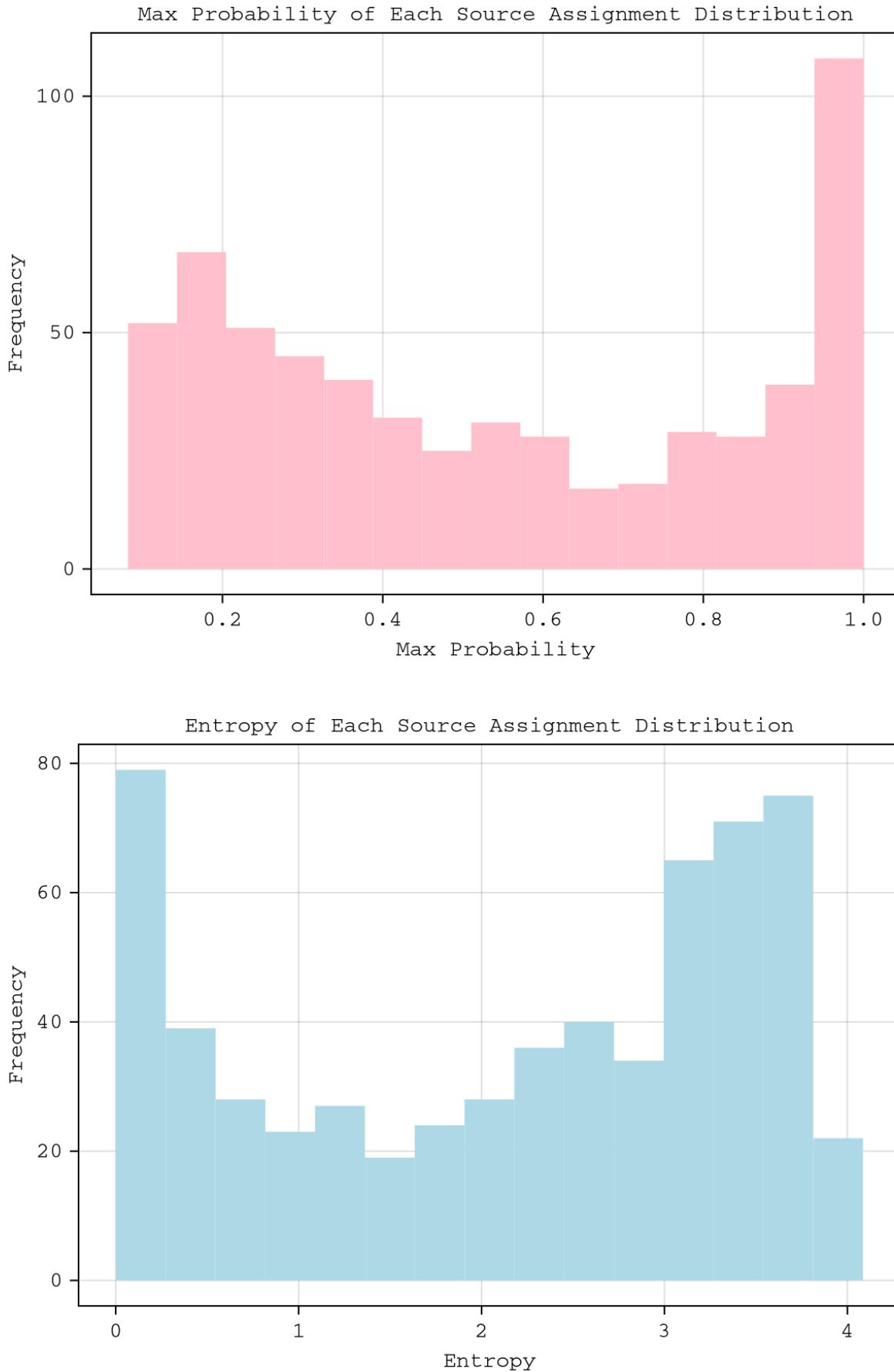


FIG. 9.— The maximum probability and entropy of the source assignment distributions. Each column in U corresponds to a probability distribution. For each of those distributions, we compute the maximum probability and Shannon entropy.

The ellipticities are predetermined, but the final positions are naturally dependent on the underlying physics. Since the catalog is produced by solving a series of ordinary differential equations (ODEs), computing the derivative just amounts to using the chain rule $\frac{\partial \xi}{\partial G} = \sum_{i=1}^N \frac{\partial \xi}{\partial \alpha_i} \frac{\partial \alpha_i}{\partial G} + \frac{\partial \xi}{\partial \delta_i} \frac{\partial \delta_i}{\partial G}$. Here, α denotes the right ascension and δ denotes the declination coordinates. We use the ODE solvers and automatic differentiation tools implemented in Julia (Bezanson et al. 2017). Specifically, we use the `OrdinaryDifferentialEquations.jl` (Rackauckas & Nie 2017) library for working with differential equations. We also tested both reverse mode automatic differentiation with `Zygote.jl` (Innes et al. 2019) and forward mode automatic differentiation with `ForwardDiff.jl` (Revels et al. 2016). The latter proved to be more effective. We also built an implementation with `Jax` (Bradbury et al. 2018) and `Diffrax` (Kidger 2021), which is available on our GitHub repository. Our adoption of these tools is motivated by the discussion of Berman & Ginesin (2024).

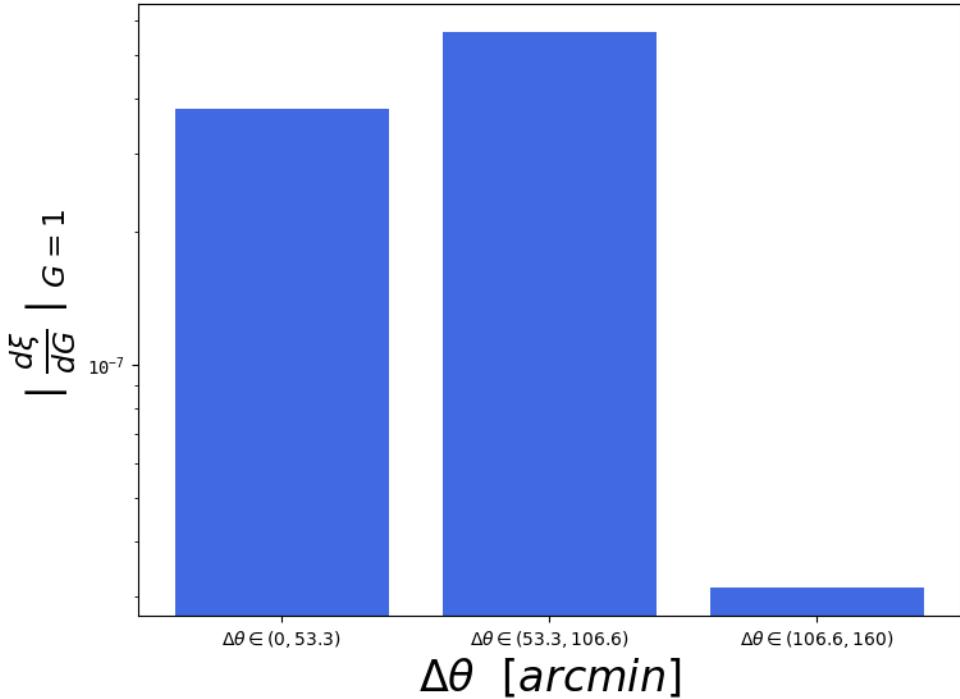


FIG. 10.— The gradient of the shear-shear correlation function ξ with respect to the input model parameter G at $G = 1$.

For both approaches b and c (as defined in §4), the naive approach to computing the gradient $\frac{\partial \xi}{\partial \alpha_i}$ or $\frac{\partial \xi}{\partial \delta_i}$ would involve tracking the computation graph through the entire fuzzy-c-means algorithm, which can sometimes take over 100 iterations. Another complication is that the distance calculations and normalization computations cannot be expressed through matrix multiplication, making it difficult to accelerate the gradient calculations with a GPU. For approach *b*, taking the gradients in this manner proved to be prohibitively expensive via reverse mode automatic differentiation, for the reasons outlined above. With forward mode differentiation, the gradient calculation was on the order of a minute. This is expected, since we are differentiating several output bins with respect to 1 input (gravity).

One workaround to the computation time bottleneck is to pre-compute the final centers and weights with fuzzy-c-means, then update the weights and centers for only a single iteration. This makes the final clustering a function of the data without keeping the entire history of the clustering procedure in the computation graph. Using this “trick” to approximate the true gradient, we can roughly halve the computation time needed to solve for the derivative $\frac{\partial \xi}{\partial G}$. However, when using approach *b*, issues related to precision still exist. When using a large number of clusters, we found that the Jacobian relating the dependence of weights on the input data was sparse. The sparsity result is expected, as local perturbations of input positions, by design, preferentially impact the assignment probabilities of nearby cluster centers. As a result, we are left with vanishing gradients, and the algorithm becomes unstable as we scale the number of clusters, producing NaN outputs. We note that these NaN issues did not occur on randomly computed inputs used for testing the differentiability of each individual function, indicating that the reliability of this method will vary for problems of different conditioning. This also confirms that while each individual step in the algorithm is differentiable,

instabilities can still occur from gradient multiplication. It is worth noting that multiplying a matrix with another matrix that is sparse does not mean the resulting matrix is necessarily sparse. Thus, one Jacobian *alone* being sparse is not the problem, rather, its the effect of a sparse Jacobian when propagated through the chain rule. Because we are reasoning about gradients, we want to ensure our representations are dense. We note that this is in contrast to other numerical problems where sparsity is necessary to maintain computational efficiency when dealing with large matrices (chapter 10 of [Saad 2003](#); [Rosen 2022](#)).

Approach *c* is able to circumnavigate some of these issues. As evidenced by Figure 10, we are successfully able to differentiate the outputs of our correlation function in each distance bin with respect to the input astrophysical model parameter G at the point $G = 1$. However, issues with scaling the number of data points still persist, as the computation becomes too large to run on a CPU well before 500 galaxies are simulated via the forward model.

The precision and computation time issues suggest that the simple solution is not always sufficient for enabling the differentiability of correlation functions with respect to the underlying model inputs. Surrogate models, however, have demonstrated the ability to successfully emulate correlation function estimators ([Van Alfen et al. 2024](#); [Pandya et al. 2024, 2025](#)). These surrogate models can be as simple as linear regression, or more involved functions such as those represented by a neural network (NN). NNs, by definition, are differentiable models (i.e. trained with gradient descent) and enjoy GPU accelerated computations.

7. DIFFERENTIABLE SURROGATES

In this section, we show how Algorithm 3 can be used to differentiate through a correlation function and present a compelling science case that supports its utility. In particular, we utilize the surrogate model developed in [Pandya et al. \(2024, 2025\)](#), IAEmu, which estimates galaxy IA correlations from an underlying set of HOD parameters. We use IAEmu to optimize model parameters that minimize the IA correlations, to study regions of parameter space that correspond to minimal IA contamination and are thus of physical interest. IAEmu serves as an emulator for Halotools-IA, which extends traditional HOD modeling to include IA information through a two parameter family, μ_{cen} and μ_{sat} , which govern central and satellite alignment strengths, respectively. More specifically, IAEmu outputs the galaxy position-position (ξ), position-orientation (ω), and orientation-orientation (η) correlations as well as estimates of the aleatoric uncertainty (shape noise) of each correlation. This noise is not physical, and corresponds to variance across realizations of the underlying HOD due to limited cosmological volumes when conducting the HOD simulations.

We pose this optimization problem because IA often act as a contaminant for weak lensing measurements, and moreover, there are known regions of parameter space that lead to low correlations. Thus, with this experiment, we are able to demonstrate that a differentiable forward model allows one to learn something about the underlying physics of IA. In other words, we are able to recover a known parameter space even without hard-coding the physical laws into the surrogate directly. This is distinct from [Pandya et al. \(2025\)](#), who use the emulator to find the IA parameters that best explain a set of observations rather than the parameters that result in the minimized correlation function.³

The surrogate model introduced in [Pandya et al. \(2024, 2025\)](#) is a deep neural network with a multilayer perceptron encoder and three convolutional neural network decoder heads for each of the three IA correlation functions. The model is also constructed to predict aleatoric uncertainties per-bin for each correlation, and epistemic uncertainties via the Monte Carlo dropout technique ([Hüllermeier & Waegeman 2021](#)). For a single initialization of the seven Halotools-IA parameters, Figure 11 shows all three IA correlation functions. Our study focuses exclusively on the $\omega(r)$ correlation. This is because $\xi(r)$ is akin to galaxy clustering and therefore does not contain any IA information itself. In addition, the galaxy shape noise that is present in $\eta(r)$ significantly increases the difficulty of extracting relevant signal for gradient-based optimization pipelines, as seen in Figure 11. This leaves us with $\omega(r)$, as it contains IA information but is significantly less obscured by shape noise when compared to $\eta(r)$. This is seen quantitatively in Figure 18, which shows the Jacobian of each correlation as a function of the model inputs. It is seen that the gradient magnitudes of IAEmu are in general larger than that of η . These gradients are also stable from bin-to-bin, and can further be accessed in less than a second. While there may be some disagreement with the exact sensitivity that the Halotools-IA⁴ model presented in [Van Alfen et al. \(2024\)](#) would predict, we will see that the gradients information from IAEmu further affirms that it is an accurate emulator for Halotools-IA.

To isolate regions of parameter space in μ_{cen} and μ_{sat} that minimize the IA correlation $\omega(r)$, we utilize Hamiltonian Monte Carlo (HMC). HMC is a gradient based Monte Carlo technique which leverages the differentiability of NNs and uses gradient information to efficiently traverse the input parameter space towards regions that maximize the

³ The exact correlation that would be subtracted is the gravitationally lensed shape - intrinsic shape (GI) correlation ([Lamman et al. 2023](#)); however, the strength of that correlation is closely related to that of ω .

⁴ <https://github.com/astropy/halotools>

posterior probability. For a general review of HMC, see [Betancourt \(2017\)](#). Our HMC experiment mirrors [Pandya et al. \(2025\)](#). We run our HMC with a No U-Turn Sampler (NUTS) ([Hoffman et al. 2014](#)) with 1000 warmup steps and 1000 samples. We fix the HOD parameters according to the fiducial values present in IllustrisTNG300-1 ([Nelson et al. 2019](#)) and Table C1 of [Van Alfen et al. \(2024\)](#). This gives us three sets of HOD parameters, which each correspond to a different stellar mass cutoff in galaxies. We impose uniform priors of $[-1, 1]$ on μ_{cen} and μ_{sat} and use a multivariate normal distribution centered at zero as our likelihood, with a covariance given by the IAEmu aleatoric uncertainty prediction. In this way, we encourage HMC to arrive at regions of parameter space that minimize the ω correlation, while also incorporating the inherent shape noise into the likelihood.

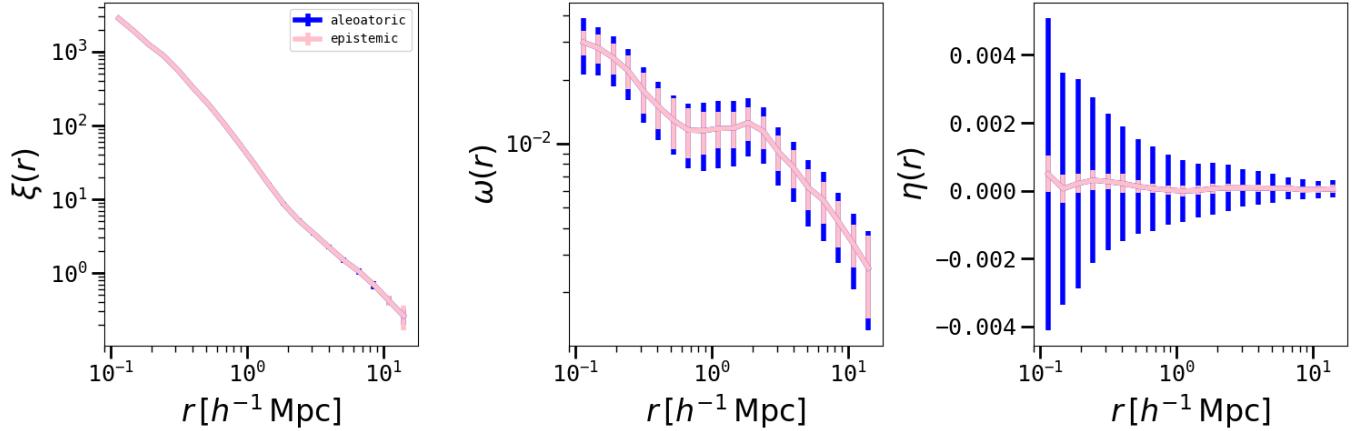


FIG. 11.— IAEmu predictions for a given initialization of the seven halo-based modeling parameters, including $\xi(r)$ on the left, $\omega(r)$ in the middle, and $\eta(r)$ on the right. For each of the twenty radial bins, the value of the correlation, as well as aleatoric and epistemic uncertainties are provided. Log-log scaling is used to better see the dynamic range of the correlations and radial dependence.

Figures 12 and 13 confirm that μ_{cen} and μ_{sat} values of zero minimize ω , with some degeneracy as indicated by the diagonal shape of the posterior. While values of $\mu_{\text{cen}} = 0 = \mu_{\text{sat}}$ are likely to minimize the correlation function, regions where the signs of μ_{cen} and μ_{sat} are opposite can also result in low correlations. Physically, this corresponds to scenarios wherein the negative (i.e. perpendicular) alignments of satellite galaxies can oppose the positive (i.e. parallel) alignment of centrals, resulting in an overall small value of $\omega(r)$.

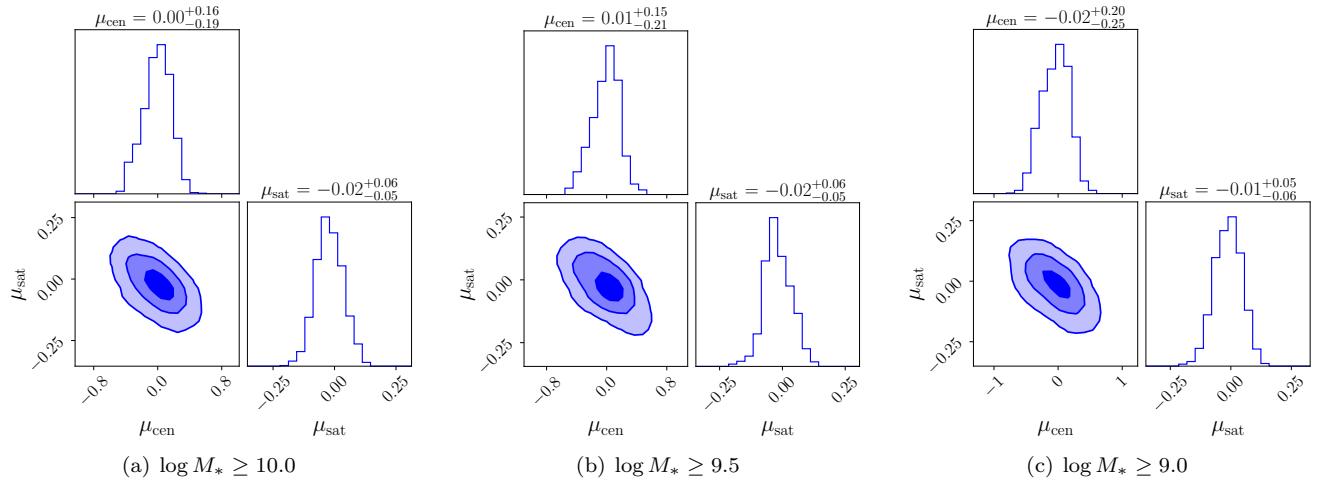


FIG. 12.— Posterior constraints on μ_{cen} and μ_{sat} obtained via HMC for three different sets of HOD parameters. Each set roughly corresponds to a different stellar mass cutoff in galaxies. We show results for $\log M_* \geq 10.0$ on the left, $\log M_* \geq 9.5$ in the middle, and $\log M_* \geq 9.0$ on the right. Contours represent 1σ , 2σ , and 3σ confidence intervals.

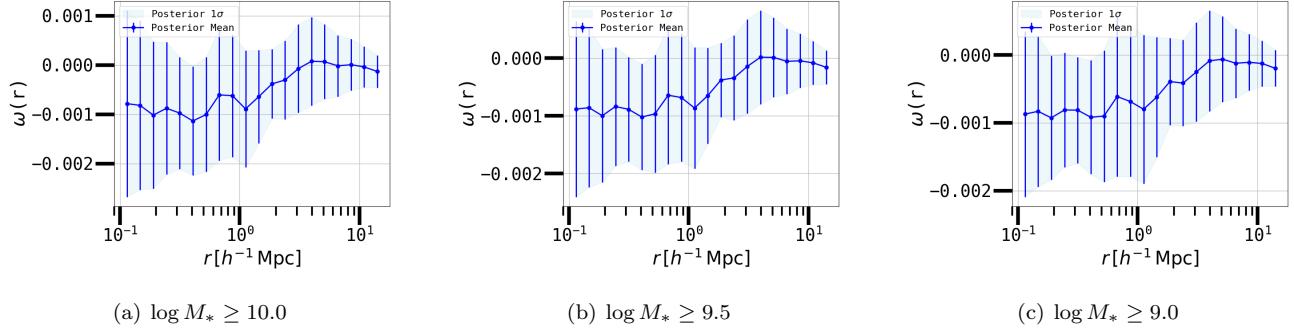


FIG. 13.— Correlation functions sampled from the posterior distributions presented in 12. The solid represents the average correlation value in all bins from the 1000 samples of (μ_{cen}, μ_{sat}) pairs obtained via HMC. The error bar represents the standard deviation of the correlation value for that same sample.

8. SUMMARY AND CONCLUSIONS

In this work, we studied correlation functions along three axes: model uncertainty, differentiable forward models, and differentiable surrogates. To better understand the model uncertainty caused by clustering approximations, we used ρ statistics as a case study. We made our assignment of objects to clusters probabilistic, and computed the mean and standard deviation of the resulting correlation estimates. Our analysis compared the uncertainty estimate to traditional bootstrapping methods. Our next experiment adapted our main algorithm to be automatically differentiable and applied this to the setting of gravitational simulations. Three different approaches of doing this were proposed, and of those, approach c proved to be the most stable. Still, the calculation of the gradient was slow and approximate, motivating the use of surrogate methods. To that end, we ended by exploring surrogate solutions. We extended the work of Pandya et al. (2024, 2025), showing how we can exploit the gradients of our surrogate model to optimize over astrophysical model parameters. Specifically, we studied galaxy intrinsic alignment, and found which astrophysical model parameters were most responsible for high / low correlations (Figures 12-13).

On uncertainty, it was found that the process of clustering objects before computing correlations can cause measurable uncertainties in limited data settings, such as with the COSMOS-Web PSF modeling efforts. That is, the model uncertainty was just as severe as the data uncertainties found via bootstrapping. In these circumstances, we conclude that the naïve approach, which considers all distance pairs without clustering to be most appropriate. Moreover, we find that our method of probabilistic clustering allows us to address one of the fundamental concerns in uncertainty quantification, that being the distinction between epistemic and aleatoric uncertainties. Again, this is quantified in Figures 8-17, where the error bars for epistemic and aleatoric uncertainties are shown for each of the five ρ statistics.

On differentiability, we showed that the simplest differentiable algorithm has features that makes the computation of the gradients slow and unstable. These findings were demonstrated through attempts to differentiate a correlation function with respect to a single parameter G that determined the results of a gravitational simulation. Three approaches, a-c, were outlined: Approach a relied on the Gumbel-Max trick, approach b used the weight matrix to compute new quantities via weighted averages, and approach c used an approximate method to calculate the gradient. Of these, approach c proved to be the most effective at circumnavigating these deficiencies. The features that caused slowness and instability were the normalization and distance function calls and sparse Jacobians that propagated through the chain rule. The normalization and distance function calls posed challenges for fast automatic differentiation as they cannot be expressed simply with matrix multiplication; the normalization function calls in particular are notorious bottlenecks for automatic differentiation pipelines. We also saw that many of our function calls yielded sparse Jacobians. Since the chain rule amounts to multiplying successive Jacobians, this caused numerical instabilities in computing the final derivative. It is worth noting that multiplying one matrix by a sparse matrix does not necessarily imply that the resulting matrix is sparse. However, through testing the function calls individually, we found that the numerical instabilities happened only once the gradients were multiplied together, indicating that matrix sparsity patterns in any one Jacobians is the culprit for numerical instability. This also explained why approach c was most successful, as approach c avoided the function call that related weights to the input data when taking the gradient. Approach c was successfully able to capture the gradient of a correlation function with respect to an input cosmology. While there remains future work in optimizing approach c for GPU support and overall making it faster, it proved to be the most effective way to write down correlation function estimators that are automatically differentiable. In the main, the differentiability experiment suggested that a more judicious way to construct differentiable correlation functions may be to use a surrogate model.

On surrogates, we found that surrogate models are a simple and effective way to optimize astrophysical model parameters that determine a correlation function. Using galaxy IA as an example, a correlation function was minimized across 20 radial bins. Using HMC, we were able to recover a known posterior over the IA parameters that lead to low correlation function outputs. A key caveat is that the true correlation functions that act as a contaminant are the GI and gI correlations (as defined in [Lamman et al. 2023](#)), but the $\omega(r)$ correlation is closely related in how it represents the presence of IA.

Prior to our work, properties such as model uncertainty and differentiability have been understudied in the context of correlation function estimators. This work is intended as a first step toward addressing these points. Our study is intentionally generic — survey systematics can vary tremendously and the correlations being studied will also vary by science case. Rather than focusing on a specific scientific inquiry, we propose general algorithms and analyze their utility across different domains. Our GitHub artifact outlines and implements these algorithms, serving as a blueprint for future large-scale surveys in each of the contexts laid out in this work — model uncertainty, differentiability, and surrogates.

ACKNOWLEDGEMENTS

E.B. thanks Quinn Arbolante, [Jacob Ginesin](#), [Luisa Li](#), Eric Zelikman for helpful discussions. E.B. is supported by a College of Science Dean’s Office Research grant. E.B. also thanks Gregory Shomo for helpful guidance with the discovery computing cluster. S.P. acknowledges support from the National Science Foundation under Cooperative Agreement PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, <https://iaifi.org>). Support for the COSMOS-Web survey was provided by NASA through grant JWST-GO-01727 and HST-AR-15802 awarded by the Space Telescope Science Institute, which is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS 526555. This work was made possible by utilizing the CANDIDE cluster at the Institut d’Astrophysique de Paris, which was funded through grants from the PNCG, CNES, DIM-ACAV, and the Cosmic Dawn Center and maintained by Stephane Rouberol. Further support was provided by Research Computing at Northeastern University.

REFERENCES

Aluffi, P. 2021, Algebra: chapter 0, Vol. 104 (American Mathematical Soc.)

Beichman, C. A., Rieke, M., Eisenstein, D., et al. 2012, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 8442, Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave, ed. M. C. Clampin, G. G. Fazio, H. A. MacEwen, & J. Oschmann, Jacobus M., 84422N, doi: [10.1117/12.925447](https://doi.org/10.1117/12.925447)

Berman, E., & Ginesin, J. 2024, arXiv preprint arXiv:2410.10908

Berman, E., & McCleary, J. 2024, Journal of Open Source Software, 9, 6144, doi: [10.21105/joss.06144](https://doi.org/10.21105/joss.06144)

Berman, E., & McCleary, J. 2025, in American Astronomical Society Meeting Abstracts, Vol. 245, American Astronomical Society Meeting Abstracts, 424.01

Berman, E. M., McCleary, J. E., Koekemoer, A. M., et al. 2024, The Astronomical Journal, 168, 174, doi: [10.3847/1538-3881/ad6a0f](https://doi.org/10.3847/1538-3881/ad6a0f)

Bertin, E., & Arnouts, S. 1996, Astronomy and astrophysics supplement series, 117, 393

Betancourt, M. 2017, arXiv preprint arXiv:1701.02434

Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. 2017, SIAM review, 59, 65

Bezdek, J. C., Ehrlich, R., & Full, W. 1984, Computers & geosciences, 10, 191

Birrer, S., Shajib, A. J., Gilman, D., et al. 2021, Journal of Open Source Software, 6, 3283, doi: [10.21105/joss.03283](https://doi.org/10.21105/joss.03283)

Bradbury, J., Frostig, R., Hawkins, P., et al. 2018, JAX: composable transformations of Python+NumPy programs, 0.3.13. [http://github.com/jax-ml/jax](https://github.com/jax-ml/jax)

Brummel-Smith, C., Bryan, G., Butsky, I., et al. 2019, Journal of Open Source Software, 4, 1636

Bryan, G. L., Norman, M. L., O'Shea, B. W., et al. 2014, The Astrophysical Journal Supplement Series, 211, 19

Campagne, J.-E., Lanusse, F., Zuntz, J., et al. 2023, arXiv preprint arXiv:2302.05163

Casey, C. M., Kartaltepe, J. S., Drakos, N. E., et al. 2023, The Astrophysical Journal, 954, 31

Chantada, A. T., Landau, S. J., Protopapas, P., Scóccola, C. G., & Garraffo, C. 2023, Phys. Rev. D, 107, 063523, doi: [10.1103/PhysRevD.107.063523](https://doi.org/10.1103/PhysRevD.107.063523)

Cuesta-Lazaro, C., & Mishra-Sharma, S. 2024, Physical Review D, 109, 123531

Dark Energy Survey Collaboration. 2024, DES Y3 PIFF Data Release, <https://des.ncsa.illinois.edu/releases/y3a2/Y3piff>

Dienes, K. R., Heurtier, L., Huang, F., et al. 2022, Physical Review D, 105, 023530

Feng, B. Y., Ferrer-Chávez, R., Levis, A., et al. 2025, arXiv preprint arXiv:2501.01912

Freedman, W. L., Madore, B. F., Jang, I. S., et al. 2024, arXiv preprint arXiv:2408.06153

Halverson, J., & Pandya, S. 2024, Phys. Rev. D, 110, 075041, doi: [10.1103/PhysRevD.110.075041](https://doi.org/10.1103/PhysRevD.110.075041)

Hearin, A. P., Campbell, D., Tollerud, E., et al. 2017, The Astronomical Journal, 154, 190, doi: [10.3847/1538-3881/aa859f](https://doi.org/10.3847/1538-3881/aa859f)

Hirata, C., & Seljak, U. 2003, Monthly Notices of the Royal Astronomical Society, 343, 459

Hoffman, M. D., Gelman, A., et al. 2014, J. Mach. Learn. Res., 15, 1593

Hüllermeier, E., & Waegeman, W. 2021, Machine learning, 110, 457

Innes, M., Edelman, A., Fischer, K., et al. 2019, arXiv preprint arXiv:1907.07587

Jagvaral, Y., Lanusse, F., & Mandelbaum, R. 2024, arXiv preprint arXiv:2409.18761

Jang, E., Gu, S., & Poole, B. 2016, arXiv preprint arXiv:1611.01144

Jarvis, M., Bernstein, G., & Jain, B. 2004, Monthly Notices of the Royal Astronomical Society, 352, 338

Jarvis, M., & collaborators. 2024, TreeCorr: A python package for fast two-point correlation functions, <https://github.com/rmjarvis/TreeCorr>

Jarvis, M., Bernstein, G., Amon, A., et al. 2021, Monthly Notices of the Royal Astronomical Society, 501, 1282

Kapoor, A., & Singhal, A. 2017, in 2017 3rd international conference on computational intelligence & communication technology (CICT), IEEE, 1–6

Kidger, P. 2021, PhD thesis, University of Oxford

Kilbinger, M. 2015, Reports on Progress in Physics, 78, 086901

Kravtsov, A. V., Klypin, A. A., & Khokhlov, A. M. 1997, The Astrophysical Journal Supplement Series, 111, 73

Lamman, C., Tsaprazi, E., Shi, J., et al. 2023, arXiv preprint arXiv:2309.08605

Landy, S. D., & Szalay, A. S. 1993, Astrophysical Journal, Part 1 (ISSN 0004-637X), vol. 412, no. 1, p. 64–71., 412, 64

Lanusse, F., et al. 2023, Publications of the Astronomical Society of Australia, 40, e001

Lanzieri, D., Lanusse, F., Modi, C., et al. 2023, Astronomy & Astrophysics, 679, A61

Maddison, C. J., Mnih, A., & Teh, Y. W. 2016, arXiv preprint arXiv:1611.00712

Mandelbaum, R., Hirata, C. M., Seljak, U., et al. 2005, Monthly Notices of the Royal Astronomical Society, 361, 1287

McCleary, J. E., Everett, S. W., Shaaban, M. M., et al. 2023, The Astronomical Journal, 166, 134, doi: [10.3847/1538-3881/ace7ca](https://doi.org/10.3847/1538-3881/ace7ca)

Michalewicz, K., Millon, M., Dux, F., & Courbin, F. 2023, Journal of Open Source Software, 8, 5340, doi: [10.21105/joss.05340](https://doi.org/10.21105/joss.05340)

Nelson, D., Springel, V., Pillepich, A., et al. 2019, Computational Astrophysics and Cosmology, 6, 1

Osband, I., Wen, Z., Asghari, S. M., et al. 2023, Advances in Neural Information Processing Systems, 36, 2795

Pandya, S., Yang, Y., Alfen, N. V., Blazek, J., & Walters, R. 2025, IAEmu: Learning Galaxy Intrinsic Alignment Correlations. <https://arxiv.org/abs/2504.05235>

Pandya, S., Yang, Y., Van Alfen, N., Blazek, J., & Walters, R. 2024

Perrin, M. D., Sivaramakrishnan, A., Lajoie, C.-P., et al. 2014, in Space telescopes and instrumentation 2014: optical, infrared, and millimeter wave, Vol. 9143, SPIE, 1174–1184

Rackauckas, C., & Nie, Q. 2017, Discrete and continuous dynamical systems. Series B, 22, 2731

Revels, J., Lubin, M., & Papamarkou, T. 2016, arXiv preprint arXiv:1607.07892

Rieke, M. J., Kelly, D., & Horner, S. 2005, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 5904, Cryogenic Optical Systems and Instruments XI, ed. J. B. Heaney & L. G. Burriesci, 1–8, doi: [10.1117/12.615554](https://doi.org/10.1117/12.615554)

Rieke, M. J., Baum, S. A., Beichman, C. A., et al. 2003, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 4850, IR Space Telescopes and Instruments, ed. J. C. Mather, 478–485, doi: [10.1117/12.489103](https://doi.org/10.1117/12.489103)

Robert, C. P., Casella, G., Robert, C. P., & Casella, G. 2004, Monte Carlo statistical methods, 267

Robitaille, T. P., Tollerud, E. J., Greenfield, P., et al. 2013, Astronomy & Astrophysics, 558, A33

Rosen, D. M. 2022, IEEE Robotics and Automation Letters, 7, 12507

Rowe, B. 2010, Monthly Notices of the Royal Astronomical Society, 404, 350

Saad, Y. 2003, Iterative methods for sparse linear systems (SIAM)

Schneider, P., Kochanek, C., & Wambsganss, J. 2006, Gravitational lensing: strong, weak and micro: Saas-Fee advanced course 33, Vol. 33 (Springer Science & Business Media)

Schneider, P., van Waerbeke, L., Kilbinger, M., & Mellier, Y. 2002, Astronomy & Astrophysics, 396, 1

Scognamiglio, D. 2024, Bulletin of the AAS, 56

Springel, V. 2005, Monthly notices of the royal astronomical society, 364, 1105

Teyssier, R. 2002, Astronomy & Astrophysics, 385, 337

Van Alfen, N., Campbell, D., Blazek, J., et al. 2024, The Open Journal of Astrophysics, 7

Vincenty, T. 1975, Survey review, 23, 88

Vogelsberger, M., Marinacci, F., Torrey, P., & Puchwein, E. 2020, Nature Reviews Physics, 2, 42

Vogelsberger, M., Zavala, J., Cyr-Racine, F.-Y., et al. 2016, Monthly Notices of the Royal Astronomical Society, 460, 1399

Zadeh, L. A. 1965, Information and control, 8, 338

Zheng, Z., Coil, A. L., & Zehavi, I. 2007, The Astrophysical Journal, 667, 760, doi: [10.1086/521074](https://doi.org/10.1086/521074)

APPENDIX

PSF MODELING

This analysis uses ShOpt.jl for PSF modeling because it was shown to be accurate and computationally fast in [Berman et al. \(2024\)](#); [Berman & McCleary \(2024\)](#). Since we are expanding our input catalog to the entire survey area of the COSMOS field instead of individual tiles as in [Berman et al. \(2024\)](#), we also expect astrometric distortions to be more severe (cf. Figure 4 in [Berman et al. \(2024\)](#)). Since ShOpt works in astrometric coordinates and is compatible with PSF cutout sizes needed for NIRCam, ShOpt is well-suited for this task. In contrast, PSFex exhibits bias over large survey areas ([Jarvis et al. 2021](#)) and PIFF does not handle cutout sizes needed for characterizing the NIRCam PSF gracefully ([Berman et al. 2024](#)). Tools like WebbPSF ([Perrin et al. 2014](#)) were not designed for the mosaiced images we are working with, making their application to our data non trivial. Other tools like STARRED ([Michalewicz et al. 2023](#)) or PSFr (Birrer et al. in prep, [Birrer et al. 2021](#)) have been applied to JWST data and could potentially also be applied to this data set. See also [Feng et al. \(2025\)](#).

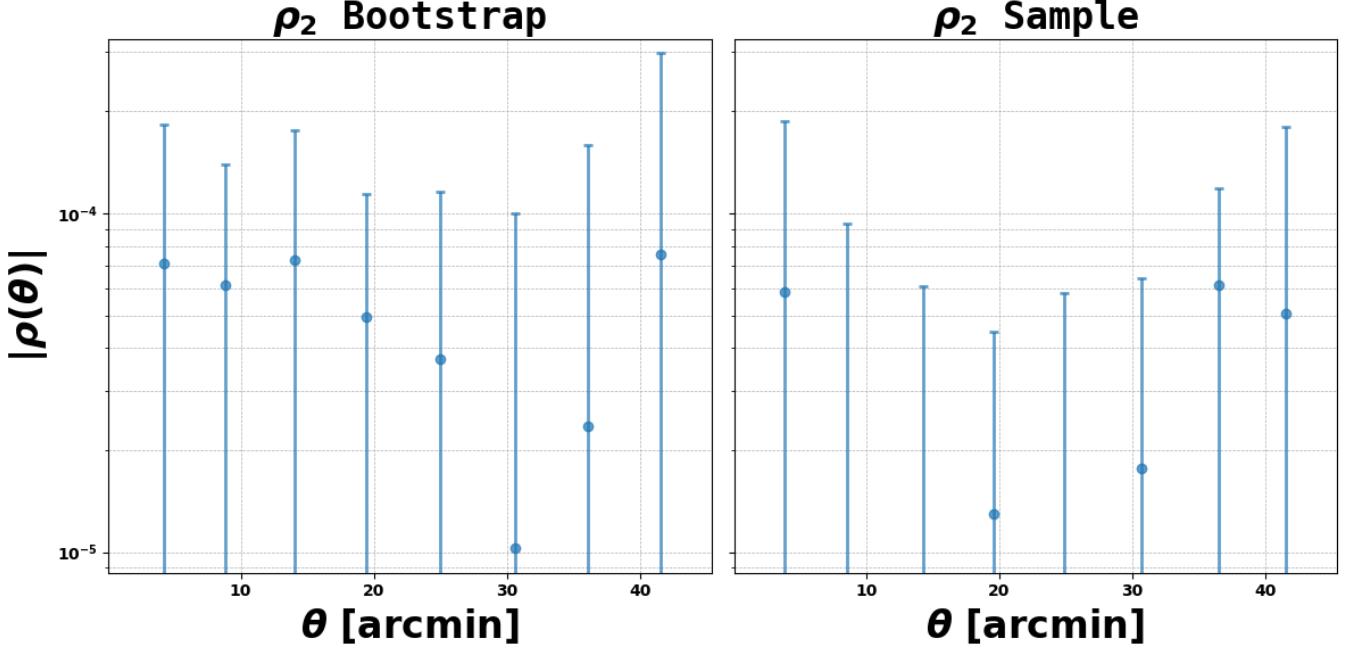
ρ STATISTICS

FIG. 14.— ρ_2 correlation function with data taken from the COSMOS-Web point source catalogs. Error bars drawn from bootstrap (left) and sampling (right) approaches. Negative correlations are shown in absolute value, and correlations are plotted in logarithmic scale.

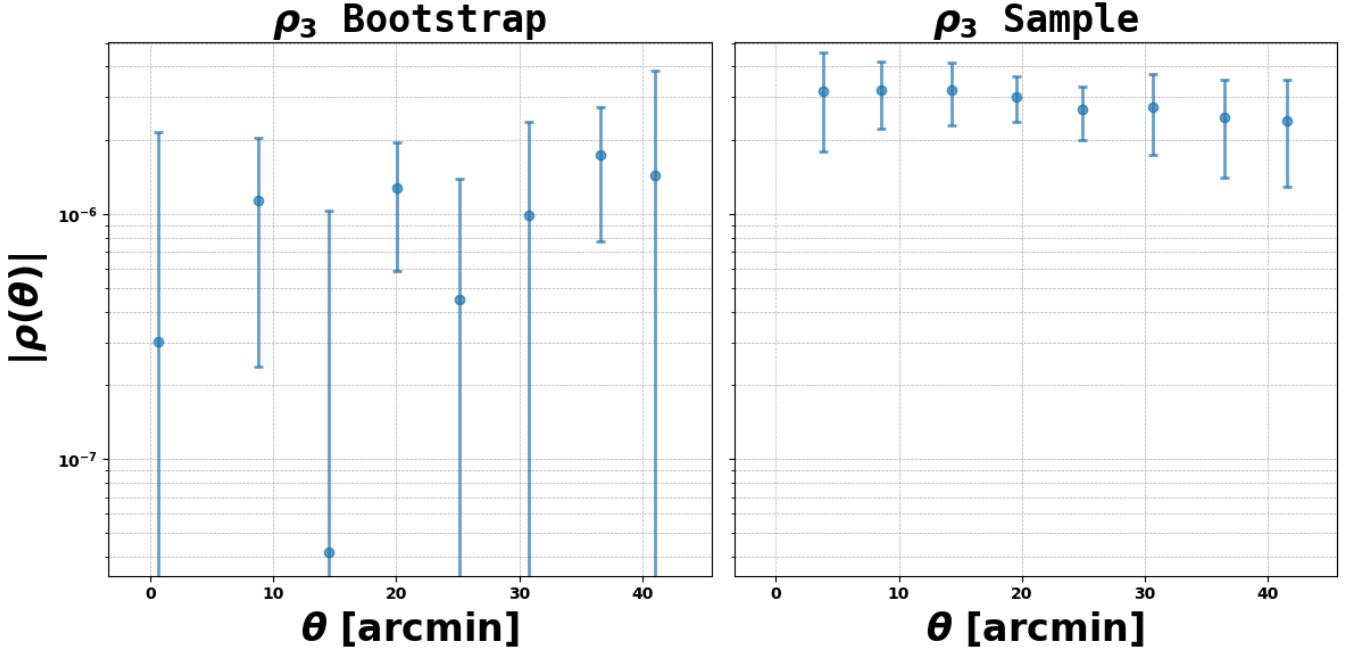


FIG. 15.— ρ_3 correlation function with data taken from the COSMOS-Web point source catalogs. Error bars drawn from bootstrap (left) and sampling (right) approaches. Negative correlations are shown in absolute value, and correlations are plotted in logarithmic scale.

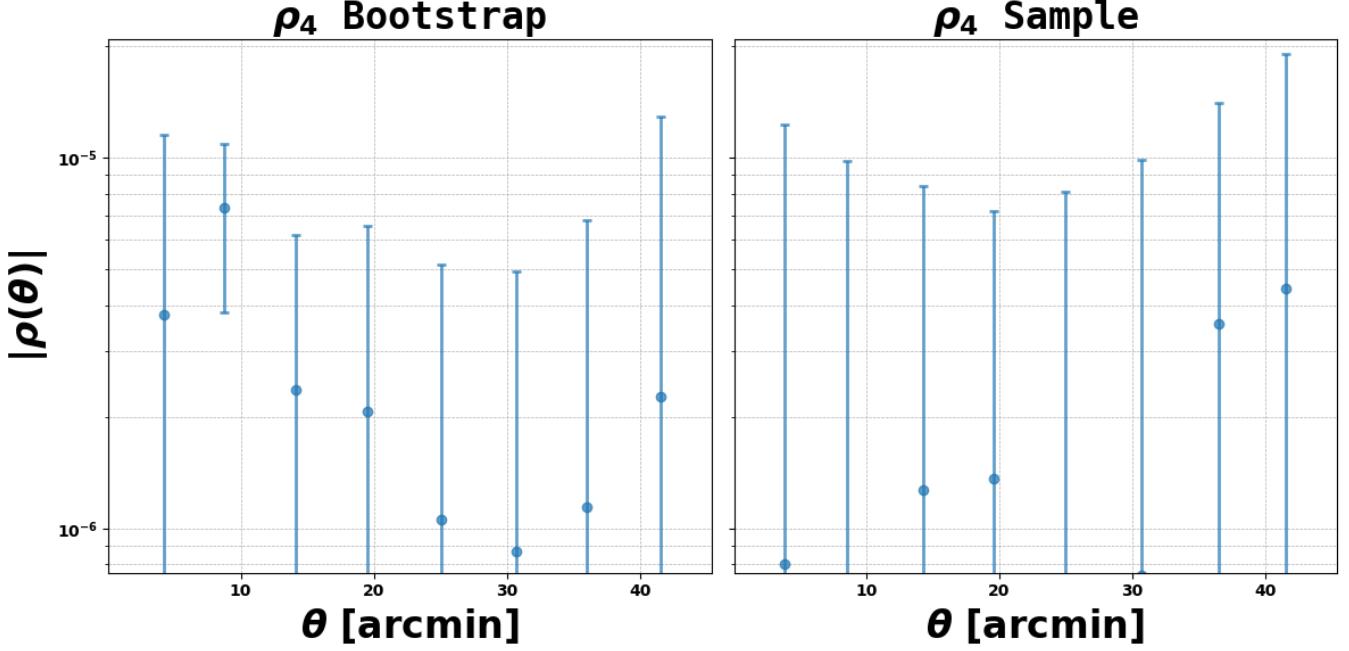


FIG. 16.— ρ_4 correlation function with data taken from the COSMOS-Web point source catalogs. Error bars drawn from bootstrap (left) and sampling (right) approaches. Negative correlations are shown in absolute value, and correlations are plotted in logarithmic scale.

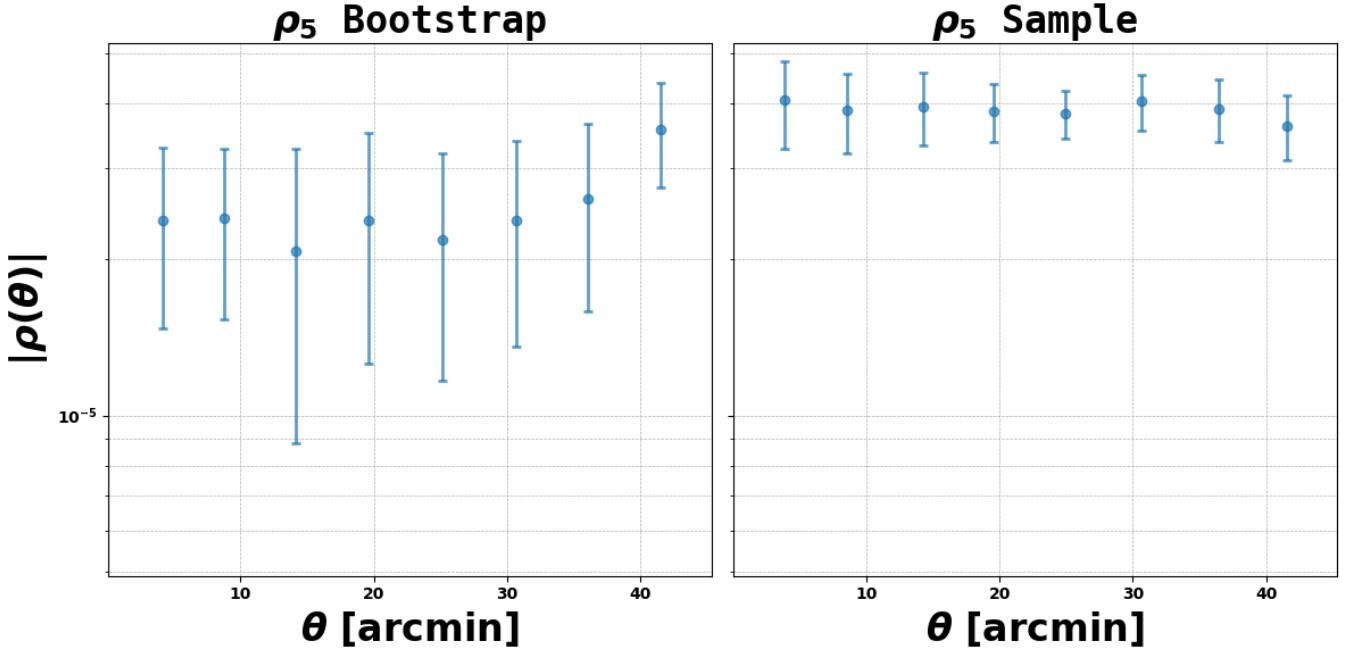


FIG. 17.— ρ_5 correlation function with data taken from the COSMOS-Web point source catalogs. Error bars drawn from bootstrap (left) and sampling (right) approaches. Negative correlations are shown in absolute value, and correlations are plotted in logarithmic scale.

JACOBIANS

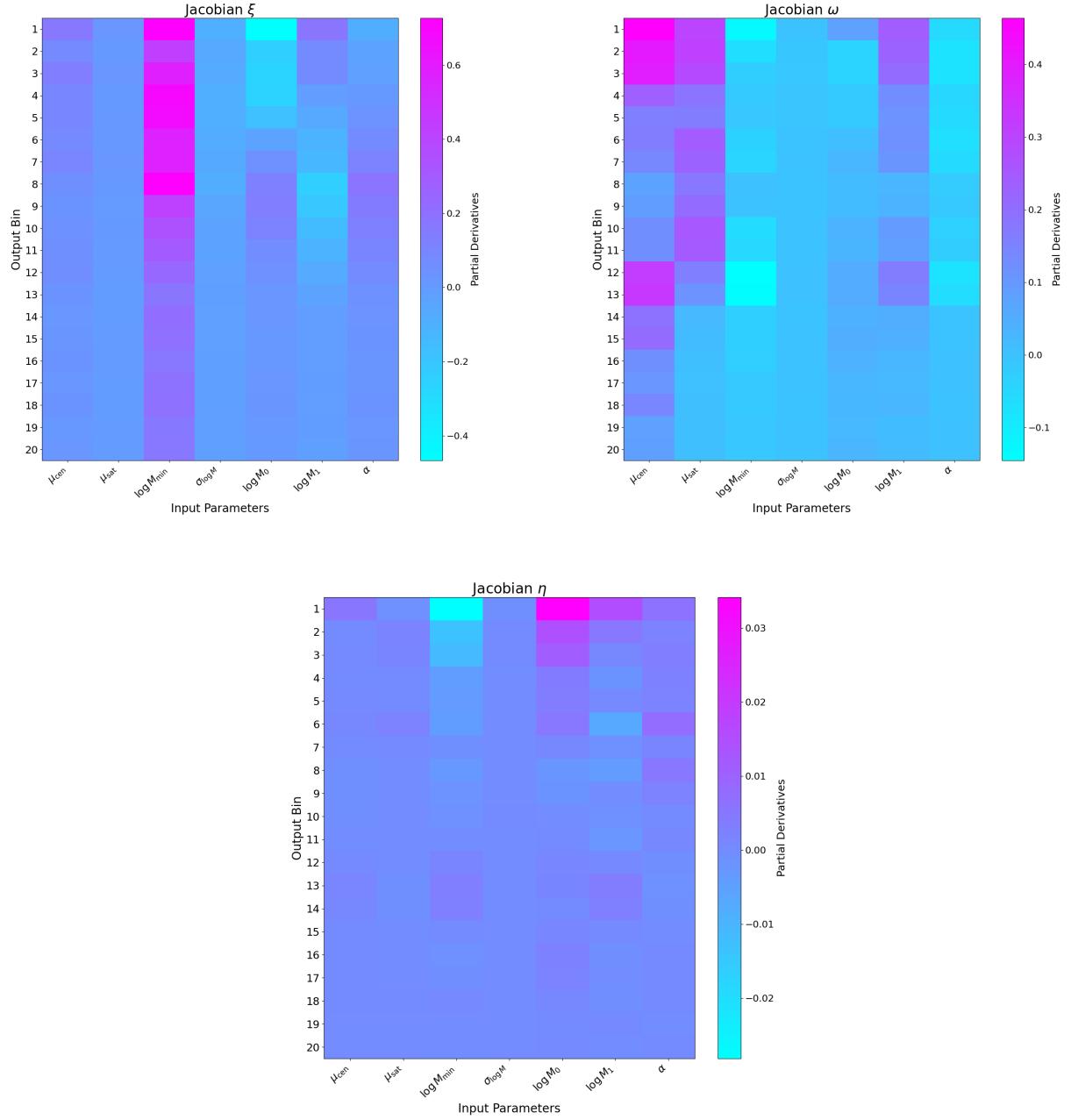


FIG. 18.— Jacobians associated with each of the 7 IA parameters and 20 output bins from IA-Emu.