A&A, 701, A44 (2025)
https://doi.org/10.1051/0004-6361/202453399

Astronomy & Astrophysics

Bridging simulations and observations: New insights into galaxy formation simulations via out-of-distribution detection and Bayesian model comparison

Evaluating galaxy formation simulations under limited computing budgets and sparse dataset sizes

Lingyi Zhou[1,⋆], Stefan T. Radev[3], William H. Oliver[1,2], Aura Obreja[1,2], Zehao Jin[4], and Tobias Buck[1,2,⋆]

[1] Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg, Im Neuenheimer Feld 205, 69120 Heidelberg, Germany
[2] Universität Heidelberg, Zentrum für Astronomie, Institut für Theoretische Astrophysik, Albert-Ueberle-Straße 2, 69120 Heidelberg, Germany
[3] Center for Modeling, Simulation, & Imaging in Medicine, Rensselaer Polytechnic Institute, NY, USA
[4] Center for Astrophysics and Space Science (CASS), New York University Abu Dhabi, UAE

Received 11 December 2024 / Accepted 7 July 2025

ABSTRACT

Context. Cosmological simulations are a powerful tool for advancing our understanding of galaxy formation. A question that naturally arises in light of high-quality observational data is the closeness of the models to reality. Because of the high-dimensionality of the problem, many previous studies evaluated galaxy simulations using simplified summary statistics.
Aims. We combine a simulation-based Bayesian model comparison with a novel mis-specification detection technique to compare galaxy images of six hydrodynamical models from the NIHAO and IllustrisTNG simulations against observations from SDSS.
Methods. Since cosmological simulations are computationally costly, we first trained a k-sparse variational autoencoder on the abundant dataset of SDSS images. The variational autoencoder learned to extract informative latent embeddings and delineated the typical set of real images. To reveal simulation gaps, we performed out-of-distribution detection based on the logit functions of classifiers trained on the embeddings of simulated images. Finally, we performed an amortized Bayesian model comparison using a probabilistic classification to identify the relatively best-performing model along with partial explanations through SHapley Additive exPlanations values (SHAP).
Results. We find that all six models are mis-specified compared to SDSS observations and can only explain part of reality. The relatively best-performing model comes from the standard NIHAO simulations without active galactic nucleus physics. Based on our inspection of the SHAP-values, we find that the main difference between NIHAO and IllustrisTNG is given by color and morphology. NIHAO is redder and clumpier than IllustrisTNG.
Conclusions. By using explainable AI methods such as SHAP values in combination with innovative methods from a simulation-based Bayesian model comparison and new mis-specification detection techniques, we were able to quantitatively compare costly hydrodynamical simulations with real observations and gain physical intuition about the quality of the simulation models. Hence, our new methods help to explain which physical aspects of a particular simulation cause the simulation to match real observations better or worse. This unique feature helps us to inform simulators to improve their simulation model.

Key words. methods: data analysis – methods: statistical – techniques: image processing – galaxies: formation – galaxies: photometry – galaxies: structure

1. Introduction

It is hard to investigate the physical processes that govern the formation and evolution of galaxies. Many of these processes span a very wide dynamical range and are coupled. To understand their importance for galaxy formation, we therefore need to run cosmological hydrodynamical simulations (Vogelsberger et al. 2020). It is a notoriously difficult task to assess the quality and realism of these simulations, however. A common approach is to compare the distribution of galaxy properties retrieved from simulations and observations as a diagnostic tool. Galaxy observations span a multidimensional complex parameter space (image-like or time series-like data), however, and it is unclear how a model comparison is performed best in this setup. Many previous works have measured the gap between simulation models and observations using traditional methods employing simple 2D or 3D summary statistics. For example, the Tully-Fisher relation of the luminosity of a spiral galaxy and its rotation velocity (Tully & Fisher 1977), the joint distribution of luminosity, optical rotation velocity, and disk size of spiral galaxies (Courteau et al. 2007), and the stellar mass and halo-mass relation (Moster et al. 2013, 2018). Many modern studies tried to match multiple observed properties of galaxies. For instance, Universe Machine (Behroozi et al. 2019), a recent algorithm for predicting observable galaxy properties based on simulations, is

⋆ Corresponding author: lingyi.zhou98@outlook.com; tobias.buck@iwr.uni-heidelberg.de

optimized to simultaneously match a wide range of these properties. This is a very limited criterion, however, because a model may closely match real observations under one such relation, but might deviate significantly from reality under another.

Even worse, a significant challenge arises when attempting to quantitatively evaluate the importance of different sets of summary statistics in the face of contradictions. In these cases, it becomes essential to explore alternative methods for the comparison that maximize the potential of the high resolution that is achieved by simulations and observations.

A natural choice here are galaxy images. Compared to simple summary statistics with high information loss, galaxy images contain a wealth of detailed information. Many previous works compared the distribution of image-based parameters in mock images with the distribution in real observations using a variety of metrics and statistical tools that have become a key tool in calibrating modern simulations (Snyder et al. 2015; Bottrell et al. 2017a,b; Rodriguez-Gomez et al. 2019; Bignone et al. 2020; De Graaff et al. 2022). These include parametric methods such as the Sérsic parameters (Sérsic 1963) and nonparametric methods such as concentration, asymmetry, clumpiness statistics (CAS Conselice 2003), the Gini-M_{20} method (Lotz et al. 2004) for identifying whether a galaxy has experienced a recent merger event, and the multimode, intensity, and deviation statistics (MID Freeman et al. 2013).

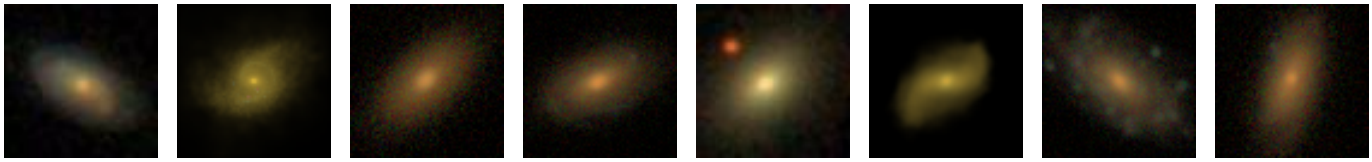
The rapid development of artificial intelligence (AI) has had a profound impact in many fields, including astrophysics. In particular, it improves our understanding of images and the comparison of simulations and real observations. Numerous studies have applied machine-learning methods to analyze galaxy images, thereby addressing challenges in astrophysics (Dieleman et al. 2015; Obreja et al. 2018; Buck & Wolf 2021; Buder et al. 2021; Cheng et al. 2021; Storey-Fisher et al. 2021; Smith et al. 2022; Tohill et al. 2024).

Recently, several works have explored machine-learning (ML) approaches to compare simulations and observations. Karchev et al. (2023) used the deep-learning method for hierarchical models proposed by Elsemüller et al. (2023b) to compare simulation-based supernova Ia light-curve models. Schosser et al. (2024) used a 3D-CNN as the summary network that compressed 3D light-cone data into six-dimensional latent vectors and an invertible neural network conditioned on an observation (cINN; Ardizzone et al. 2018) as the inference network for extracting the model parameter posterior. Zanisi et al. (2021) compared Illustris (Vogelsberger et al. 2014) and IllustrisTNG (Pillepich et al. 2018b) with r-band Sloan Digital Sky Survey (SDSS; Kollmeier et al. 2019) images by combining the output of two PixelCNN networks (Van Den Oord et al. 2016) to produce pixel-wise anomaly scores assigned to simulation images. Jin et al. (2024) proposed to use GANomaly (Akcay et al. 2019), an anomaly-detection network based on generative adversarial networks (GAN; Goodfellow et al. 2020) to rate NIHAO simulations (Numerical Investigation of Hundred Astrophysical Objects; Wang et al. 2015; Buck et al. 2019b, 2020) against SDSS images by assigning anomaly scores to galaxy images. In addition, Margalef-Bentabol et al. (2020) employed Wasserstein generative adversarial networks (WGANs) (Arjovsky et al. 2017) to find outliers in Horizon-AGN simulation (Dubois et al. 2014) using H-band CANDELS (Grogin et al. 2011; Koekemoer et al. 2011) images and the WGAN loss as the anomaly score. In another approach, Eisert et al. (2024) used representation-learning techniques, specifically, contrastive learning, to encode mock IlustrisTNG images and real HSC images into a joint embedding space. Among others, they used this embedding space in combination with a nearest-neighbor search to identify outlier galaxies.

A fundamental problem of all these approaches is that they rely on large training sets. Galaxy formation simulations are computationally expensive (∼10−100 k CPUh per instance), however, and we therefore propose a novel approach here. We leveraged a large set of real images (643 553) to pre-train a sparse embedding network that compresses simulated and real galaxy images into a structured latent space. This allowed us to highlight notable simulation gaps (Schmitt et al. 2023). Then, we used the amortized Bayesian model comparison (BMC) (Radev et al. 2021, 2023), which is a novel simulation-based inference method (SBI; Cranmer et al. 2020) for comparing analytically intractable high-dimensional models, to determine the relative fit of each model. This allowed us to efficiently handle a large number of images, which would not be computationally feasible with standard Bayesian methods. Because the size of our simulation dataset is limited (an inherent challenge for SBI applications, which typically require large amounts of data) we used ensemble methods to enhance the classifier performance and robustness despite the data scarcity.

The aim of this work is to provide a unified framework for a robust comparison of different numerical models of galaxy formation against observations. In particular, we compare models with different parameter choices, different numerical recipes, and sample sizes equally. In contrast to previous methods, our approach begins with a k-sparse variational autoencoder that is pre-trained on a large set of SDSS images. By learning a latent embedding that is rooted in abundant observational data, our method produces a representation space that better reflects the true distribution of galaxy properties. The use of sparsity in the autoencoder encourages the model to capture the most salient and physically interpretable features, thereby disentangling different aspects of the galaxies in a way that facilitates later analysis. Another approach to find useful representations was recently presented by Eisert et al. (2024), who presented a framework that uses a self-supervised contrastive-learning method to map both simulated and observed galaxy images into a 256-dimensional representation space. Their approach makes use of an E(2)-equivariant ResNet with steerable convolutional layers so that the resulting features are invariant to rotations and reflections, ensuring robustness against common observational variations. Their work focused on aligning simulated and observed distributions and on evaluating the overlap in the feature space, mostly in qualitative terms. Our approach, however, takes a step further by casting the model comparison as an amortized Bayesian inference problem. We interpret the classification task as deriving probabilistic model posteriors and use calibrated ensemble classifiers to produce these relative probabilities. This framework not only quantitatively ranks the performance of different simulation models, but also allows us to overcome the challenge of scarce simulation budgets by leveraging the large readily available observational dataset.

In terms of outlier detection, Eisert et al. (2024) focused on comparing the distribution of images through visualizations such as UMAP projections and nearest-neighbor distances, while we have developed a more robust, quantitative approach for out-of-distribution (OOD) detection. Eisert et al. (2024) primarily relied on geometric measures in the learned representation space to detect out-of-domain images. In contrast, we trained an ensemble of classifiers on the latent embeddings generated from simulation images to calculate a Generalized ENtropy (GEN) score (Liu et al. 2023), which provides a clear statistical criterion for identifying images that do not belong to the simulation domain.



(a) SDSS 100128 (b) TNG100 (c) AGN (d) NOAGN (e) SDSS 557508 (f) TNG50 (g) UHD (h) n80

Fig. 1. Example rgb images of SDSS (left), TNG100/TNG50 (second from left), and the various NIHAO models of the galaxy g8.26e11 with varying viewing angles (right two columns).

By comparing the GEN score distributions of the simulated data with those of the SDSS test set, we can discard OOD observations before we perform the model comparison, which ensures that only data compatible with at least one model are used.

A major innovation in our method is the integration of explainable AI tools, specifically, SHAP values, which we use to interpret the latent features that drive the model comparison. This analysis links abstract latent dimensions back to concrete physical properties such as color and light concentration. In our results, we observe that certain latent features strongly correlate with these physical characteristics, enabling us to explain why some simulation model variants appear relatively better than other models. This insight offers actionable guidance for simulation improvement by determining the areas in which subgrid physics may need refinement.

Compared to previous model comparison works in galaxy formation, our approach is distinguished by (1) its grounding in a richly detailed observational dataset; (2) its rigorous and quantitative OOD detection mechanism; (3) the application of a Bayesian model comparison framework that produces calibrated and interpretable posterior probabilities; and (4) the use of explainable AI to connect latent features with physical galaxy properties. These advancements not only provide a more robust comparison between simulations and observations, but also offer a clear pathway for improving the physical realism of cosmological simulations.

This paper is structured as follows. In Section 2 we start by describing the datasets of SDSS galaxy images together with the simulated galaxy images, followed by Section 3, where we discuss our methods, including our k-sparse VAE architecture, out-of-distribution detection for model mis-specification, and our amortized BMC procedure. In Section 4 we present our results of out-of-distribution detection, model comparison results, and our physical interpretation. Finally, Section 6 presents our conclusions and outlook. In Section 7 we provide the link to our Github repository for the reproducibility of our results, and in Appendix C, we present additional figures of various simulation models.

2. Image datasets and simulation details

2.1. Observed galaxy images

The Sloan Digital Sky Survey (SDSS; Kollmeier et al. 2019; Castander 1998) is one of the most influential astronomical surveys ever conducted. Its main goal is to create detailed multi-dimensional maps of the Universe, capturing images and spectra of millions of celestial objects. SDSS operates by capturing high-resolution images and acquiring spectral data that provide important information about the distances, composition, and motion of galaxies, quasars, and stars.

The SDSS images are produced in a set of broadband filters, of which we used three: the near-infrared (i), red (r), and green (g), which can then be combined into multicolor images by mapping the i, r, g-bands to the red, green. and blue color channels. Following the work of Jin et al. (2024), we used the galaxy catalog by Meert et al. (2015) with redshift from 0.005 to 0.395 (with a mean of 0.109), which provides coordinates and stellar masses for 670 722 observed galaxies. To avoid star contamination, we only use galaxies with a stellar mass greater than $10^9 M_\odot$ which leaves us with 643 553 galaxy images for the SDSS dataset in our work. SDSS images are cropped to a resolution of 64×64 pixels around each galaxy's coordinates, with the size of the cropped region determined by the pixel scale of the SDSS camera (0.396 arcseconds per pixel). For training our embedding network, we split the SDSS dataset into a training set (70%), a test set (10%) and two validation sets: one for early stopping (10%) and one for hyperparameter tuning (10%). Two examples of the SDSS images are provided in the left most column of Figure 1.

2.2. Simulated galaxy images

We compare simulated galaxy images from six different candidate models taken from two different simulation projects: TNG50 and TNG100 from the IllustrisTNG simulations (Nelson et al. 2019a,b; Pillepich et al. 2018a, 2019; Springel et al. 2018) which only differ in terms of resolution, and four models from the NIHAO simulation suite that vary the physics implementation and resolution: NIHAO-AGN (Blank et al. 2019; Waterval et al. 2022), NIHAO-NOAGN (Wang et al. 2015), NIHAO-UHD (Ultra High Definition; Buck et al. 2020) and NIHAO-n80 (Buck et al. 2019a; Macciò et al. 2022). Below we describe in detail the different aspects of the simulation models.

2.2.1. IllustrisTNG models

IllustrisTNG (Pillepich et al. 2018a; Nelson et al. 2018) is a suite of magneto-hydrodynamical cosmological simulations that model the formation and evolution of galaxies within the ΛCDM cosmology simulated with the moving mesh code Arepo (Springel 2010). The IllustrisTNG suite models galaxy formation in three uniform mass resolution cosmological volume simulations with side lengths $35h^{-1} \approx 50$ Mpc, $75h^{-1} \approx 100$ Mpc and $205h^{-1} \approx 300$ Mpc, referred to as TNG50, TNG100 and TNG300. In this work we only use data from the former two runs, TNG50 and TNG100. The TNG models are in particular well suited for studying large-scale structures and statistical properties of galaxies. It offers valuable data on galaxy formation and evolution, the distribution of dark matter, and how explosions from stars and black holes affect galaxies. The only difference between TNG100 and TNG50 is the physical resolution of the underlying simulation. Sample images from the TNG100 and TNG50 simulations are shown in the second column of Figure 1.

2.2.2. NIHAO models

The NIHAO (Numerical Investigation of Hundred Astrophysical Objects) simulation (Wang et al. 2015) is a suite of hydrodynamical cosmological zoom-in simulations computed with the GASOLINE2 code (Wadsley et al. 2017). NIHAO adopts a flat ΛCDM cosmology and parameters from the Planck satellite results (Planck Collaboration XVI 2014). It includes Compton cooling, photoionization from the ultraviolet background following Haardt & Madau (2012), star formation and feedback from supernovae (Stinson et al. 2006) and massive stars (Stinson et al. 2013), metal cooling, and chemical enrichment. A series of prior work has proven that NIHAO simulated galaxies reproduce galaxy scaling relations very well, including the Stellar Halo-Mass relation (Wang et al. 2015), the gas mass and size relation of the disk (Macciò et al. 2016), the Tully-Fisher relation (Dutton et al. 2017), the diversity of galaxy rotation curves (Santos-Santos et al. 2018), and the mass-metallicity relation (Buck et al. 2021).

In what follows, we refer to this basic version of NIHAO without AGN feedback as 'NIHAO NoAGN'. NIHAO NoAGN is the basis for other variations of NIHAO that explore different physical models for star formation and feedback in addition to increased resolution. We describe those versions below.

- NOAGN: NOAGN is the vanilla version of the NIHAO simulations as described above.
- AGN: The NIAHO model with AGN feedback (Blank et al. 2019; Waterval et al. 2022) has the same initial conditions, parameters, and physics as the NOAGN model. It adds Active Galactic Nuclei (AGN) physics in addition to the fiducial physics modeled in NOAGN.
- UHD: NIHAO-UHD (Ultra High Definition; Buck et al. 2020) includes a higher resolution version of several Milky Way-like galaxies from the NOAGN model. It has the same initial conditions, parameters, and physics as NOAGN. NIHAO-UHD has produced results that closely match the observed properties of both the Milky Way (MW) and Andromeda (M31) galaxies, such as satellite mass function and MW bulge properties. For more details, see Buck et al. (2019b, 2018), and Buck et al. (2019c).
- n80: star formation is typically simulated using a density threshold n, measured in particles per cm^3. The transformation of gas particles into star particles begins only when this threshold is attained. All other NIHAO models have a threshold of $n = 10\,\mathrm{cm}^{-3}$ while the n80 model uses a higher value of $n = 80\,\mathrm{cm}^{-3}$ to re-simulate a few galaxies from NIHAO NOAGN (Macciò et al. 2022). The detailed impact of the star formation threshold for the NIHAO simulation is discussed in Dutton et al. (2019); Buck et al. (2019a) and Dutton et al. (2020).

Example images generated from the NIHAO models are provided in the two right most columns of Figure 1.

2.2.3. Mock observation image pipeline

All simulated galaxy images are created with the same image pipeline based on radiative transfer (RT) post-processing of the simulated galaxies using the SKIRT code (Camps & Baes 2015). For the IllustrisTNG models we use the synthetic image data from Rodriguez-Gomez et al. (2019), available on the open data access website[1]. We use the redshift zero snapshot (snap number 99) for both TNG100 and TNG50 data. IllustrisTNG galaxies are restricted to subhalos with stellar mass greater than $10^{9.5} M_\odot$ for which morphological measurements are reliable with parameter $flag = 0$ and $sn\_per\_pixel > 2.5$. For the NIHAO models we use the same synthetic data as Jin et al. (2024) based on RT postprocessing done by Faucher et al. (2023).

For both simulation projects, we create RGB images from the raw RT output following the image pipeline of Jin et al. (2024) which combines the i, r, g images using an `arcsinh` stretch as proposed by Lupton et al. (2004), applies a point spread function (PSF) and adds shot noise and Gaussian sky noise to model observational uncertainties following procedures taken from the REALSIM code by Bottrell et al. (2019). More specifically, to account for the SDSS PSF, we use a Gaussian PSF with a full width at half maximum corresponding to the average seeing of all SDSS Legacy galaxies: 1.286, 1.356, and 1.496 arcseconds for the i, r, and g bands, respectively. The physical sizes of the simulated galaxies are converted to angular scales by placing them at a hypothetical redshift of 0.109, matching the mean redshift of our SDSS training sample. Shot noise is modeled as Poisson noise, determined by the survey field parameters, including zero-points, airmass, extinction, and CCD gain. Gaussian sky noise is derived from the average sky noise across all Legacy galaxies. Finally, an `arcsinh` stretch, as proposed by Lupton et al. (2004), is applied to align with the standard SDSS imaging scheme.

In the following figures, NIHAO models are named with an additional suffix "rt" to distinguish them from a previous version of our image data. Since simulated images have different resolutions other than SDSS, we upsample or downsample them to a resolution of 64×64 pixels. Our final simulated image dataset includes 11 334, 1523, 1521, 1540, 120, and 240 images for TNG100, TNG50, AGN, NOAGN, UHD, and n80, respectively. Since this leads to an imbalanced final dataset (TNG100 is the majority class), we oversample the images in the minority classes, see Section 3.4.

For the subsequent steps of model comparison, we stratify the dataset of simulation models into a training set (85%) and a test set (15%), ensuring that the class proportions remain unchanged after the split. Importantly, we performed the train-test split before oversampling so that copies do not get split across training and test sets.

3. Methods

3.1. Learning summary statistics with limited simulation budgets

In many fields of natural science, models are developed to explain natural phenomena based on some theories. Due to randomness in physical processes, measurement processes and other influence factors, predictions from these models are not deterministic, and we instead obtained statistical distributions of parameters. In most cases, we are interested in performing inference on model parameters θ given observations $\mathcal{D}$, that is, we seek to recover the parameter posterior $p(\theta \mid \mathcal{D})$. Bayes' proportionality $p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta)p(\theta)$ connects the posterior $p(\theta \mid \mathcal{D})$ with the likelihood $p(\mathcal{D} \mid \theta)$ and the prior $p(\theta)$.

The data generation process can be described by sampling some unobserved internal states y according to $y \sim p(y \mid \theta)$. The likelihood function is then calculated by marginalizing the joint distribution $p(\mathcal{D}, y \mid \theta)$ over all possible internal states y,

$$p(\mathcal{D} \mid \theta) = \int p(\mathcal{D}, y \mid \theta)\mathrm{d}y. \quad (1)$$

[1] https://www.tng-project.org/data/docs/specifications/#sec5l

Even in the case of relatively simple models, however, this integral is usually intractable. Through implicit likelihood models (Cranmer et al. 2020) that are also known as simulation-based inference (SBI), however, we can still approach these problems. Given a model, a simulator is created to produce simulated data from model parameters θ for the phenomena of interest: samples $\sim$ simulator(θ). Given a sufficiently fast simulator, we can then make use of a lot of data parameter pairs generated by the simulator to learn the connection between the two and hence enable inferring model parameters from data.

Our model comparison problem is a variant of SBI. The difference is that we are interested in the model posterior $p(M|\mathcal{D})$ conditioned on data $\mathcal{D}$ rather than the posterior of model parameters.

A typical approach in SBI methods for model comparison is reducing the original data into fixed summary statistics (also called embeddings in our context) to avoid working with high-dimensional observables, such as galaxy images. Additionally, Radev et al. (2021) proposed to train embedding networks that capture the structure of the original data, avoiding catastrophic information loss and biased results caused by hand-crafted summary statistics (Robert et al. 2011; Marin et al. 2018). End-to-end learning of summary statistics requires large simulation budgets, however, which are not feasible in our setting. For instance, running the TNG100 simulations alone on the Cray XC40 Hazel Hen supercomputer[2] demands 1.5 years of runtime on several ten thousand cores, equivalent to millions of CPU hours, making such simulation efforts prohibitively expensive for different model variants. To overcome this problem, we take another approach and leverage the large body of real observational data from SDSS to train an embedding network (i.e., an encoder) in a fully unsupervised manner as part of an information maximizing variational autoencoder architecture.

After training on observational images, we "freeze" the encoder and embed the simulated images into the lower dimensional latent space. This small dataset of "labeled" embeddings then serves as the training data for an ensemble of classifiers. Once training has converged, we apply the trained ensemble classifiers to the embeddings of SDSS test set galaxies and perform out-of-distribution detection to find those SDSS images that cannot be accounted for by any of the simulators (i.e., due to model mis-specification). Finally, we discard this part of SDSS images in our final model selection task and perform amortized Bayesian model comparison only on the subset of "in-simulation" embeddings. Only in this way we obtain trustworthy model posterior probabilities.

In the next subsections we explain in detail each step of our methods. To better illustrate our workflow, we show a flow chart of our pipeline in Figure 2.

3.2. Auto-encoding galaxy images

We use a k-sparse variational autoencoder (VAE; Kingma & Welling 2013) based on the k-sparse autoencoder (Makhzani & Frey 2013) to encode galaxy images to embeddings. Compared to plain autoencoders, VAEs provide a probabilistic framework by encoding original data to a distribution instead of single points and help prevent overfitting. Additionally our approach of employing sparsity in the VAE helps us achieve better disentanglement in the latent space, thus facilitating physical interpretability in a later step.

[2] https://www.tng-project.org/people/

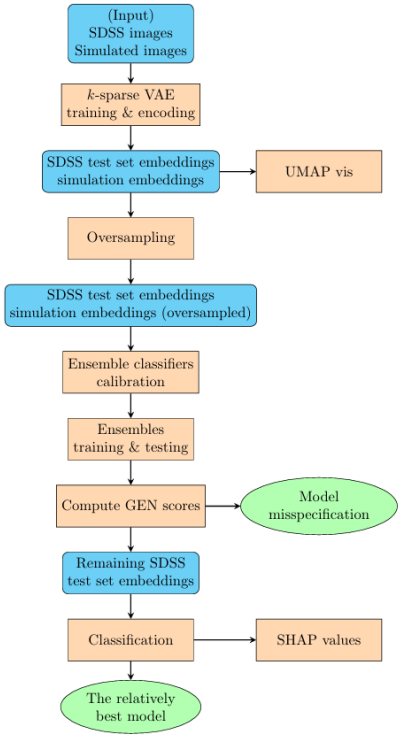


Fig. 2. Flow chart of our workflow pipeline. The blue boxes with rounded corners represent the data products of the previous step. The grange boxes show the main algorithmic steps in our method. The green ellipses represent output results.

In practice, we sampled an embedding z from the distribution of embeddings by using the reparameterization trick. This technique allows gradients to flow through random variables by expressing them as a deterministic function of the distribution parameters and a noise term. We also incorporated the MMD-VAE loss from the InfoVAE family (Zhao et al. 2017), where MMD stands for maximum mean discrepancy, to avoid common problems with the standard VAE KL divergence loss (Kullback-Leibler divergence) and encourage maximally informative compression. Our final loss function is

$$\mathcal{L} = \mathbb{MMD}^2(q_\phi(z) \,\|\, p(z)) + \mathrm{MSE}(x_{\mathrm{recon}}, x) \tag{2}$$

where $q_\phi(z)$ represents the approximate distribution of the embedding z, $p(z) \sim \mathcal{N}(0, \mathbb{I})$, MSE stands for the mean squared error, x is the input image and x_{recon} is the image reconstructed by the decoder. Here we set a 1:1 ratio of the MMD and MSE terms, following the official implementation of InfoVAE. Additionally, we explored the impact of varying this ratio by testing two alternative configurations: $\mathcal{L} = 0.1 \times \mathbb{MMD}^2(q_\phi(z)\|p(z)) + \mathrm{MSE}(x_{\mathrm{recon}}, x)$ and $\mathcal{L} = \mathbb{MMD}^2(q_\phi(z)\|p(z)) + 0.5 \times \mathrm{MSE}(x_{\mathrm{recon}}, x)$, to assess the robustness of our results. Under both conditions, our findings and conclusions remained consistent, however, and we therefore do not present these additional results in Section 4.

During training, we compute the embedding z in the feed forward phase, then sparsify it by keeping only the k largest activations (absolute values) and setting the rest to zero. The computation of the loss function and the input of the VAE decoder both use the sparsified z. We train the k-sparse VAE on the SDSS training set with dimension of z equal to 512 and $k = 32$. The embedding dimension and sparsity level k are hyperparameters that can be tuned. We choose the sparsity ratio to

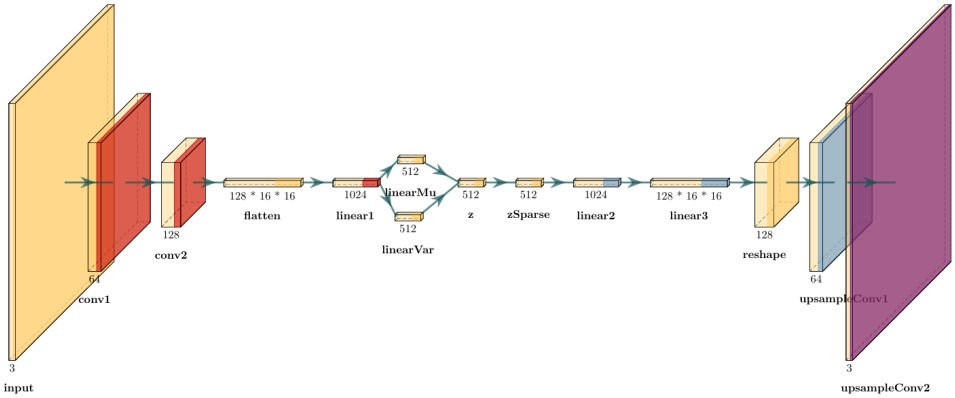


Fig. 3. Visualization of the k-sparse VAE structure. Deep orange stands for LeakyReLU activation function, blue stands for ReLU and purple stands for sigmoid activation function. The numbers near illustration mean filter sizes.

be $32/512 = 0.0625$ for our k-sparse VAE achieving a balance between capturing local features and global features (Makhzani & Frey 2013). Then we encode the SDSS test set and the simulated images to 512 dimensional embeddings with $k = 64$. The sparsity level $k = 64$ is determined by following the result of the original paper, as a larger k during the encoding phase reduces the error rate of downstream classification task (Makhzani & Frey 2013). This suggests that different k values for training and testing do not introduce any systematic effects that could resemble OOD behavior; otherwise, the classifier's performance would worsen as it wouldn't handle OOD data well. In our setup, the choice of larger k during testing yields more robust embeddings without compromising the disentanglement or interpretability of the latent space. We also verified that our main results are qualitatively stable across a reasonable range of k values (from 16 to 64), indicating that our framework is not overly sensitive to this hyperparameter.

The network structure of our k-sparse VAE is illustrated in Figure 3. The encoder consists of convolutional layers, flatten layers and linear layers. In the decoder, we employ an upsampling layer followed by a convolutional layer, rather than using a deconvolution layer, in order to mitigate the checkerboard effect. The checkerboard effect refers to a visual artifact in reconstructed images, characterized by abrupt, unnatural transitions in pixel colors or luminance values, which result in a grid-like pattern resembling a checkerboard.

3.3. Training and encoding of k-sparse VAE

As suggested by Makhzani & Frey (2013), we deploy a scheduling of the sparsity level during training by starting with a larger sparsity level $k = 81$ and then linearly decrease it to the target sparsity level $k = 32$ in the first 7 epochs. By doing so, we avoid the problem of dead hidden units that can appear in the k-sparse autoencoder, where some hidden units are selected during the initial epochs and reinforced in later ones, while others remain unadjusted. For training the k-sparse VAE on the SDSS training set, we use the Adam optimizer (Kingma & Ba 2014) with an initial learning rate of 10^{-3} and a reduce-on-plateau schedule for dynamically reducing the learning rate by a factor of 0.1 if the average validation loss per epoch has no improvement after 5 epochs. We have an early stopping mechanism in place, where the training halts if the average validation loss does not decrease for 10 consecutive epochs in which case we choose the model checkpoint at those previous 10 epochs. Using a batch size of 400, the training of the k-sparse VAE for 38 epochs on a single A100 GPU takes 6802 seconds (~1 hour 53 minutes). The final model checkpoint which we evaluate is hence at the 28th epoch. The average training and validation loss of each epoch during training is shown in Figure 4.

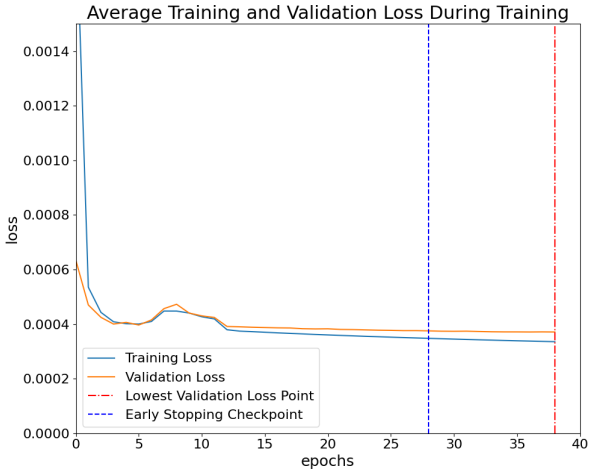


Fig. 4. Average training and validation loss of each epoch during k-sparse VAE training on the SDSS training set and validating on the SDSS early stopping validation set. The vertical red line shows the lowest validation loss checkpoint and the vertical blue line indicates the epoch where our VAE model is saved. This early stopping mechanism avoids overfitting.

3.4. Dealing with an imbalanced dataset

As stated in Section 2.2, our simulation data from 6 different simulation models form an imbalanced dataset where TNG100 is the majority class and all other models are minority classes. Imbalanced datasets are a common and challenging problem which often results in a biased machine learning model that performs well on the majority classes but poorly on the minority classes. To alleviate this issue, we oversample the images in the minority classes by copying them once. In our experiments, without oversampling, the classifiers in later steps are poorly calibrated with suboptimal confusion matrices and calibration curves.

We are aware that the usual procedure in oversampling is to create a balanced data set by replicating the minority instances

according to the class ratios. In our experiments, if images of minority classes are copied more than once, then the calibration curve becomes worse (see the classifier calibration section later). We have also considered modifying these copies (e.g. geometric transformations, color space transformations, etc). There are physical meanings behind the color and shape of galaxies, however, and we therefore cannot change these properties randomly.

While undersampling the TNG100 class could in principle balance the dataset, we chose not to do so in order to preserve the full diversity of available simulated galaxy images, which is especially important given the limited sample sizes across all models. Reducing the majority class would have discarded valuable information and likely weakened classifier performance and calibration. Additionally, although standard augmentation techniques such as rotations and flips are common in image classification, we deliberately avoided them in our setup. The simulated images include structured observational effects (see Section 2.2) – such as directional PSF convolution and realistic sky noise – which are sensitive to spatial orientation. Applying such augmentations would introduce unphysical artifacts that are not present in SDSS observations and could lead to spurious classification features or affect OOD detection. We therefore adopt a conservative oversampling strategy by duplicating each minority class once, which we found to improve classifier calibration while maintaining physical fidelity in the training set.

In addition, oversampling of embeddings does not work in our case. Each embedding corresponds to a simulated galaxy image. Using resampling techniques like SMOTE (Synthetic Minority Over-sampling Technique; Chawla et al. 2002) may create embeddings that match unrealistic images.

3.5. Amortized Bayesian model comparison and detecting a model mis-specification

BMC offers a probabilistic / relative comparison of multiple models rather than an absolute assessment of any individual model. BMC compares candidate models by evaluating the ratio of their posterior probabilities given known data and prior knowledge. It can be cast into a classification task by training a classifier with a proper loss (Gneiting & Raftery 2007) to induce a categorical distribution over the model indices $\mathcal{M}$ given the observed data D (Radev et al. 2021):

$$\mathcal{M} \sim p(\mathcal{M} \mid \mathcal{D}) \propto p(\mathcal{D} \mid \mathcal{M})\, p(\mathcal{M}). \tag{3}$$

Correspondingly, we train an ensemble of classifiers on the "labeled" embeddings of the simulated images and then use the trained ensemble to estimate posterior model probabilities from the SDSS test set. The model with the best relative fit from a Bayesian perspective (MacKay 2003; Radev et al. 2021) is then the one that is preferred by the classifier. The classifier can be efficiently reused for inference as new observations come in, hence the training cost *amortizes* over multiple observations. This is a notable advantage compared to posterior predictive methods like WAIC (widely applicable information criterion; Watanabe 2013) and cross-validation based methods (Vehtari et al. 2017), which need to calculate the likelihood of each observed data point given each model is required.

We were unable to simply apply the ensemble to all embeddings of our SDSS test set because some of them might be out of distribution (OOD; Yang et al. 2024) relative to the simulations, which can lead to incorrect or unstable predictions (Schmitt et al. 2023; Elsemüller et al. 2023a). Intuitively speaking, when classifying a data point that lies far from the distributions of all candidate classes, classifiers may assign a disproportionately high probability to the class that is only slightly closer to the data point.

In our context, OOD embeddings occur when the simulations differ significantly from the (actually observed) SDSS test data and thus indicates model mis-specification, which shows that the simulation differs significantly from the true data generating process (also called simulation gap). We can use any *post hoc* OOD score (Yang et al. 2024) to *detect* observations for which the models are mis-specified. A *post hoc* OOD score identifies OOD samples using a trained model's outputs without retraining or altering the model. In practice, we perform out-of-distribution detection using the Generalized ENtropy score (GEN score; Liu et al. 2023) which is defined as

$$G_\gamma(p) = \sum_j p_j^\gamma (1 - p_j)^\gamma \tag{4}$$

with $\gamma \in (0, 1)$, where p are the probabilities of all classes calculated by applying the softmax function to the logits produced by the classifiers. In practice, we follow the recommendation from (Liu et al. 2023) in setting $\gamma = 0.1$, which was shown to perform well across diverse classification tasks. We additionally verified that our downstream results – specifically, the classifier calibration and relative model rankings – remain qualitatively stable for alternative values of γ. This indicates that our OOD detection method is not overly sensitive to the specific choice of γ.

We consider all simulation models in the computation of the GEN score since we only have 6 classes rather than hundreds of classes as in the paper of Liu et al. (2023). Finally, following the implementation of the original paper (Liu et al. 2023), we compute negative GEN scores. In order to perform OOD detection, we proceed as follows: We fit a classifier to the SDSS test set and compute the corresponding GEN score distribution. Similarly, we compute the reference GEN score distribution by classifying the simulation test set. If the GEN score distribution of the SDSS test set lies significantly outside of the reference GEN score distribution, then the SDSS test set is OOD, implying that the simulations deviate from reality. In this case, we perform model comparison only on the subset of SDSS data that "agree" with the simulation embeddings. For this, we take the percent point corresponding to 95% of the reference GEN score distribution as a threshold and ignore all SDSS latent embeddings with a GEN score beyond this threshold. To this "cleaned" SDSS test dataset we apply the classifiers once again to derive our final model posteriors. In this way, we can increase the robustness of model posterior estimates and the corresponding theoretical implications. Additionally, in Appendix B we conduct a sensitivity analysis on the threshold of the reference GEN score distribution to justify our choice of 95% as a reasonable value.

We like to note here that by using the GEN score to select SDSS galaxies that are in agreement with the simulation models circumvents the need for any simplified cut in SFR or total stellar mass. Using our embedding vectors combined with a GEN score actually allows for a more nuanced comparison between simulated and real galaxy images than any simple cut in stellar mass or SFR would have accounted for. Additionally, since we pre-train our embedding network on real images, any type of distribution shift not only a shift in stellar mass or SFR but also a shift in a mismatch in noise patterns or PSF of the simulated images will lead to the simulation data being out-of-distribution compared to real galaxy images. This is the key motivation for doing model comparison only on in-distribution data. Combined

with our explainable AI approach described in Section 3.7 our method is able to investigate the reasons for out-of-distribution data and should reveal mismatches in PSF or noise patterns.

3.6. Ensemble classifiers and calibration

As a baseline model we choose a random forest classifier (Breiman 2001), which has been used in previous model comparison papers (Pudlo et al. 2016; Marin et al. 2018). As stated before, we have a small set of simulation data and ensemble methods handle this situation better than a single neural network. We also train an XGBoost (eXtreme Gradient Boosting; Chen & Guestrin 2016) classifier and we additionally train a stacking ensemble classifier (referred to as stacking-MLP-RF-XGB in the following), which combines 3 base classifiers (multilayer perceptron, random forest and XGBoost) with a random forest serving as the final meta estimator.

In the random forest classifier, we set the number of trees to 100 and class weight to balanced subsample, which means the weight assigned to each class is determined by its inverse proportionality to the frequency of the class within the bootstrap sample used for growing each tree. We use the XGBoost classifier from the XGBoost library (Chen & Guestrin 2016), in which we use the multiclass softmax objective and hist tree method, which is the fastest approximated training algorithm. For stacking-MLP-RF-XGB classifier, the parameters of random forest and XGBoost are the same as described above. For the MLP, we set two hidden layers: the first one with 128 neurons and the second one with 64 neurons. And we set the activation function to ReLU, use the Adam optimizer (Kingma & Ba 2014) with an L2 regularization alpha set to 0.01, and maximum iteration of 300.

The training of our stacking classifier proceeds as follows: Each of the 3 base classifiers are trained to output a 6-dimensional class probability vector for each instance. Then we concatenate the class probability predictions from each of the 3 base estimators to obtain an 18-dimensional vector as the meta-feature vector (input feature for meta classifier) for each instance. Finally, the meta classifier (random forest) is trained on this new dataset with 18 features.

We perform the classifier calibration by doing twice repeated stratified 5-fold cross-validation on the simulation training set. In practice, we use the calibration curve function from the bayesflow library[3] (Radev et al. 2023). It integrates the computation of expected calibration error (ECE) of a model comparison network proposed by Pakdaman Naeini et al. (2015). We can better explain the calibration curve by describing its creation, which involves the following steps for a simulation model:

- The probability range [0,1] is divided into 10 bins of equal length. Then we assign predicted probabilities to bins.
- In each bin, we compute the mean of predicted probabilities and this is the value on the x-axis.
- In each bin, we compute the proportion of data that actually belongs to this model and this is the value on the y-axis.

From the calibration curves in Figure A.1 we can see that the stacking MLP-RF-XGB classifier is the best one with lowest ECE score and calibration curves close to the optimal diagonal line. Overall, the stacking-MLP-RF-XGB classifier achieves better recovery than random forest or XGBoost. The second best calibrated classifier is XGBoost which performs better than random forest. Note that the calibration curves of all classifiers for UHD and n80 are not well calibrated. Since these two classes are the smallest, we attribute this simply to the lack of data. This limitation highlights a general requirement of our approach: while the method is designed to handle relatively small simulation datasets, reliable calibration typically requires a minimum of several hundred images per model class to ensure meaningful posterior estimation. For extremely small datasets classifier confidence and interpretability can degrade.

The resulting figures for the confusion matrices are shown in Figure A.2 also in Appendix A. Each of the 3 classifiers produces a satisfying confusion matrix. All classifiers achieve a very high accuracy of classifying AGN, NOAGN and TNG100, while the accuracy of classifying TNG50, UHD and n80 is lower. Again, for UHD and n80, this may be due to the lack of data. Notice that the accuracy of classifying TNG50 is relatively poor in the sense that classifiers tend to falsely predict some data of TNG50 as TNG100. This is a reasonable result, however, because the only difference between TNG50 and TNG100 is the higher resolution of TNG50. Hence, we hypothesize that this makes the two less distinguishable.

Considering both confusion matrix and calibration curve, we conclude that we should trust the stacking-MLP-RF-XGB classifier most in the final classification results.

Finally, the random forest is trained on CPU (2 × 32-Core AMD Epyc 7452) while the XGBoost classifier is trained on a GPU (A100). Among the base estimators of stacking-MLP-RF-XGB, XGBoost is trained on GPU while others are trained on CPU. The training of random forest takes 12.6 seconds, XGBoost takes 14.6 seconds and stacking-MLP-RF-XGB takes 293.8 seconds.

3.7. Physical insights through explainable AI

SHAP (Lundberg & Lee 2017, SHapley Additive exPlanations) is an XAI (explainable machine learning) method used to explain the contribution of each feature to the final prediction of a machine learning model. It borrows the concept of Shapley values from game theory which are designed to distribute total gain among players in a group based on each one's contribution. For a game, the Shapley value for player i is defined as

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \left(v(S \cup \{i\}) - v(S)\right) \tag{5}$$

where N is the set of all players, S is a subset of players not including player i, $|S|$ is the number of players in subset S, and $v(S)$ represents the total payoff of coalition S resulting from the cooperation of players. Intuitively, the Shapley value computes the average marginal contribution of a player to all possible coalitions.

In XAI the SHAP method adapts Equation (5) by converting players to features and the value function $v(S)$ now becomes the model prediction function. In this sense, the SHAP value of feature i is the average marginal contribution of the feature to the prediction, accounting for all possible combinations of other features. Obviously, the exact computation of Equation (5) is not feasible and hence in practice SHAP values are estimated (Lundberg & Lee 2017). For an instance vector x, the sum of SHAP values of all features is equal to the difference between the actual prediction and the average prediction:

$$\sum_{j=1}^{n} \phi_j = f(x) - \mathbb{E}_x(f(x)). \tag{6}$$

[3] https://bayesflow.org

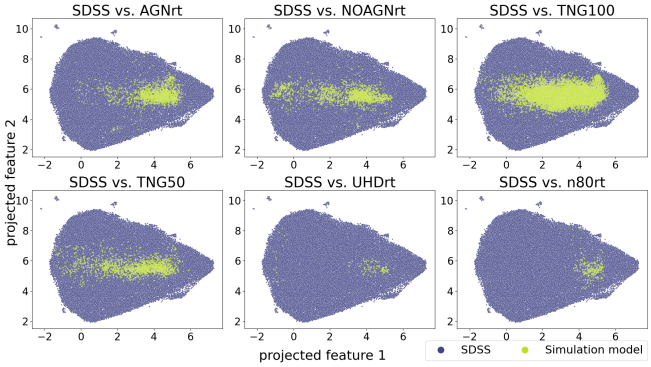


Fig. 5. UMAP projection of simulated latent embeddings (yellow) compared to the SDSS test set (purple).

In order to compute the expected value of the model prediction the SHAP software uses "background data" to calculate a reference point for the SHAP value interpretation. Thus, the background data should be chosen to represent the data distribution, which is typically selected from the training dataset. We therefore used stratified sampling to select 50% of the simulation training set as background data due to the imbalanced datasets. Then we apply the trained SHAP tree explainer to 60% randomly sampled "in-distribution" SDSS test data for balancing calculation time and result reliability.

In Section 4.4 we interpret our model comparison results qualitatively in light of physical difference between the simulation models via an analysis of SHAP values on the XGBoost classifier. In practice, we use a variant of the original SHAP method designed for tree-based machine learning models called TreeSHAP (Lundberg et al. 2020). We like to note that the SHAP method does not work for complex stacked classifiers such as our preferred stacking-MLP-RF-XGB. All classifier results we presented and tested are qualitatively the same, however, and the insights we gained from XGBoost are applicable to the more complex classifiers.

4. Results

4.1. Latent space visualization using UMAP

We visualize the latent space embedding using UMAP (Uniform Manifold Approximation and Projection; McInnes et al. 2018) to project the 512-dimensional embeddings into a 2-d space in Figure 5. We train the UMAP model solely on the SDSS test set to obtain the corresponding embeddings and apply it then to each simulation to visualize the relative positions of simulated data and SDSS test set. By doing so we can get an intuition of the gap between the simulation models and reality.

From Figure 5 it is clear that the embeddings from the 6 simulation models overlap only with a small part of the SDSS test data which implies that all simulation models can only explain a small fraction of observed galaxies. The difference in simulation models that appear are mainly due to differences in galaxy populations and not in physical resolution of the simulation models. For example, IlustrisTNG50 models a smaller volume compared to IlustrisTNG100 and hence misses a couple massive galaxies that are included in IlustrisTNG100. The same is true for NIHAO-UHD vs. NIHAO-noAGN.

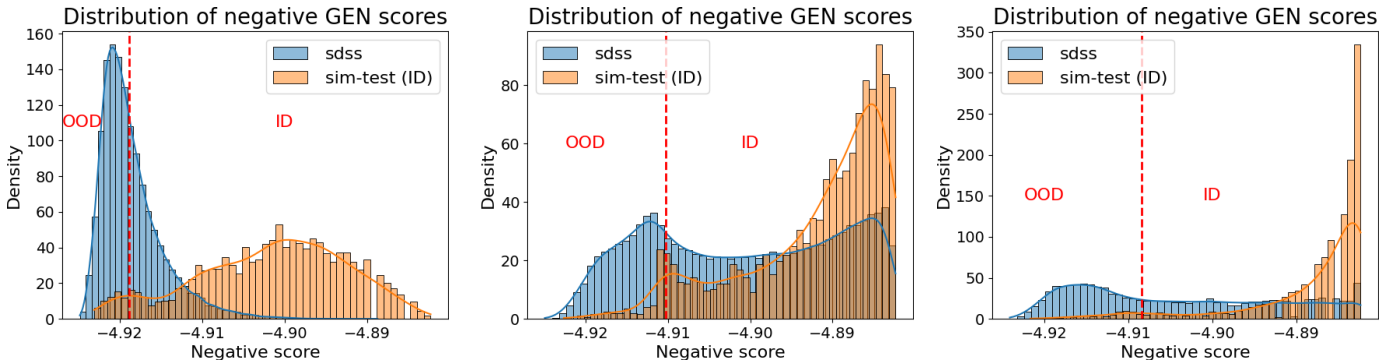
This is further confirmed by the GEN score distributions which are generally of different shape for the observational dataset and the simulation models (see Section 4.2). This result suggests that all simulation models are somewhat mis-specified and our approach opens up various avenues to improve upon the simulation models using explainable AI methods, such as SHAP values (see our results in Section 4.4).

4.2. Detection of a model mis-specification via OOD

As explained in Section 3.5 we used GEN scores for OOD detection. The resulting figures for the GEN score distributions are shown in Figure 6. As we described before, we compute GEN score distribution of SDSS test set (blue distribution) and simulation model test set (orange distribution). We define the threshold as the percent point corresponding to 95% (for a sensitivity analysis on this threshold see Appendix B) of the reference GEN score distribution. More specifically, SDSS embeddings with GEN score left to this threshold on the Figure 6 are detected as OOD data and right to this threshold are defined as in-distribution data. The fraction of OOD data for each classifier show 55% in-distribution and 45% OOD SDSS data for the stacking-MLP-RF-XGB classifier, 42% in-distribution and 58% OOD SDSS data for the random forest, and 71% in-distribution and 29% OOD SDSS data for XGBoost.

We note that the GEN score distributions differ across classifiers, which can be attributed to differences in their probability calibration behavior. Random forests often produce overconfident class assignments due to their discrete tree-based structure, while XGBoost tends to yield more flexible but sometimes still overconfident predictions. The stacked ensemble classifier combines multiple base estimators, leading to better-calibrated probabilities and smoother GEN score distributions. Since the GEN score is computed from the class probability outputs, such calibration differences naturally propagate into differences in OOD detection outcomes. Importantly, despite these differences, the relative model ranking and classification outcomes remain qualitatively stable across classifiers and all 3 ensemble classifiers show a consistently high OOD level, indicating that all simulation models are somewhat mis-specified. Since we focus here on model comparison an in depth analysis of the detailed origin of this model mis-specification is beyond the scope of this paper and will be investigated in future work. This high OOD fraction can have several reasons beyond just physical model inaccuracies, however. A mismatch in redshift range, size, SFR or stellar mass can also lead to a high OOD fraction and we deliberately decided to not apply hand-crafted cuts on these parameters. Instead, we apply nonparametric cuts via the GEN score to allow for a more nuanced selection of galaxy images for the model comparison task.

We note, however, that a high OOD fraction does not necessarily imply that the simulation models are physically inaccurate. It may also arise due to limited coverage of the observational parameter space by the simulations. For instance, our simulated galaxy images were all generated at a fixed redshift of 0.109, matching the mean redshift of the SDSS sample. As a result, the simulations lack galaxies with smaller apparent sizes that would occur at higher redshifts. Additionally, the selection functions in stellar mass and SFR differ between the SDSS and the simulations. While we deliberately avoid imposing matching cuts in these parameters – favoring instead a latent-space-based,nonparametric OOD filtering – we acknowledge that this design choice contributes to the observed simulation gap. A promising direction for future work would be to forward-model simulated galaxies across the full redshift range of SDSS by varying their apparent size and observational conditions accordingly. Such an approach may enhance the overlap in latent space and reduce the OOD fraction.



(a) random forest (b) XGBoost (c) stacking-MLP-RF-XGB

Fig. 6. GEN score distribution of all simulation model test sets (orange) and the SDSS test set (blue). Left: random forest. Middle: XGBoost. Right: stacking-MLP-RF-XGB. The vertical red line represents the 95% threshold. The region to its right corresponds to in-distribution data, and the region to its left indicates OOD data.

The in-distribution SDSS subset selected via GEN score thresholding is not based on morphology or color alone, but rather reflects latent features learned from the full image – including structural, photometric, and observational characteristics. The filtered sample thus spans a range of galaxy types that are well-represented in the simulations, rather than exclusively smooth or quiescent galaxies. This avoids unfairly favoring certain models and preserves diversity within the evaluation set.

4.3. Amortized Bayesian model comparison

Applying our Bayesian model comparison pipeline to the in-distribution dataset of the SDSS embeddings, we derive our final result shown in Figure 7. For all classifiers explored here we find rather low model probabilities for AGN, TNG50, UHD and n80. For no AGN and TNG100 we find a bimodal distribution with varying strength depending on the classifier. The bimodal shape results from the fact that the discarded OOD SDSS test data are primarily those classified between 40% and 60% as either NOAGN or TNG100, indicating that it is difficult for the classifier to make a definite prediction for them (see also Appendix B).

There is a clear preference for the NOAGN model by all 3 classifiers considering the violin shape and the position of box-and-whisker plot inside each "violin". This relative preference does not necessarily mean that NOAGN fits the SDSS test set best in an absolute sense. It simply points to the fact that, among all mis-specified models, NOAGN generates the most realistic images when compared to all other models. Hence, the model comparison we perform here is a relative comparison across all participating models.

Note though that also a tiny fraction of the TNG100 and UHD galaxies are well in agreement with SDSS. Interestingly, comparing TNG100/TNG50 and NOAGN/UHD, we find that higher physical resolution does not necessarily provide better agreement with observations. This might reveal a mismatch between simulation resolution and the employed subgrid physics which might fail to result in realistic simulations if not adapted for higher resolution.

4.4. Physical insights from the SHAP analysis

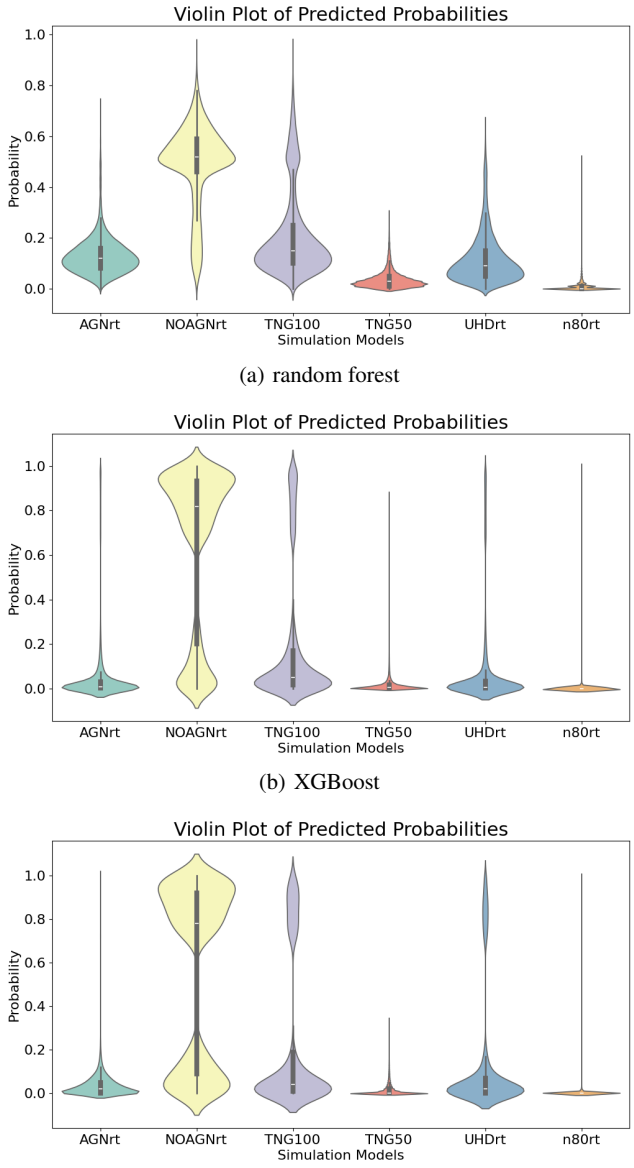
As mentioned in Section 3.7, we use TreeSHAP from the SHAP software to explain predictions from the XGBoost classifier to gain physical insights into our model comparison results. This should further give us an intuition about the influence of different features on the classification results also for other classifiers explored in this work.

Two examples of the resulting SHAP plots are shown in Figure 8 additional plots for all other models are shown in Appendix C. The overall figure structure is as follows: in the left hand side each row shows a visualization of generated galaxy images varying a single feature value in each column. The middle column indicated by a red title shows the reconstruction from an encoded latent embedding of an example galaxy. To the left (right) we reduce (increase) the value of that entry in the latent embedding. We vary latent embedding entries by about 3σ around the mean in each dimension. Note, we keep every other feature value fixed per row and only vary a single feature value. Note, the variation in each dimension encodes the behavior of all encoded galaxy vectors and we display qualitatively what happens to a representative galaxy (a mean galaxy) if its encoding vector is varied within 3σ of the data distribution. Hence, the images on the left aid the interpretation of the SHAP feature vectors and are representative of the behavior on the entire dataset not just one example galaxy shown there.

The right part of Figure 8 shows the result of the SHAP analysis. Feature n in the right panel corresponds to the n-th dimension of the latent embeddings ordered by their importance on the prediction of the XGBoost classifier result. The last row in the right part refers to the impact of all other features combined. The x-axis shows the SHAP value, i.e. the impact of that feature on the prediction while the color coding shows the feature value, e.g. high (red) vs. low (blue) feature values. Each point in the distribution refers to an individual prediction from the test set.

Panel a of Figure 8 shows that feature 189 has the strongest impact on the prediction. The distribution of SHAP values and its color distribution clearly shows that feature 189 generally has a positive impact on the prediction. More nuanced, we find that if its feature value is lower (bluer) the impact is stronger and less strong in case of large feature values. It can even have a negative impact on classification in case of very large feature values (red). This result can now be interpreted by investigating the left side of this plot to gain insights into what feature 189 physically does. The visualizations of images generated by varying feature 189 show that a low (high) feature value corresponds to redder (bluer) galaxies, while at the same time, the central light concentration decreases with increasing feature value.

The physical interpretation of this finding is hence as follows: SDSS galaxies are more likely to be classified as NIHAO



(a) random forest

(b) XGBoost

(c) stacking-MLP-RF-XGB

Fig. 7. Violin plots show the classification of the in-distribution data of the SDSS test set for three classifiers. Top: random forest. Middle: XGBoost. Bottom: stacking-MLP-RF-XGB. The horizontal white line inside the box marks the median. The whole gray box shows the interquartile range (IQR), which is defined as the difference between the third quartile (Q3) and the first quartile (Q1) of a group of data. Hence it shows the middle 50% of the distribution. And the whisker is calculated as 1.5 × IQR. It displays the distribution of predicted model probabilities $p(\mathcal{M} \mid \mathcal{D}_i)$ over all data $\mathcal{D}_i$ (y-axis) for 6 simulation models (x-axis).

noAGN if their color is redder and the central light concentration is stronger. This can be interpreted that in general NIHAO noAGN galaxies are redder and more concentrated. This is completely contrary to the findings for TNG100 shown in panel b of Figure 8.

Overall, since the feature dimensions are ordered by their importance on the prediction of the XGBoost classifier result, the upper and lower panels in Figure 8 have slightly different rankings and show different feature vectors. Features 364 and 189 are similarly important for the NOAGN and TNG100 classification, however. Additionally, the importance of features 113 and 205 is also shared between the two models. Interestingly, the effect of feature 364 and 189 on the classification output is exactly opposite – for NOAGN (TNG100) these features have an overall positive (negative) impact on the classification if they show a large feature value. The meaning of these two features is that 189 strongly correlates with color, where a low feature value represents redder galaxies, and a higher value encodes bluer galaxies. Similarly, feature 364 encodes green to red galaxies in which the substructure inside the galaxy also varies with the feature value. The larger (smaller) this feature, the greater the number of red (green) spots in the galaxies. We conclude that NOAGN tends to be redder and clumpier than TNG100, which is in turn bluer and smoother by comparison. This difference might point towards different star formation histories and present day star formation rates since younger stars are on average bluer. A similar conclusion can be drawn from the other two common features 205 and 113. Hence, we conclude that BMC combined with SHAP value analysis is suited to interpret model comparison results and to gain insights into the physical meaning of the results. This offers opportunities to not only explain model results but further help to improve the physical realism of galaxy formation simulations by pointing out physical features that match or do not match specific observational data.

We like to note here that physical interpretation through SHAP values is based on interpretable and disentangled feature dimensions in the embedding space. Through our choice of a k-sparse VAE we have used a sufficiently capable encoder algorithm to disentangle the embedding features. We do not expect perfect disentanglement, however, because already from a physical perspective, we would expect some type of feature correlation as we already know, for instance, that elliptical galaxies are redder and older than spiral galaxies, which are bluer and younger.

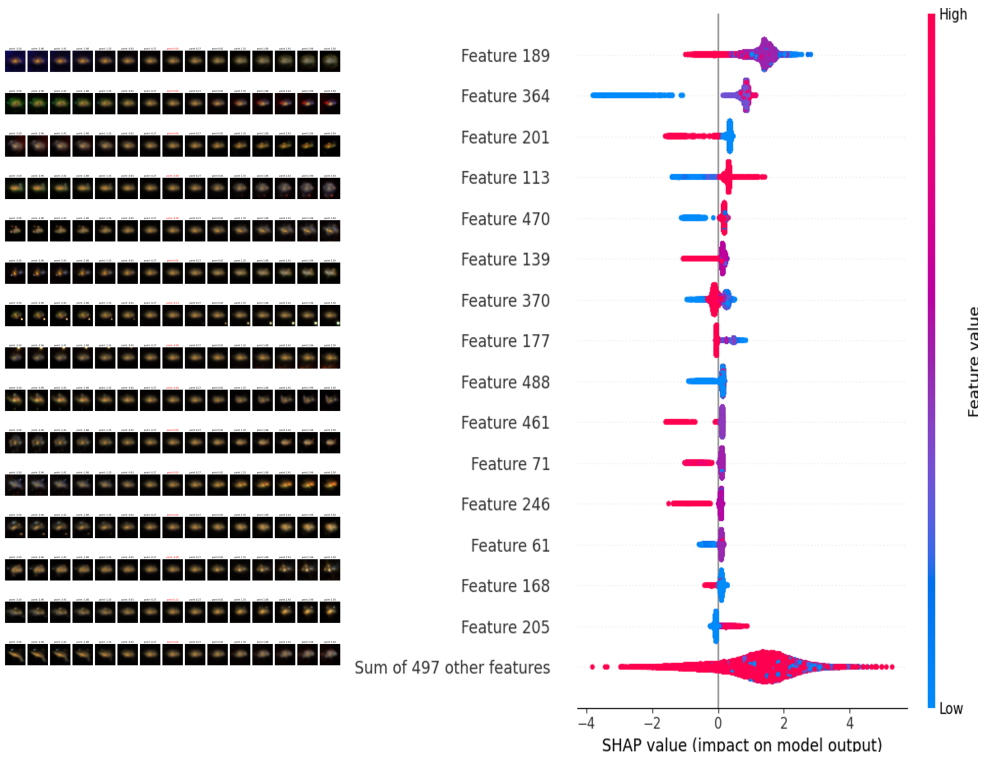
5. Discussion

While our framework provides a robust, explainable, and computationally efficient pipeline for simulation-based model comparison, several limitations and caveats merit discussion. Below we discuss in detail the effects of numerical resolution, sample size, the survey realism and our model choices.

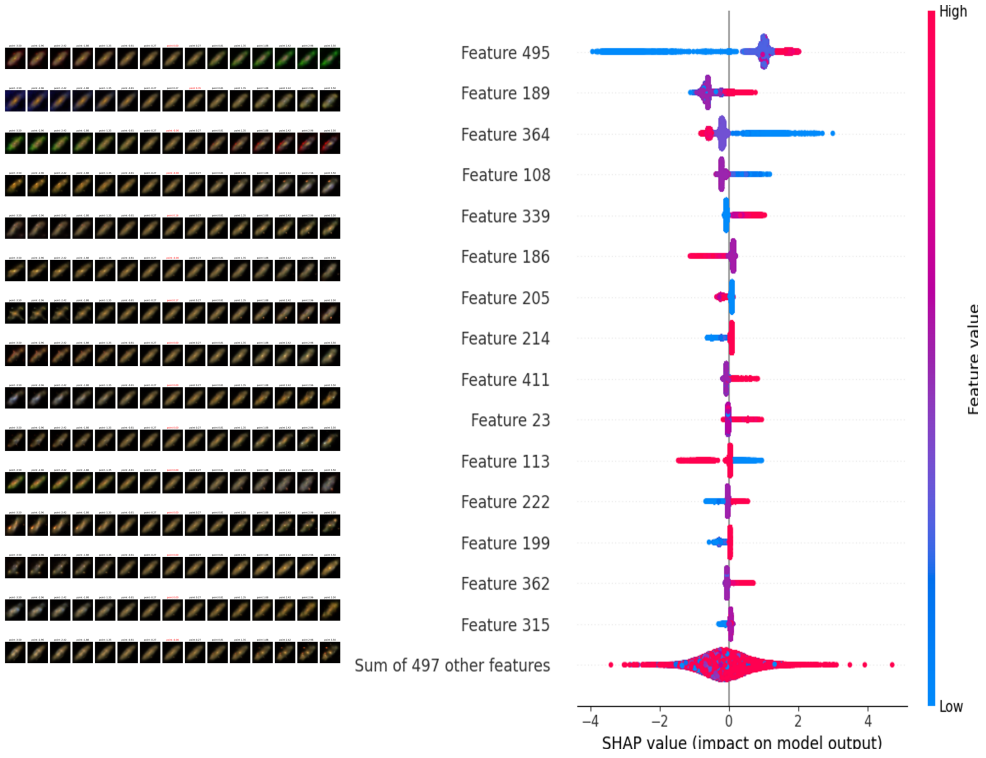
Numerical effects and resolution sensitivity. Our approach is inherently image-based and thus may be sensitive to both image and simulation resolution. All images here are resampled to a fixed resolution of 64×64 to ensure comparability with the SDSS data. This standardization may suppress fine-grained morphological details, especially for higher-resolution simulations. We mitigated potential biases via a k-sparse VAE, however, which focuses on robust, sparsity-induced features that are less sensitive to pixel-level artifacts. Additionally, we focussed on methodological advancements here and our method will equally be applicable to future higher resolution observational images.

Sample size. We note that while our framework supports model comparison under limited simulation budgets, a minimal number of a few hundred simulation images per class is recommended to ensure classifier calibration and reliable posterior estimates. The term "sparse" in our title reflects both this constraint and our use of sparse latent representations in the VAE architecture.

Similarly, we include simulation variants of different resolutions (e.g., TNG50 vs. TNG100, NIHAO-UHD vs. NOAGN), and our classifier's ability to partially separate these supports robustness to physical mass resolution changes. In fact,



(a) NOAGN



(b) TNG100

Fig. 8. SHAP plots for the XGBoost classifier. Top: NOAGN. Bottom: TNG100. Left: visualization of the generated galaxy images in which a single feature value is varied in each column. The middle column indicated by a red title shows the reconstruction from an encoded latent embedding of an example galaxy. To the left (right) we reduce (increase) the value of that entry in the latent embedding varying it by $\sim 3\sigma$ around the mean in each dimension. Right: n-th feature dimension of the latent embeddings ordered by their importance on the prediction of the XGBoost classifier result. The x-axis shows the SHAP value, i.e. the impact of that feature on the prediction while the color coding shows the feature value, e.g., high (red) vs. low (blue) feature values. Each point in the distribution refers to an individual prediction from the test set. For example, features 189 (top) and 495 (bottom) show strong color variation while features 470 (top) and 205 (bottom) show strong structural variation.

misclassification patterns (e.g., some confusion between TNG50 and TNG100) are consistent with the models' intrinsic similarity and not indicative of methodological weaknesses.

Survey realism and OOD robustness. All simulation images undergo a consistent radiative transfer and noise-injection pipeline, designed to closely replicate SDSS observing conditions. This includes PSF convolution, noise injection, and color mapping, following the RealSim (Bottrell et al. 2019) procedures as discussed in Section 3. Nonetheless, real survey effects such as background contamination (e.g., stars, cosmic rays) are not explicitly modeled. Our OOD detection strategy (via GEN scores) explicitly removes data that deviate in *any* latent dimension – be it physical properties or observational artifacts – thus making our method agnostic to such nuisances. In practice, this ensures that only comparable (in-distribution) galaxies are evaluated, boosting the reliability of model comparison.

Model choice and representation learning. Our results rely on a stacking ensemble for classification and a k-sparse VAE for dimensionality reduction. Both choices are motivated by performance and interpretability. The ensemble method increases robustness under data scarcity, while the sparse VAE ensures disentanglement for physical interpretability (via SHAP values). While different model choices might yield slightly varied results, we have tested alternatives (e.g., different VAE architectures, varying VAE sparsity and loss functions) and found qualitative stability in conclusions. In particular, we extensively compared different classifiers to inspect the sensitivity of our result on the exact architecture. In all our experiments we have found our results to be qualitatively robust against any changes in the setup. Moreover, our explainable framework allows transparent inspection of learned features, enhancing interpretability and reproducibility.

6. Summary and conclusions

We have set out to explore novel approaches for a model comparison in the context of galaxy images and hydrodynamical simulations. To to this, we developed novel methods for detecting a model mis-specification and comparing Bayesian models. By casting the Bayesian model comparison task as a classification task, we were able to select the relatively best-matching model without the need for potentially lossy hand-crafted summary statistics. Furthermore, our approach enables the usage of explainable AI techniques, such as SHAP values, for a deeper insight into the advantages and disadvantages of individual models. Additionally, our innovative approach for detecting a model mis-specification not only enabled us to gauge the misfit of individual models, but also enabled insights into why or in which respect these models were insufficient to model key physical aspects of galaxies.

In detail, our model mis-specification detection and Bayesian model comparison pipeline employs k-sparse VAEs as compression algorithms and performs the model comparison based on classifier networks in the lower dimensional embedding space. The model mis-specification detection is made via Generalized ENtropy scores before the model comparison is performed. Our approach is especially well suited in cases where it is expensive to calculate simulation models, and hence, where training data are scarce. We not only derived these methods and their corresponding network architectures, but further provided thorough guidelines on how to calibrate classifiers to derive robust results.

Finally, we used these new methods to perform a quantitative comparison between various current hydrodynamical simulations taken from the IlustrisTNG project and the NIHAO project. We gauged them against ~600 000 real SDSS galaxy images.

Our results are summarized below.

- We combined a novel simulation-based Bayesian model comparison with a new mis-specification detection technique to compare simulated galaxy images of six hydrodynamical models from the NIHAO and IllustrisTNG simulations against real observations from the Sloan Digital Sky Survey (SDSS). Thereby, we addressed the typical problem of low simulation budgets by first training a k-sparse variational autoencoder (VAE) on the abundant observational dataset of the SDSS images. The VAE learned to extract informative latent embeddings and delineated the typical set of real images (see Figure 2 for a flow chart of our method).
- Typically, hydrodynamical simulations might not explain every detail of real-world galaxy images. To reveal these simulation gaps, we then performed an out-of-distribution (OOD) detection based on the logit functions of classifiers trained on the embeddings of simulated images and found that all simulation models tested here were partially OOD (see, e.g., Figures 5 and 6).
- We then performed an amortized Bayesian model comparison using probabilistic classification only on the in-distribution data to ensure reliable comparison results. Our comparison approach was able to quantitatively identify the relatively best-performing model (see Figure 7) along with partial explanations through SHAP values (see Figure 8).
- We found that all six simulation models we tested were mis-specified compared to real SDSS observations and were only able to explain part of the reality. The relatively best-performing model comes from the standard NIHAO simulations without active galactic nucleus physics (see Figure 7). Based on our inspection of the SHAP values, we found that the main difference between NIHAO and IllustrisTNG is given by color and morphology. NIHAO is redder and clumpier than IllustrisTNG (see Figure 8).

In conclusion, we developed and tested an innovative method for a simulation-based Bayesian model comparison with a new mis-specification detection technique to compare simulated galaxy images produced by hydrodynamical models against real galaxy observations. By additionally using explainable AI methods such as SHAP values, we were able to gain physical intuition about our results and were able to explain which physical aspects of a particular simulation caused it to match real observations better or worse. This unique feature helps to inform simulators to improve their simulation model.

Data availability

We publicly release our code via Github: `https://github.com/z01ly/model-comparison`. For running our pipeline, see `src/main_func.py` in the repository.

Acknowledgements. This project was made possible by funding from the Carl Zeiss Stiftung.

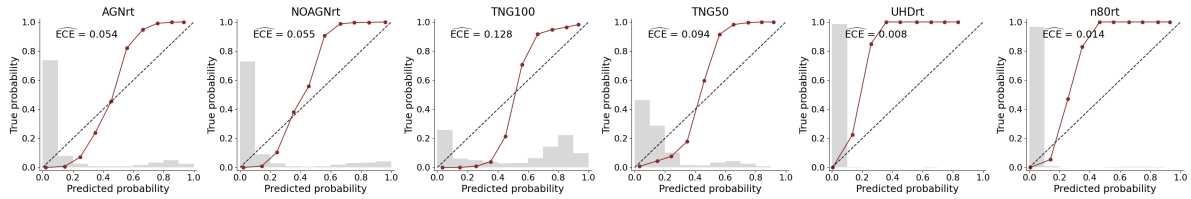
References

Akcay, S., Atapour-Abarghouei, A., & Breckon, T. P. 2019, in Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14, Springer, 622

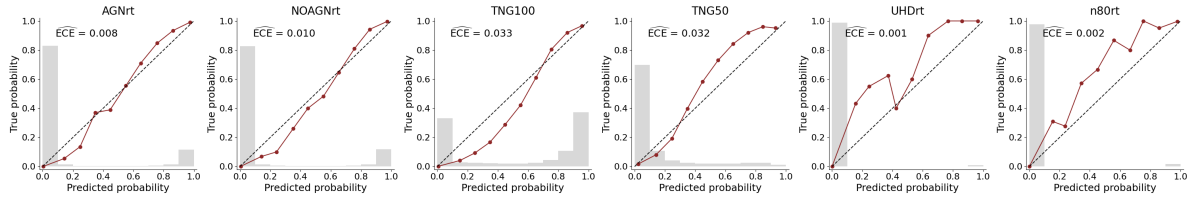
Ardizzone, L., Kruse, J., Wirkert, S., et al. 2018, arXiv e-prints [arXiv:1808.04730]

Arjovsky, M., Chintala, S., & Bottou, L. 2017, arXiv e-prints [arXiv:1701.07875]
Behroozi, P., Wechsler, R. H., Hearin, A. P., & Conroy, C. 2019, MNRAS, 488, 3143
Bignone, L. A., Pedrosa, S. E., Trayford, J. W., Tissera, P. B., & Pellizza, L. J. 2020, MNRAS, 491, 3624
Blank, M., Macciò, A. V., Dutton, A. A., & Obreja, A. 2019, MNRAS, 487, 5476
Bottrell, C., Torrey, P., Simard, L., & Ellison, S. L. 2017a, MNRAS, 467, 1033
Bottrell, C., Torrey, P., Simard, L., & Ellison, S. L. 2017b, MNRAS, 467, 2879
Bottrell, C., Hani, M. H., Teimoorinia, H., et al. 2019, MNRAS, 490, 5390
Breiman, L. 2001, Mach. Learn., 45, 5
Buck, T., & Wolf, S. 2021, arXiv e-prints [arXiv:2111.01154]
Buck, T., Ness, M. K., Maccio, A. V., Obreja, A., & Dutton, A. A. 2018, ApJ, 861, 88
Buck, T., Dutton, A. A., & Macciò, A. V. 2019a, MNRAS, 486, 1481
Buck, T., Macciò, A. V., Dutton, A. A., Obreja, A., & Frings, J. 2019b, MNRAS, 483, 1314
Buck, T., Ness, M., Obreja, A., Macciò, A. V., & Dutton, A. A. 2019c, ApJ, 874, 67
Buck, T., Obreja, A., Macciò, A. V., et al. 2020, MNRAS, 491, 3461
Buck, T., Rybizki, J., Buder, S., et al. 2021, MNRAS, 508, 3365
Buder, S., Sharma, S., Kos, J., et al. 2021, MNRAS, 506, 150
Camps, P., & Baes, M. 2015, Astron. Comput., 9, 20
Castander, F. J. 1998, Astrophys. Space Sci., 263, 91
Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. 2002, J. Artif. Intell. Res., 16, 321
Chen, T., & Guestrin, C. 2016, in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 785
Cheng, T.-Y., Huertas-Company, M., Conselice, C. J., et al. 2021, MNRAS, 503, 4446
Conselice, C. J. 2003, ApJS, 147, 1
Courteau, S., Dutton, A. A., Van Den Bosch, F. C., et al. 2007, ApJ, 671, 203
Cranmer, K., Brehmer, J., & Louppe, G. 2020, Proc. Natl. Acad. Sci., 117, 30055
De Graaff, A., Trayford, J., Franx, M., et al. 2022, MNRAS, 511, 2544
Dieleman, S., Willett, K. W., & Dambre, J. 2015, MNRAS, 450, 1441
Dubois, Y., Pichon, C., Welker, C., et al. 2014, MNRAS, 444, 1453
Dutton, A. A., Obreja, A., Wang, L., et al. 2017, MNRAS, 467, 4937
Dutton, A. A., Macciò, A. V., Buck, T., et al. 2019, MNRAS, 486, 655
Dutton, A. A., Buck, T., Macciò, A. V., et al. 2020, MNRAS, 499, 2648
Eisert, L., Bottrell, C., Pillepich, A., et al. 2024, MNRAS, 528, 7411
Elsemüller, L., Olischläger, H., Schmitt, M., et al. 2023a, arXiv e-prints [arXiv:2310.11122]
Elsemüller, L., Schnuerch, M., Bürkner, P.-C., & Radev, S. T. 2023b, arXiv e-prints [arXiv:2301.11873]
Faucher, N., Blanton, M. R., & Macciò, A. V. 2023, ApJ, 957, 7
Freeman, P., Izbicki, R., Lee, A., et al. 2013, MNRAS, 434, 282
Gneiting, T., & Raftery, A. E. 2007, J. Am. Stat. Assoc., 102, 359
Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al. 2020, Commun. ACM, 63, 139
Grogin, N. A., Kocevski, D. D., Faber, S. M., et al. 2011, ApJS, 197, 35
Haardt, F., & Madau, P. 2012, ApJ, 746, 125
Jin, Z., Macciò, A. V., Faucher, N., et al. 2024, MNRAS, 529, 3536
Karchev, K., Trotta, R., & Weniger, C. 2023, arXiv e-prints [arXiv:2311.15650]
Kingma, D. P., & Ba, J. 2014, arXiv e-prints [arXiv:1412.6980]
Kingma, D. P., & Welling, M. 2013, arXiv e-prints [arXiv:1312.6114]
Koekemoer, A. M., Faber, S. M., Ferguson, H. C., et al. 2011, ApJS, 197, 36
Kollmeier, J., Anderson, S., Blanc, G., et al. 2019, Bull. Am. Astron. Soc., 51, 274
Liu, X., Lochman, Y., & Zach, C. 2023, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 23946
Lotz, J. M., Primack, J., & Madau, P. 2004, AJ, 128, 163
Lundberg, S. M., & Lee, S.-I. 2017, Advances in neural information processing systems, 30
Lundberg, S. M., Erion, G., Chen, H., et al. 2020, Nat. Mach. Intell., 2, 56
Lupton, R., Blanton, M. R., Fekete, G., et al. 2004, PASP, 116, 133
Macciò, A. V., Udrescu, S. M., Dutton, A. A., et al. 2016, MNRAS, 463, L69
Macciò, A. V., Ali-Dib, M., Vulanovic, P., et al. 2022, MNRAS, 512, 2135
MacKay, D. J. 2003, Information Theory, Inference and Learning Algorithms (Cambridge: Cambridge university press)
Makhzani, A., & Frey, B. 2013, arXiv e-prints [arXiv:1312.5663]
Margalef-Bentabol, B., Huertas-Company, M., Charnock, T., et al. 2020, MNRAS, 496, 2346
Marin, J.-M., Pudlo, P., Estoup, A., & Robert, C. 2018, in Handbook of Approximate Bayesian Computation (Chapman and Hall: CRC Press), 153
McInnes, L., Healy, J., & Melville, J. 2018, arXiv e-prints [arXiv:1802.03426]
Meert, A., Vikram, V., & Bernardi, M. 2015, MNRAS, 446, 3943
Moster, B. P., Naab, T., & White, S. D. M. 2013, MNRAS, 428, 3121
Moster, B. P., Naab, T., & White, S. D. M. 2018, MNRAS, 477, 1822
Nelson, D., Pillepich, A., Springel, V., et al. 2018, MNRAS, 475, 624
Nelson, D., Pillepich, A., Springel, V., et al. 2019a, MNRAS, 490, 3234
Nelson, D., Springel, V., Pillepich, A., et al. 2019b, Comput. Astrophys. Cosmol., 6, 1
Obreja, A., Macciò, A. V., Moster, B., et al. 2018, MNRAS, 477, 4915
Pakdaman Naeini, M., Cooper, G., & Hauskrecht, M. 2015, Proceedings of the AAAI Conference on Artificial Intelligence, 29
Pillepich, A., Nelson, D., Hernquist, L., et al. 2018a, MNRAS, 475, 648
Pillepich, A., Springel, V., Nelson, D., et al. 2018b, MNRAS, 473, 4077
Pillepich, A., Nelson, D., Springel, V., et al. 2019, MNRAS, 490, 3196
Planck Collaboration XVI. 2014, A&A, 571, A16
Pudlo, P., Marin, J.-M., Estoup, A., et al. 2016, Bioinformatics, 32, 859
Radev, S. T., D'Alessandro, M., Mertens, U. K., et al. 2021, IEEE Trans. Neural Netw. Learn. Syst., 34, 4903
Radev, S. T., Schmitt, M., Schumacher, L., et al. 2023, arXiv e-prints [arXiv:2306.16015]
Robert, C. P., Cornuet, J.-M., Marin, J.-M., & Pillai, N. S. 2011, Proc. Natl. Acad. Sci., 108, 15112
Rodriguez-Gomez, V., Snyder, G. F., Lotz, J. M., et al. 2019, MNRAS, 483, 4140
Santos-Santos, I. M., Di Cintio, A., Brook, C. B., et al. 2018, MNRAS, 473, 4392
Schmitt, M., Bürkner, P.-C., Köthe, U., & Radev, S. T. 2023, in DAGM German Conference on Pattern Recognition (Berlin: Springer), 541
Schosser, B., Heneka, C., & Plehn, T. 2024, arXiv e-prints [arXiv:2401.04174]
Sérsic, J. 1963, La Plata Argentina, 6, 41
Smith, M. J., Geach, J. E., Jackson, R. A., et al. 2022, MNRAS, 511, 1808
Snyder, G. F., Torrey, P., Lotz, J. M., et al. 2015, MNRAS, 454, 1886
Springel, V. 2010, MNRAS, 401, 791
Springel, V., Pakmor, R., Pillepich, A., et al. 2018, MNRAS, 475, 676
Stinson, G., Seth, A., Katz, N., et al. 2006, MNRAS, 373, 1074
Stinson, G. S., Brook, C., Macciò, A. V., et al. 2013, MNRAS, 428, 129
Storey-Fisher, K., Huertas-Company, M., Ramachandra, N., et al. 2021, MNRAS, 508, 2946
Tohill, C., Bamford, S. P., Conselice, C., et al. 2024, ApJ, 962, 164
Tully, R. B., & Fisher, J. R. 1977, A&A, 54, 661
Van Den Oord, A., Kalchbrenner, N., & Kavukcuoglu, K. 2016, in International conference on machine learning, PMLR, 1747
Vehtari, A., Gelman, A., & Gabry, J. 2017, Stat. Comput., 27, 1413
Vogelsberger, M., Genel, S., Springel, V., et al. 2014, MNRAS, 444, 1518
Vogelsberger, M., Marinacci, F., Torrey, P., & Puchwein, E. 2020, Nat. Rev. Phys., 2, 42
Wadsley, J. W., Keller, B. W., & Quinn, T. R. 2017, MNRAS, 471, 2357
Wang, L., Dutton, A. A., Stinson, G. S., et al. 2015, MNRAS, 454, 83
Watanabe, S. 2013, in Proceedings of the Workshop on Information Theoretic Methods in Science and Engineering, 90
Waterval, S., Elgamal, S., Nori, M., et al. 2022, MNRAS, 514, 5307
Yang, J., Zhou, K., Li, Y., & Liu, Z. 2024, Int. J. Comp. Vision, 1, 1213
Zanisi, L., Huertas-Company, M., Lanusse, F., et al. 2021, MNRAS, 501, 4359
Zhao, S., Song, J., & Ermon, S. 2017, arXiv e-prints [arXiv:1706.02262]

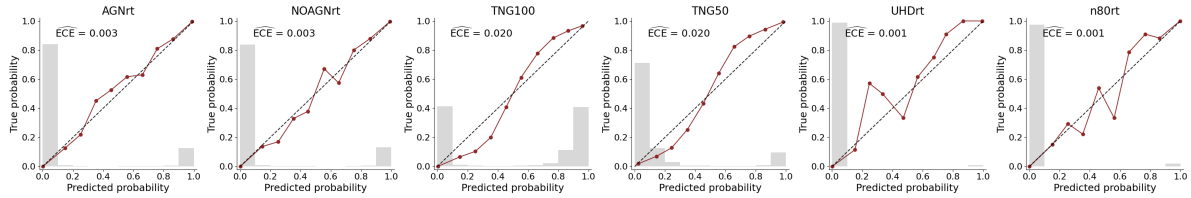
Appendix A: Classifier calibration



(a) random forest

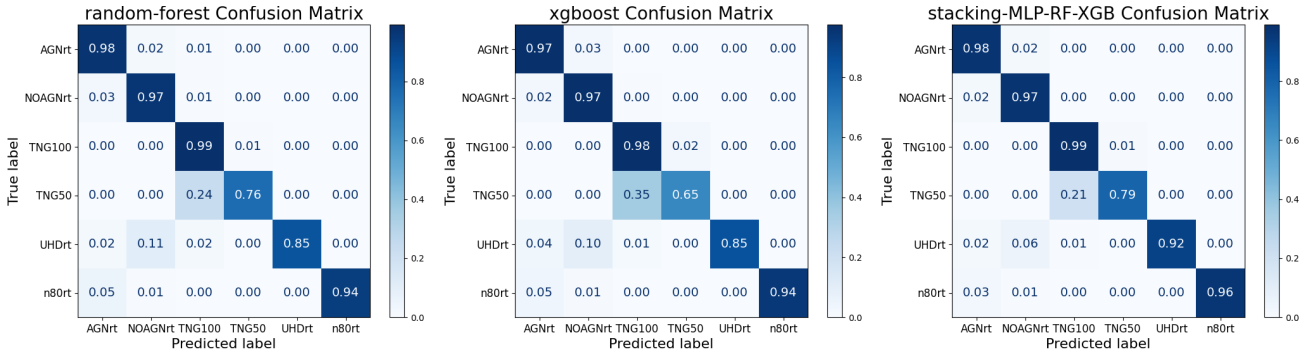


(b) XGBoost



(c) stacking-MLP-RF-XGB

Fig. A.1. Calibration curves of classifiers. Top: random forest, Middle: XGBoost, Bottom: stacking-MLP-RF-XGB



(a) random forest (b) XGBoost (c) stacking-MLP-RF-XGB

Fig. A.2. Confusion matrices of classifiers. Top: random forest, Middle: XGBoost, Bottom: stacking-MLP-RF-XGB

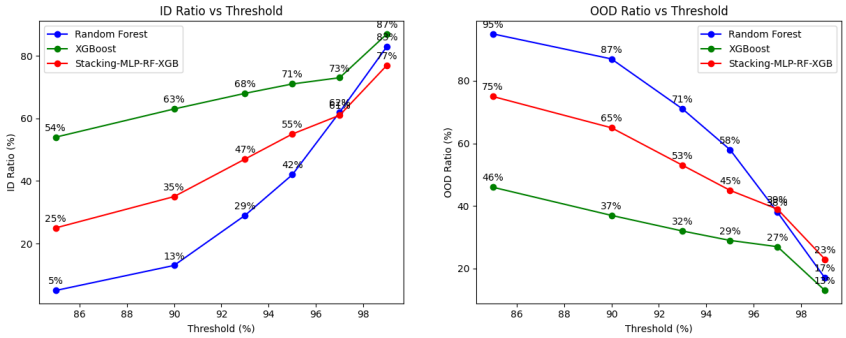
Appendix B: Sensitivity analysis of OOD threshold

Our main results are based on setting an OOD threshold of 95% in the reference GEN score distribution generated from the simulation test set. Here we adjust the threshold (99%, 97%, 95%, 93% 90% and 85%) to explore how the model comparison results and hence the violin plots and the percent of out-of-distribution SDSS test data changes accordingly.

We illustrate the relationship between threshold value and ID / OOD ratio of the random forest, XGBoost and stacking-MLP-RF-XGB classifiers in Figure B.1. The curve of random forest is the steepest one, showing that it is sensitive to the threshold change. As the threshold increases, ID / OOD ratios from the 3 ensembles tend to be more similar.

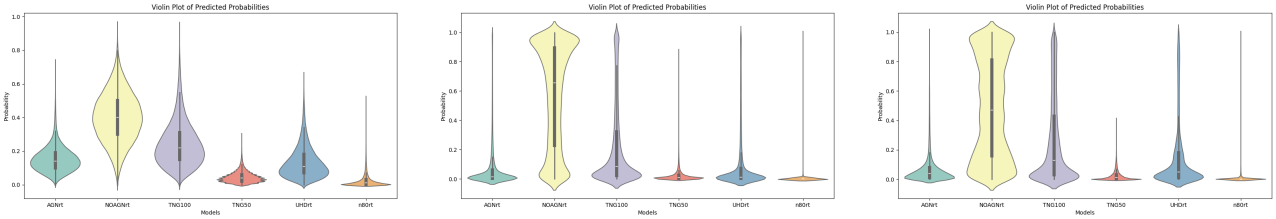
To present the relationship between threshold and our final violin plot, we use the violin plots from classifying the whole SDSS test set as a reference, see Figure B.2. The violin plots corresponding to threshold values of 99%, 97%, 95%, 93%, 90% and 85% are shown in Figure B.3, Figure B.4, Figure 7,Figure B.5, Figure B.6 and Figure B.7, respectively. Compared to the reference violin plots, discarded out-of-distribution SDSS test data are mainly those classified as 40% to 60% NOAGN or TNG100. This is reasonable since a probability around 50% implies that the classifier does not know how to handle these data. This in turn confirms OOD. We take a look at two extreme cases: for threshold 99%, to little SDSS data is discarded, while for threshold 85%, too much data are discarded as the discarded range of NOAGN has expanded from 20% to 80%.

Hence, we empirically decide for a threshold value by considering both the line chart Figure B.1 and the violin plots from different thresholds. We find that a threshold between ~93% and ~97% is reasonable – justifying our threshold choice of 95% in the main pipeline.



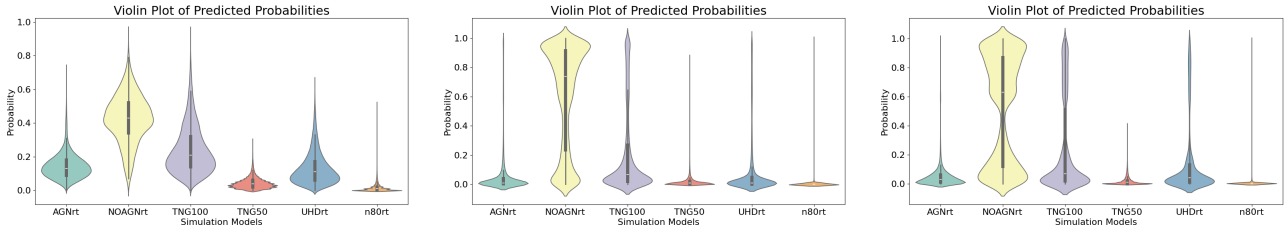
(a) threshold vs. ID ratio (b) threshold vs. OOD ratio

Fig. B.1. The relationship between threshold choice and ID / OOD ratio of random forest, XGBoost and stacking-MLP-RF-XGB.



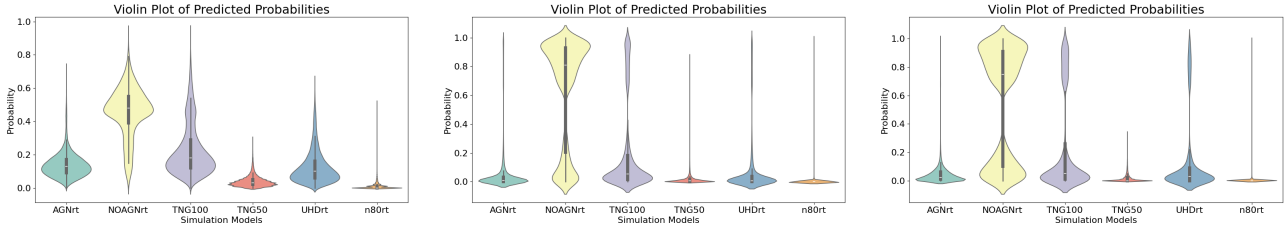
(a) random forest (b) XGBoost (c) stacking-MLP-RF-XGB

Fig. B.2. Violin plots from classifying the whole SDSS test set. Left: random forest, Middle: XGBoost, Right: stacking-MLP-RF-XGB



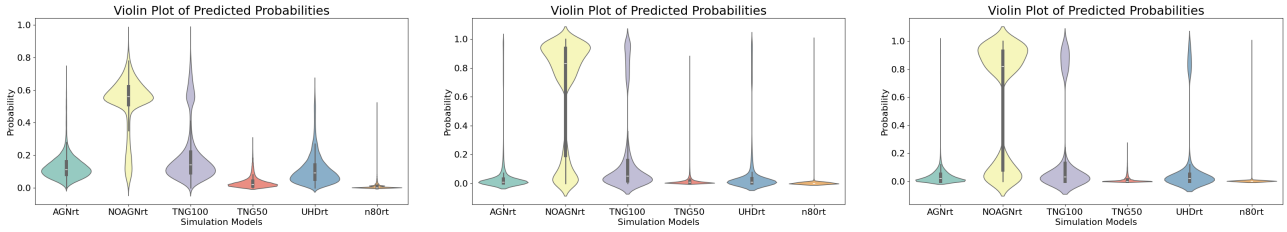
(a) random forest (b) XGBoost (c) stacking-MLP-RF-XGB

Fig. B.3. Violin plots of threshold 99%. Left: random forest, Middle: XGBoost, Right: stacking-MLP-RF-XGB



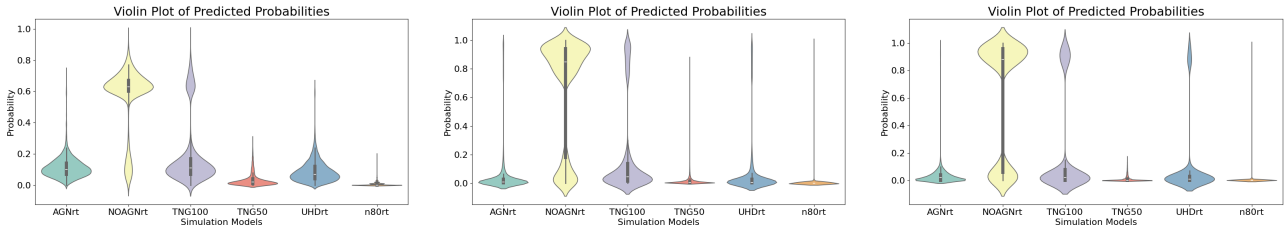
(a) random forest (b) XGBoost (c) stacking-MLP-RF-XGB

Fig. B.4. Violin plots of threshold 97%. Left: random forest, Middle: XGBoost, Right: stacking-MLP-RF-XGB



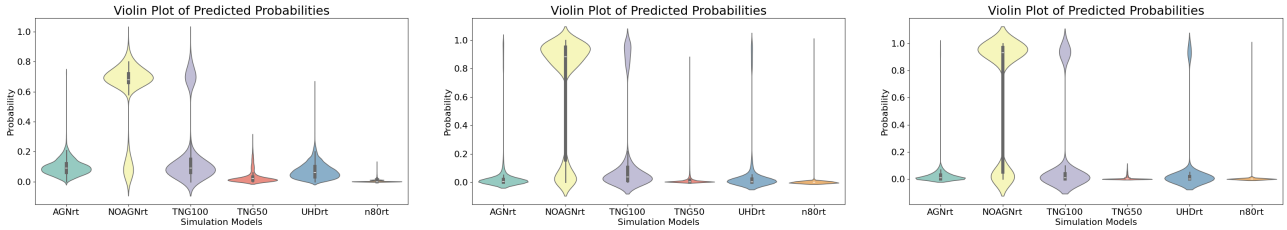
(a) random forest (b) XGBoost (c) stacking-MLP-RF-XGB

Fig. B.5. Violin plots of threshold 93%. Left: random forest, Middle: XGBoost, Right: stacking-MLP-RF-XGB



(a) random forest (b) XGBoost (c) stacking-MLP-RF-XGB

Fig. B.6. Violin plots of threshold 90%. Left: random forest, Middle: XGBoost, Right: stacking-MLP-RF-XGB

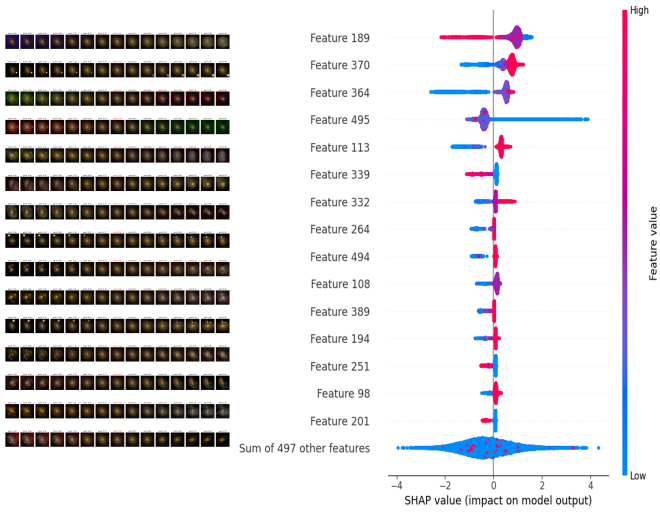


(a) random forest (b) XGBoost (c) stacking-MLP-RF-XGB

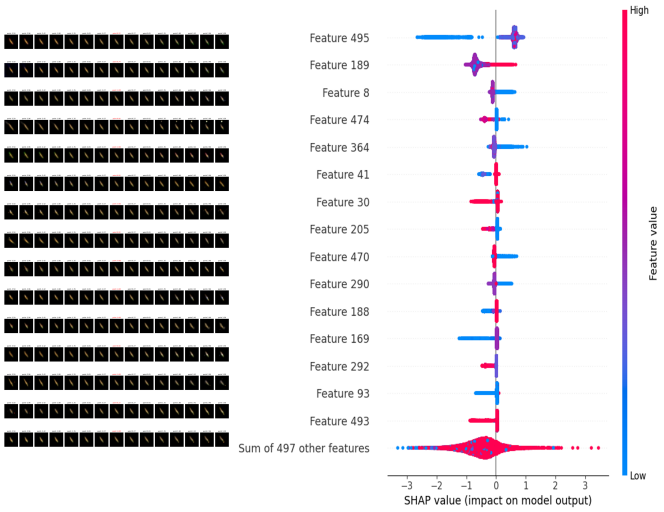
Fig. B.7. Violin plots of threshold 85%. Left: random forest, Middle: XGBoost, Right: stacking-MLP-RF-XGB

Appendix C: SHAP plots of AGN, TNG50, UHD and n80

Here we present SHAP plots of AGN, TNG50, UHD and n80 from XGBoost classifier.

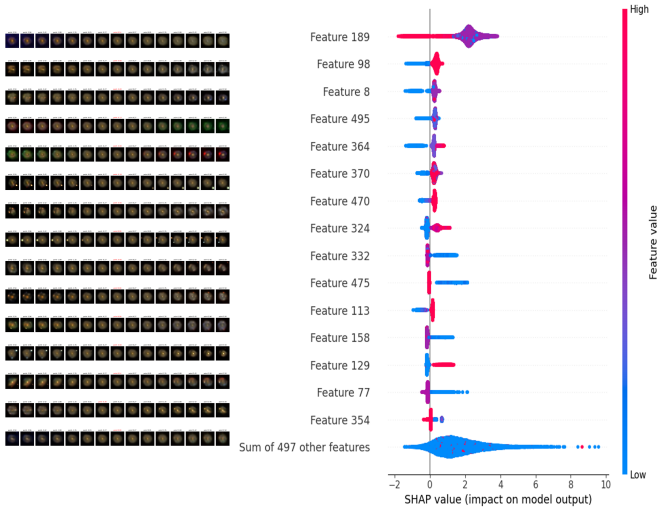


(a) AGN

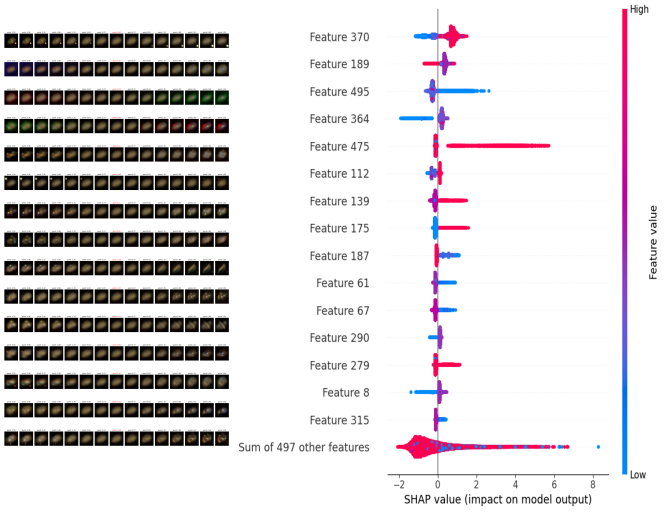


(b) TNG50

Fig. C.1. Same as Figure 8 but for AGN and TNG50



(a) UHD



(b) n80

Fig. C.2. Same as Figure 8 but for UHD and n80