AI-assisted super-resolution cosmological simulations

Yin Li[1,3]⋆, Yueying Ni[2]†, R.A.C. Croft[2], T. Di Matteo[2], S. Bird[4], Y. Feng[5]

[1] *Center for Computational Astrophysics & Center for Computational Mathematics, Flatiron Institute, 162 5th Avenue, 10010, New York, NY, USA*
[2] *McWilliams Center for Cosmology, Dept. of Physics, Carnegie Mellon University, Pittsburgh, PA 15213, USA*
[3] *Kavli Institute for the Physics and Mathematics of the Universe, University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa, Chiba 277-8583, Japan*
[4] *Department of Physics and Astronomy, University of California Riverside, 900 University Ave, Riverside, CA 92521*
[5] *Berkeley Center for Cosmological Physics and Department of Physics, University of California, Berkeley, CA 94720, USA*

ABSTRACT

Cosmological simulations of galaxy formation are limited by finite computational resources. We draw from the ongoing rapid advances in Artificial Intelligence (specifically Deep Learning) to address this problem. Neural networks have been developed to learn from high-resolution (HR) image data, and then make accurate super-resolution (SR) versions of different low-resolution (LR) images. We apply such techniques to LR cosmological N-body simulations, generating SR versions. Specifically, we are able to enhance the simulation resolution by generating 512 times more particles and predicting their displacements from the initial positions. Therefore our results can be viewed as new simulation realizations themselves rather than projections, e.g., to their density fields. Furthermore, the generation process is stochastic, enabling us to sample the small-scale modes conditioning on the large-scale environment. Our model learns from only 16 pairs of LR-HR simulations, and is then able to generate SR simulations that successfully reproduce the matter power spectrum and the halo mass function of the HR targets. We successfully deploy the model in a box 1000 times larger than the training simulation box, showing that high-resolution mock surveys can be generated rapidly. We conclude that AI assistance has the potential to revolutionize modeling of small-scale galaxy formation physics in large cosmological volumes.

Key words: methods: numerical – methods: statistical – Cosmology: large-scale structure of Universe

1 INTRODUCTION

As telescopes and satellites become more powerful, observational data on galaxies, quasars and the matter in intergalactic space becomes more detailed, and covers a greater range of epochs and environments in the Universe. Our cosmological simulations (see e.g., Vogelsberger et al. 2020 for a review) must also become more detailed and more wide ranging in order to make predictions and test the effects of different physical processes and different dark matter candidates. Even with supercomputers we are forced to decide whether to maximize either resolution, or volume, or else compromise on both. These limitations can be overcome through the development of methods that leverage techniques from the Artificial Intelligence (AI) revolution (see e.g., Russell & Norvig 2020), and make super-resolution (SR) simulations possible. In the present work we begin to explore this possibility, combining knowledge and existing super-scalable codes for petascale-plus cosmological simulations (Feng et al. 2015) with Machine Learning (ML) techniques to effectively create representative volumes of the Universe that incorporate information from higher-resolution models of galaxy formation. Our first attempts, presented here, involve simulations with dark matter and gravity only, and extensions to full hydrodynamics will follow. This hybrid approach which will imply offloading simulations to Neural Networks (NN) and other ML algorithms has the promise to enable the prediction of quasar, supermassive black hole and galaxy properties in a way which is statistically identical to full hydrodynamic models but with a significant speed up.

Adding details to images below the resolution scale (SR image enhancement) has become possible with the latest advances in Deep Learning (DL, ML with NN: Goodfellow et al. 2016), including Generative Adversarial Networks (GANs: Goodfellow et al. 2014). The technique has applications in many fields, from microscopy to law enforcement (Wang et al. 2020). It has been used for observational astronomical images by Schawinski et al. (2017), to recover galaxy features from below the resolution scale in degraded HST images. Besides SR image enhancement, DL has started to be used in cosmological simulations. For example, He et al. (2019) and Berger & Stein (2019) showed how NNs can predict the nonlinear formation of structures given simple linear theory predictions. NN models have also been trained to predict galaxies (Modi et al. 2018; Zhang et al. 2019) and 21 cm emission from neutral hydrogen (Wadekar et al. 2020), from simulations that only contain dark matter. GANs have been used (Rodríguez et al. 2018) to generate image slices of cosmological models, and to generate dark matter halos from density fields (Ramanah et al. 2019). ML techniques other than DL find many

⋆ E-mail: yinli@flatironinstitute.org
† E-mail: yueyingn@andrew.cmu.edu

applications too. For example, Kamdar et al. (2016) have applied extremely randomized trees to dark matter simulations to predict hydrodynamic galaxy properties.

Generating mocks for future sky surveys requires large volumes and high resolutions, a task that quickly becomes computationally prohibitive. To alleviate the need for such large simulations, recently, Dai & Seljak (2020) developed a Lagrangian based ML model to predict various hydrodynamical outputs from the dark matter density field. In other work, Dai et al. (2018, 2020) sharpened the particle distribution using a potential gradient descent (PGD) method starting from low-resolution simulations.

On the DL side, recently Ramanah et al. (2020) has explored using the SR technique to map density fields of LR cosmological simulations to that of the HR ones. While the goal is similar, our work has the following three key differences. First, instead of focusing on the dark matter density field, we aim to enhance the number of particles and predict their displacements, from which the density fields can be inferred. This approach allows us to preserve the particle nature of the N-body simulations, therefore to interpret the SR outputs as simulations themselves. Second, we test our technique at a higher SR ratio. Compared to Ramanah et al. (2020) that increased the number of Eulerian voxels by 8 times, we increase the number of particles and thus the mass resolution by a factor of 512. Finally, to facilitate future applications of SR on hydrodynamic simulations in representative volumes, we test our method at much smaller scales, and in big simulations whose volume is much larger than that of the training data.

Our plan for this paper is as follows. In Sec. 2 we introduce our DL approach to SR modeling in cosmology. We start with a survey of the relevant DL methods and then outline the physical nature of the datasets and algorithms we will deal with. In Sec. 2.3 we describe our specific NN architecture. We give details of the simulation data sets in Sec. 2.4, and explain how our model is trained and tested in Sec. 2.5. In Sec. 3, we show how well the SR simulation performs, both visually and statistically. Our discussion forms Sec. 5.

2 DEEP LEARNING SUPER-RESOLUTION

In this work we employ DL, the class of machine learning algorithms involving artificial neural networks, to progressively transform a LR input simulation via multiple layers of neurons into an output that statistically reproduces the HR target.

2.1 A Survey of Models

A variety of deep learning models have been applied to tackle SR tasks, ranging from early convolutional neural networks (CNN) plus simple loss function based supervised approaches (e.g. SRCNN; Dong et al. 2015), to more recent unsupervised methods such as Generative Adversarial Networks (GAN) (e.g. SRGAN; Ledig et al. 2017).

To generate SR outputs, both types of models use fully convolutional networks, which apply multiple convolution layers with learnable kernels of finite size. The difference between the two lies in their loss functions. By applying a simple loss function, e.g. the ℓ_2 or ℓ_1 norm, on the difference between the HR output and target, the first approach is a fully supervised learning task, and therefore easy to train. But the drawback is that those simple loss functions typically lead to blurry output images, due to the fact that the target almost always contains information not present in the input, i.e. high frequency features in Fourier space.

As a remedy, more complex loss functions have been used to match the output to the target on high-level features (Johnson et al. 2016). Known as perceptual loss or content loss, it feeds the output and target separately to another neural network (pre-trained on some other image data set), and compares their feature maps at intermediate layers, still by a simple loss function. Because those intermediate results contains higher-level features compared to the raw output or target, this method is able to generate less blurry images. However, such models are still deterministic and generate only one output for each fixed input, whereas in principle each input could map to infinitely many outputs due to the variability in the high-frequency modes.

The output image quality can be further improved by unsupervised learning methods, such as GAN (Ledig et al. 2017). A GAN (Goodfellow et al. 2014) is a class of DL system in which two NNs contest with each other in a game: the generative network generates candidates while the discriminative network evaluates them. Given a training set, this technique learns to generate new data with the same characteristics as the training set. GANs can be used to generate entirely new data from initially random inputs, or in the case of SR, HR images from LR ones. When the discriminator receives, in addition to SR or HR, the LR data as input, it is able to identify true and generated data conditioning on its large-scale knowledge. This technique is known as the conditional GAN (cGAN) (Mirza & Osindero 2014).

To solve the one-to-many problem in the input-to-target mapping, a model needs more than the LR input to generate non-deterministic output. Different ways of introducing random factors have been attempted. For example, the image-to-image translation model pix2pix (Isola et al. 2017) uses test-time dropout to add uncertainty to its output. More successfully, the image-generation models StyleGAN and StyleGAN2 (Karras et al. 2019b,a) have achieved state-of-the-art results on generating human faces[1]. It achieves a non-deterministic mapping by simply adding noise after every convolution layer. In our architecture, we use the same noise mechanism as in StyleGAN2 to add stochasticity to the output.

In designing our neural network model, we are inspired mostly by StyleGAN2. Even though this model is originally designed for image-generation tasks, it does so by successively upsampling some initial (constant) LR input, a process extremely similar to our SR task. We have also been influenced by the SR model SRGAN and the image-to-image translation model pix2pix.

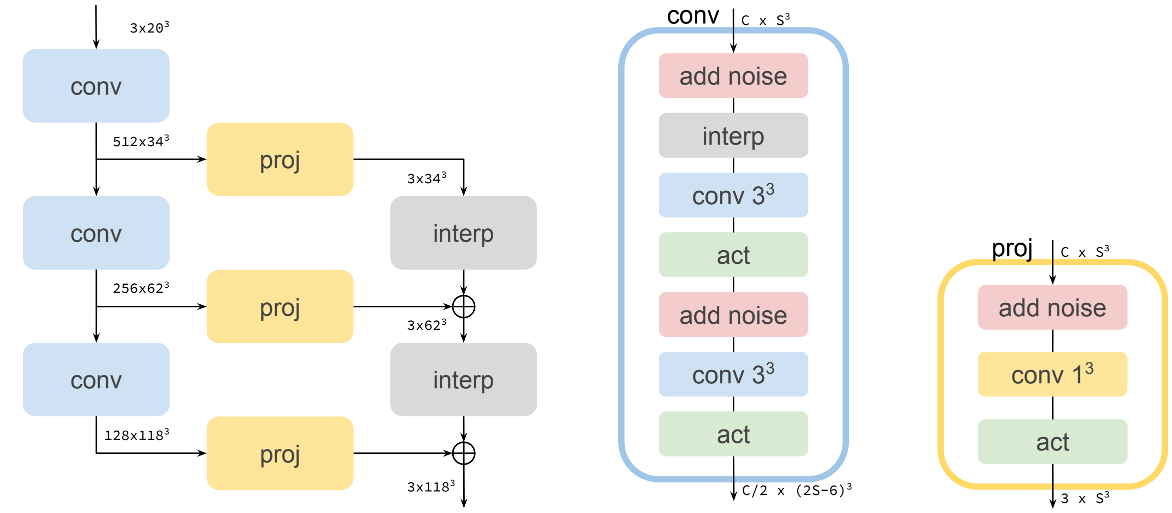
2.2 Physical considerations

We perform the SR simulation task in the Lagrangian description. Most often the particles are originally located on a uniform grid, so we can structure a displacement field from the N-body particles as a 3D image with 3 channels. Each channel corresponds to one component of the displacement vector, and the value in each voxel provides the displacement of the particle originally from that voxel.

The deep learning model takes the LR particle displacements as the input, and outputs a possible realization of their HR counterparts. Therefore, the outcome can be viewed as a higher-resolution simulation with more particles and higher mass resolution. Thus we name them the “super-resolution simulations”.

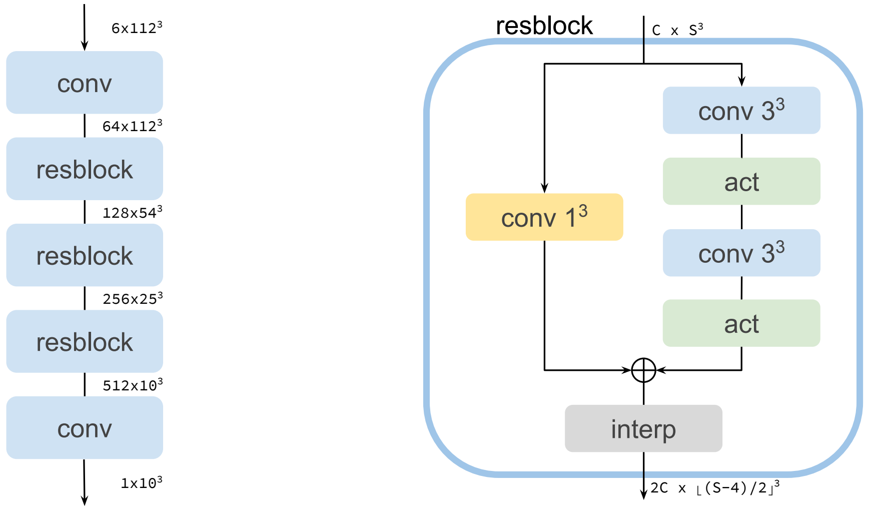
Limited by the size of GPU memory and the fact that 3D data consume more memory than lower dimensional tasks, we cannot feed the whole simulations into the GPU during training and testing. To overcome this, we crop each simulation into smaller spatial chunks. While doing so it is desirable to preserve the translational symmetry

[1] https://thispersondoesnotexist.com/



(a) generator (b) convolution block (c) projection block

Figure 1. Generator network architecture, inspired by StyleGAN2. The whole network structure is shown in (a), with components enlarged in (b) and (c). The colored plates are different operations, connected by lines or arrows from the input to the output. The sizes (channel number × spatial size) of the input, intermediate, and output tensors are next to the arrows. The generator takes the shape of a ladder, where each rung upsamples the data by 2×. The left rail consists of consecutive convolution blocks ("conv" in blue plates) operating in the high-dimensional latent space, and is projected ("proj" in yellow plates) at every step to the low-dimensional output space on the right rail. The projected results are then upsampled by linear interpolation ("interp" in gray plates), before being summed into the output. A key ingredient is the addition of noise (on red plates), that add stochasticity absent from the input at each level of resolution. The added noises are then tranformed into high-frequency features by the subsequent convolutions and activations. The kernel sizes of the convolutions are labeled in their plates (that distinguish them from the "conv" block). Note that with a kernel size 1, "conv 1^3" is simply an affine transformation along the channel dimensions, thus a convolution only in the technical sense. All activation functions ("act" in green plates) are Leaky ReLU with slope 0.2 for negative values. All "conv" blocks have the same structure as shown in (b) except the first one, which starts with an additional 1^3 convolution and an activation.



(a) discriminator (b) residual block

Figure 2. Discriminator (critic) network architecture, inspired by the StyleGAN2. The whole network structure is shown in (a), with the residual block enlarged in (b). The residual block consists of two branches: the so-called "skip" branch is only a 1^3 convolution, and on top of that the other branch has real convolutions and activations to learn the "residuals". It then downsamples by 2× the sum of the two branches with linear interpolation. Aside from the residual blocks, the first "conv" block include a 1^3 convolution followed by an activation, and the last "conv" block has a 1^3 convolution to double the channel, an activation, and a 1^3 convolution to reduce the channel to 1. See Fig. 1 for other details.

that arises naturally from the fully convolutional networks. We explain the procedure in more detail in Sec. 2.5.

In addition to translational symmetry, the equations of motion of an N-body system are also invariant under rotation. Under the periodic boundary conditions imposed in numerical cosmological simulations, the full 3D rotation group SO(3) reduces to a finite group of 48 elements, known as the octahedral group $\mathrm{O_h}$. Each group element can be decomposed into a permutation of the simulation box axes and reflections along those axes. During training, we feed the neural networks with input and output pairs that are randomly transformed by these group elements. Such data augmentation greatly enlarges the training data set, and better enforces the said symmetry in the trained model.

With many more particles in the HR simulations, they can contain small-scale information that is not present in their LR counterparts. To generate SR simulations that statistically match the HR targets, a generator neural network requires extra stochasticity in addition to that present in the LR input. Also, this additional stochasticity needs to be transformed in a way such that it results in the right correlations on different scales. To this end, we add white noise to the intermediate feature maps, with learnable amplitudes, throughout all stages of the neural network. Noise injected at early stages is later upsampled, and can then introduce correlations across multiple voxels.

2.3 Network architecture and loss function

The design of our neural networks is mostly inspired by StyleGAN2 (Karras et al. 2019a). The generator is illustrated in Fig. 1. The ladder-like structure of the generator upsamples the resolution by a factor of two at each of its rungs. See the figure captions for more architectural details.

With the LR simulations l as inputs, our generator G produces SR outputs $G(l)$ to mimic HR target simulations h. To train the generator, one simple option is to minimize the voxel-by-voxel mean squared error (related to the ℓ_2 norm) between $G(l)$ and h

$$L_2 = \mathrm{E}_{l,h}\left[\|G(l) - h\|_2\right], \tag{1}$$

where $\mathrm{E}_{l,h}$ is the expected value while sampling all LR and HR simulation pairs. With the L_2 loss, the generator model is easier and faster to train, and we can produce output tightly correlated with the HR target on large scales. However, below the input resolution scale, L_2 leads to blurry results that lack high-frequency features, as explained in Sec. 2.1.

This problem can be addressed by replacing the simple loss function with a discriminator network, which provides higher-level feedback to the generator on its performance. As shown in Fig. 2, our discriminator utilizes residual connections (He et al. 2016) that perform well on classification tasks. See the figure captions for more architectural details. Given either a generated SR or an original HR sample as input, the discriminator D learns to assign a score for it being real. In the original GAN, this score is a probability to be compared to the known values (0 or 1) using the binary cross entropy.

Wasserstein GAN (WGAN Arjovsky et al. 2017) introduced a different score which can be used to evaluate the wasserstein distance, the distance between the distributions of real and fake images by optimal transport. The discriminator (usually referred to as the critic in WGAN) is subject to constraint of it being Lipschitz continuous with a Lipschitz constant of 1, i.e. $|D(x_1) - D(x_2)| \leq |x_1 - x_2|$. Compared to the vanilla GAN, WGAN is empirically superior for it requires less tuning for training stability, and has a loss function that converges as the generated image quality improves. But it requires more computation per batch to maintain the Lipschitz constraint. This is most often achieved by adding a gradient penalty regularization term to the WGAN loss function (WGAN-gp Gulrajani et al. 2017). We train our networks using the WGAN-gp method, and only penalize the critic gradient every 16 batches for training efficiency.

Because the HR and LR images must correlate on large scales, and high-frequency features should depend on the low-frequency context, we can make the discriminator more powerful by giving it the LR images as additional input, making our model a cGAN. To achieve this we concatenate the upsampled LR images (by linear interpolation) to both the SR output and HR target, respectively, as inputs to the discriminator. The final adversarial loss function we use is

$$\begin{aligned} L_{\mathrm{WGAN-gp}} = \mathrm{E}_l\left[D(l, G(l))\right] - \mathrm{E}_{l,h}\left[D(l, h)\right] \\ + \lambda\, \mathrm{E}_{l,h}\left[\left(\|\nabla_i D(l, i)\|_2 - 1\right)^2\right], \end{aligned} \tag{2}$$

where the first line gives the Wasserstein distance and the last term is the gradient penalty. i is a random sample drawn uniformly from the line segment between pairs of real (h) and fake ($G(l)$) samples. We refer the readers to (Gulrajani et al. 2017) for more details on WGAN-gp. During training, we update the discriminator to minimize $L_{\mathrm{WGAN-gp}}$, involving all three terms, and update the generator to maximize it when only the first term takes effect.

In both the generator and the discriminator, we use convolution layers without any padding. Recall that the inputs and outputs of our networks are cropped parts of the bigger simulations, as limited by the size of GPU memory. Convolutions with zero padding or other forms of non-periodic padding break translational invariance. Because padding in the convolution layers is forbidden, the outputs of the generator are smaller than a simple factor of 8 scaling. We compensate for this by adding extra padding to the inputs of the generator. Note that this one-time padding is different from the padding in each convolution layer.

2.4 N-body Simulation Dataset

To train and validate our SR model, we use N-body simulations which only contains dark matter interacting via gravity. The dynamics of dark matter are evolved using `MP-GADGET` (Feng et al. 2015, 2016a). It is an N-body and hydrodynamics cosmological simulation code optimized to run on the most massively parallel high performance computer systems. MP-Gadget was used to run the BlueTides simulation (Feng et al. 2016a) on BlueWaters, the only cosmological hydrodynamic code that has carried out full machine runs, scaling to 648,000 cores and producing more than 6 petabytes of usable data. In all simulations the gravitational force is solved with a split Tree-PM approach, where the long-range forces are computed from a particle-mesh method and the short-range forces are obtained with a hierarchical octree algorithm.

We use 1/30 of the mean spatial separation of the dark matter particles as the gravitational softening length. The simulations have the WMAP9 cosmology with matter density $\Omega_{\mathrm{m}} = 0.2814$, dark energy density $\Omega_\Lambda = 0.7186$, baryon density $\Omega_{\mathrm{b}} = 0.0464$, power spectrum normalization $\sigma_8 = 0.82$, power spectrum spectral index $n_s = 0.971$, and Hubble parameter $h = 0.697$. We choose two snapshots at redshifts $z = 4$ and $z = 2$ from our simulations and train our model on these two redshifts separately. The former has the majority of its scales in linear and quasilinear regimes and becomes nonlinear on small scales, whereas the latter spans mostly in the nonlinear to deeply nonlinear regimes. We expect that it is harder for the model to learn to form nonlinear structures than linear ones, and these two redshifts allow us to test the performance degradation due to nonlinearity.

For training and testing we run, repectively, 16 and 1 LR-HR pairs of dark matter only simulations with box size of $(100\,h^{-1}\mathrm{Mpc})^3$, and 64^3 and 512^3 particles for LR and HR respectively. The mass resolution is $m_{\mathrm{DM}} = 2.98 \times 10^{11}\,M_\odot/h$ for LR and $m_{\mathrm{DM}} = 5.8 \times 10^8\,M_\odot/h$ for HR. So our SR task is to enhance the spatial resolution by 8× and mass resolution by 512×. We also run a $(1\ \ h^{-1}\mathrm{Gpc})^3$ LR simulation, of otherwise the same configuration as the smaller LR runs, to test deploying our model to a larger volume.

2.5 Training and Testing

Our GAN model is trained upon the displacement field in Lagrangian space. We first pre-process the training data by converting the particle position to their displacement vector with respect to the initial grid. Then for each mini-batch, we crop a 14^3 grid from the LR displacement field as input, pad 3 cells on each side to compensate for the loss of voxels during the generator convolution layers. This transforms into an SR output of size 118^3 through the generator (see sizes annotated in Fig. 1), out of which we crop the inner part to match the corresponding $(8{\times}14)^3 = 112^3$ grid from the HR target displacement field. Therefore, all the mini-batches for training are about $(22\,h^{-1}\mathrm{Mpc})^3$ in size. We then concatenate the tri-linear interpolations (8× upsampling per dimension) of the LR inputs to the SR outputs or the HR targets. The results are 6 channel images to be taken by the discriminator and transformed in a similar way as shown in Fig. 2. Data augmentation is applied for each mini-batches.

We train our neural network model using the Adam optimizer with learning rate 1×10^{-5} and exponential decay rates $\beta_1 = 0$ and $\beta_2 = 0.99$. We start with 5 epochs of supervised training with the simple loss function given in Eq.1. This allows the generator to quickly learn to generate SR outputs that are consistent with LR inputs above the input resolution. We then proceed to the adversarial training and update the generator and the discriminator alternately, for 150 epochs.

For the test set, we use a new pair of 64^3 LR and 512^3 HR simulation with same volume of $(100\,h^{-1}\mathrm{Mpc})^3$. The test set shares the same cosmological parameters as the training sets, but is a new realization with different initial conditions. Therefore our comparison is free of overfitting to the training data set. Remarkably, we are able to deploy our model to LR input of 1000× bigger volume and demonstrate the scaling ability of our approach.

The procedure of generating the SR simulation from the LR input goes as follows. We first preprocess the LR input by converting the particle position to the displacement field of shape 3×64^3. Then we use our trained GAN model to generate the SR displacement field with shape 3×512^3, and obtain the particle positions by moving them from their original positions on a lattice by these displacement vectors. The generation process from LR to SR is done in chunks due to the limit of GPU memory. We crop the LR input into pieces, generate their corresponding SR field, and stitch the output patches together to exactly match the shape of the target. The generated full SR field is periodically continuous through this procedure thanks to the translational symmetry preserved by our GAN model.

3 RESULTS

In this section, we validate our model by comparing the generated SR simulations to the authentic HR simulations both visually and quantitatively, using summary statistics including the power spectrum and the halo mass function.

3.1 Visual comparison

First let us visually compare the generator outputs to the HR simulation results. In Fig. 3, we show the dark matter density field of the LR (left), HR (middle) and SR (right) simulations at $z = 2$. The top panel shows the view of the full 100 h^{-1}Mpc box in slabs of thickness 20 h^{-1}Mpc. The middle and bottom panels shows multiple zoom-in regions of $(20\,h^{-1}\mathrm{Mpc})^3$ and $(10\,h^{-1}\mathrm{Mpc})^3$ in size to magnify the finer structures. We render 2D slices of the particle density distribution by Gaussian smoothing on a scale of 5 h^{-1} kpc using `gaepsi2`[2]. We visualize all dark matter particles in blue and highlight the FOF halos (see Sec. 3.3) in orange.

As shown in Fig. 3, our SR technique is able to recover sharp features and small-scale structures starting from inputs of extremely poor resolution. It forms halos where the input LR can not resolve them, e.g., in the figure the LR can only form the most massive handful of halos in the box whereas SR is able to improve the halo mass range by orders of magnitude. The SR outputs look statistically indistinguishable from the HR target on large scales. Only in the deeply nonlinear regime is a difference in image sharpness noticeable. This will be quantified using the power spectrum below.

Our method also allows us to map the same LR input to multiple outputs. We achieve this by adding noise between layers in the generator as explained in Sec. 2. In Fig. 4, we show 2D projections of the $(20\,h^{-1}\mathrm{Mpc})^3$ zoom-in regions (same as box A in Fig. 3) at $z = 4$ (top) and $z = 2$ (bottom). The first and second columns show the LR and HR fields, and the last three columns show some random SR variants generated from that same LR input. As expected we see that on scales immediately below the input resolution, the SR variants are consistent with each other, as well as the HR field, because environment affects how enclosed structures form. On smaller scales, the SR realizations show apparent differences among themselves and from the HR target.

The visual tests have demonstrated that our SR technique is able to not only generate small-scale structures that statistically resemble the target, but also sample them conditional on their larger-scale environment. These capabilities make our method extremely promising for a variety of other SR applications.

3.2 Power spectrum

The matter power spectrum, that describes the amplitude of density fluctuations as a function of scale, is perhaps the most commonly used summary statistic in cosmology. It describes in Fourier-space the 2-point correlation, which completely specifies the statistical properties of the Gaussian random field. And to the best of our knowledge, the cosmological density field is well represented by an initial Gaussian random field and remains so in the linear growth regime. Therefore it is standard to compare the SR and HR simulations on their matter power spectra $P(k)$.

For a periodic simulation box, the matter power spectrum can be measured by a discrete Fourier transform

$$P(k_i) = \frac{1}{N_i} \sum_{k_i < k \le k_{i+1}} \frac{\left|\delta(\boldsymbol{k})\right|^2}{V}, \tag{3}$$

where V is the simulation volume and $\delta(\boldsymbol{k})$ is the discrete Fourier transform of the overdensity field $\delta(\boldsymbol{x}) \equiv \rho(\boldsymbol{x})/\bar{\rho} - 1$, with $\rho(\boldsymbol{x})$ and $\bar{\rho}$ being the matter density field and its mean value respectively. Due to statistical isotropy, the power spectrum is only a function of

[2] https://github.com/rainwoodman/gaepsi2

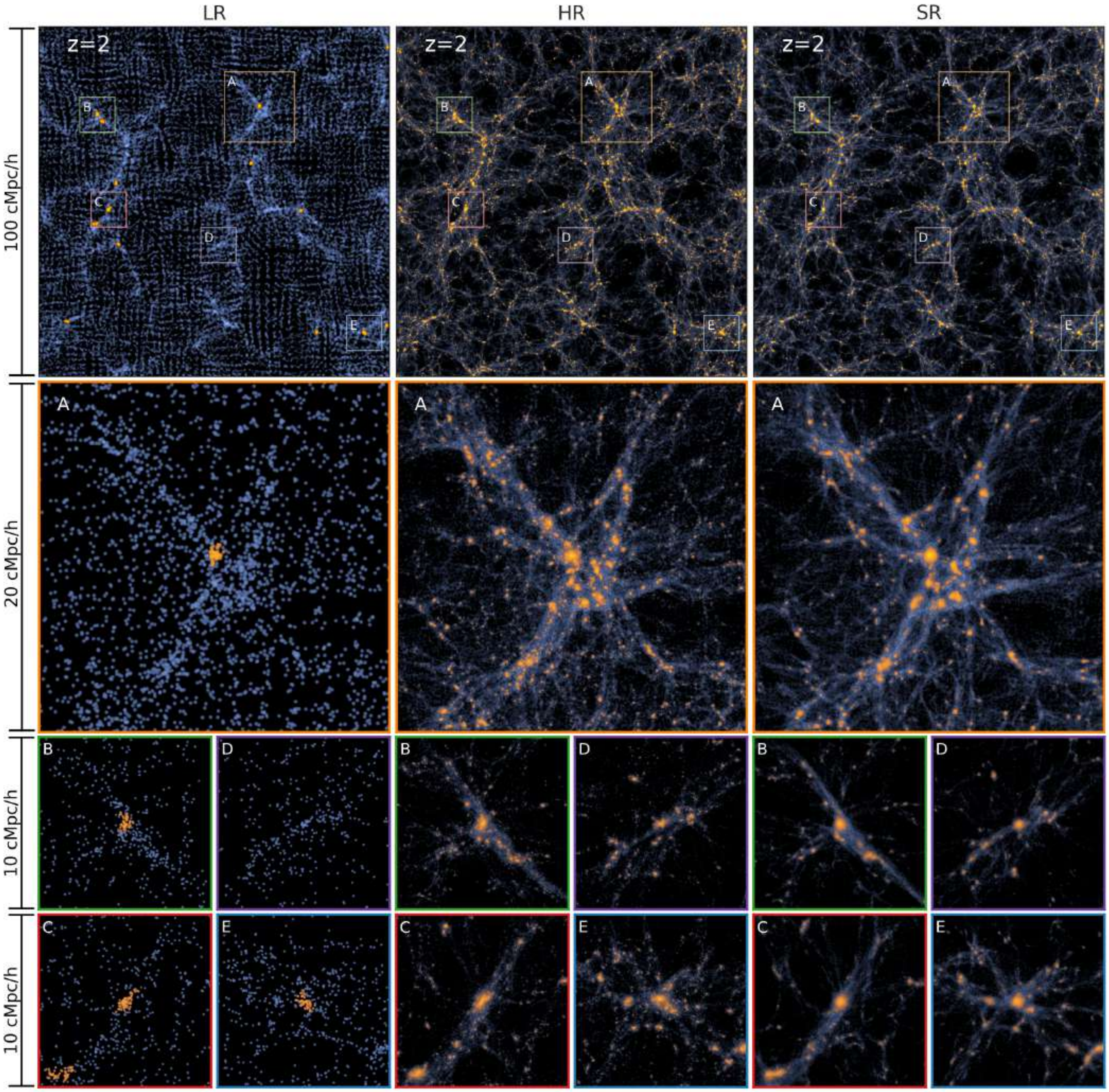


Figure 3. 2D slices of the low-, high-, and super-resolution (LR, HR, and SR) dark matter density field at $z = 2$. The blue background shows the smoothed density field of all the dark matter particles. The particles in FOF groups are highlighted in orange to help visually identify the halos. The top panels show slabs from the full box of 100 h^{-1}Mpc side length and 20 h^{-1}Mpc thickness. The middle panels zoom into the orange boxes (A) from the top panel, each of size $(20\,h^{-1}\mathrm{Mpc})^3$. The bottom two rows show the four zoom-in boxes (B, C, D, and E) which are $(10\,h^{-1}\mathrm{Mpc})^3$ in size to reveal even finer details. The first two columns show the LR and HR simulations, which have the same initial conditions but a factor of 512× different mass resolution. The rightmost column shows one of the SR realizations generated by our trained model. All density images are rendered with Gaussian smoothing on a scale of 5 h^{-1} kpc.

the wavenumber $k = |\boldsymbol{k}|$ and independent of the direction of $\boldsymbol{k}$. The above estimator has already exploited this by averaging spherically within each k-bin. N_i is the number of modes falling in the i-th bin $(k_i, k_{i+1}]$.

As explained in Sec. 2.5, we move the particles with the predicted displacement field to obtain their positions. We then compute the density field by assigning the particle mass to a 512^3 mesh using the CIC (Cloud-in-Cell) scheme, for all LR, HR, and SR simulations. A common practice in power spectrum estimation is deconvolution of the resampling window (here CIC) after the discrete Fourier transform. However, doing this leads to artifacts in the LR results. Instead, using a large FFT grid for both resolutions, we avoid deconvolving the resampling window Jing (2005) and can compare power spectra from different resolutions on an equal footing.

In Fig. 5, we compare the dimensionless power spectra $\Delta^2(k) \equiv k^3 P(k)/2\pi^2$ of LR, HR, and SR, at redshift $z = 4$ (left) and $z = 2$

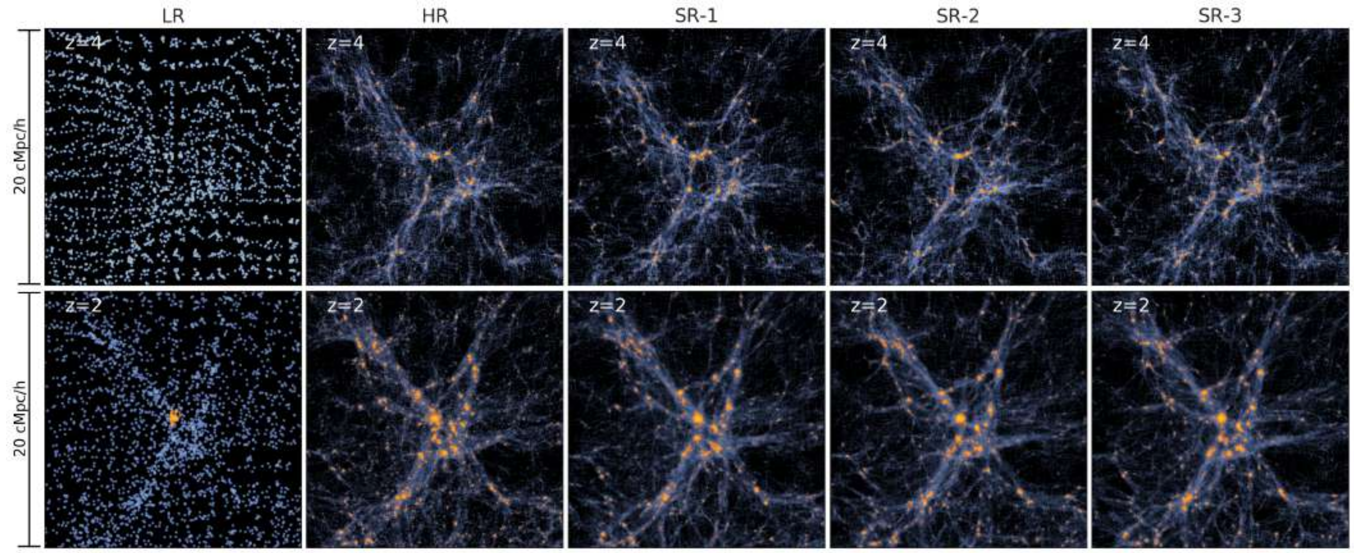


Figure 4. Using our GAN-based algorithm, we can generate different SR realizations from the same LR inputs. Here we show 2D slices of the same $(20\,h^{-1}\mathrm{Mpc})^3$ regions at $z = 4$ (top panels) and $z = 2$ (bottom panels) to illustrate the difference in the generated SR fields. The first two columns give the LR and HR fields, and the last three columns show three variants of the corresponding SR outputs. The color scheme and rendering method are same as in Fig. 3.

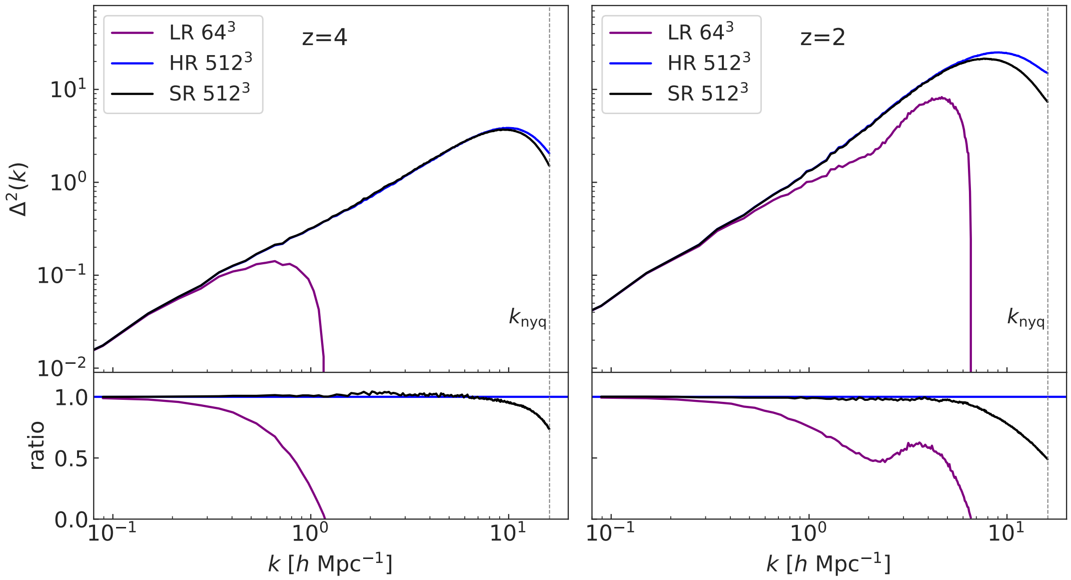


Figure 5. Dimensionless matter density power spectrum Δ^2 comparison at $z = 4$ and $z = 2$, with purple, blue and black lines representing the results of LR, HR and SR simulation from the test set. The vertical dashed lines mark the Nyquist wavenumber $k_{\mathrm{nyq}} = \pi N_{\mathrm{mesh}}/L_{\mathrm{box}}$ with $N_{\mathrm{mesh}} = 512$ and $L_{\mathrm{box}} = 100\,h^{-1}\mathrm{Mpc}$. The wavenumber k where Δ^2 crosses 1 is an indicator of the scale below (towards increasing k) which structure formation becomes nonlinear. At $z = 4$ where the majority of scales are in the linear and quasilinear regime, the SR works well everywhere except close to the Nyquist; whereas the LR power spectrum drops quickly beyond the scale limited by its mass resolution as expected. Later at $z = 2$, structure formation has grown to be much more nonlinear as can be seen from the increasing clumpiness in Fig. 4. The deficit in the LR power persists and is partly compensated by formation of poorly resolved halos, while the SR realizations are able to reproduce the power well into the nonlinear regime where $\Delta^2 \sim 20$. The SR power starts to deviate from that of HR in the deeply nonlinear regime. This can be due to limitations by the capacity of our neural network and/or the intricacy of the GAN training process, and will potentially improve in the future works.

(right). The vertical dashed lines mark the Nyquist wavenumber $k_{\mathrm{nyq}} = \pi N_{\mathrm{mesh}}/L_{\mathrm{box}}$ with $N_{\mathrm{mesh}} = 512$ and $L_{\mathrm{box}} = 100\,h^{-1}\mathrm{Mpc}$. The monotonically increasing $\Delta^2(k)$ is a good proxy for the variance of matter density as a function of scale, and thus an indicator that can be used to divide the linear and nonlinear regimes.

At $z = 4$, where the majority of the simulated scales are still in the linear and quasilinear regimes, the LR power quickly vanishes below the grid scale, while the SR power spectrum successfully matches the HR one up to $k \sim 10\ \,h\,\mathrm{Mpc}^{-1}$, within about 10%. At lower redshift $z = 2$, structure formation becomes increasingly more nonlinear, with the nonlinear scale starting around $k = 1\ \,h\,\mathrm{Mpc}^{-1}$. However the SR power is still able to closely matches the HR one to deeply nonlinear scales, at 10% accuracy to about $7\ \,h\,\mathrm{Mpc}^{-1}$ where Δ^2 is as large as 20. This is a dramatic improvement over the LR power which again deviates quickly below 0.5 h Mpc^{-1}. We also find that different SR variants have almost identical power (within $\sim 1\%$) in Fig. 5. This is because sample variance on small scales is suppressed under the large-scale constraint.

We note that the SR power starts to drop in the deeply nonlinear regime above $\Delta^2 \sim 20$, (this is associated with the decreased sharpness on the smallest scales). This is as illustrated in the visual appearance of the density fields in the zoomed-in panels of Fig. 3. The performance degradation is expected as non-linearity increases the difficulty of the learning task. Another possible cause is the pure Lagrangian description in our modeling. As explained in Sec. 2, modeling with the displacement field has several advantages, including preserving the particle nature of a simulation and being able to account straightforwardly for the nonlocal impact of the larger environment. However, this also makes predictions for variables in the Eulerian description, such as the density (used in the calculation of P(k), for example), more difficult. Matter collapses to form dense structures. There the interactions between closely spaced particles would be nonlocal in the Lagrangian description, and thus require a larger communication radius in a neural network model. We leave improvements in these aspects for future work.

3.3 Halo mass function

Beyond the 2-point correlation test, i.e. the power spectrum, we need to compare higher-order statistics of the fields, which quantify the non-Gaussianity arising in the nonlinear regime. As the most nonlinear dark matter structures, halos host most of the luminous extragalactic observables in the sky. Therefore halo abundance is a natural choice when considering non-Gaussian statistics. We find the dark matter halos in the simulations using a Friends-of-Friends (FOF) halo finder (Davis et al. 1985) with linking length parameter $b = 0.2$. The FOF algorithm links all pairs of particles within b times their mean separation, and collects each group of connected particles into one halo. We only keep halos with at least 32 particles.

In Fig. 6 we compare the halo mass functions measured in the LR, HR, and SR $(100\,h^{-1}\mathrm{Mpc})^3$ test set simulation, at $z = 4$ (left) and $z = 2$ (right). The halo mass function is defined as $\phi \equiv \mathrm{d}N/\mathrm{d}\log_{10}M_\mathrm{h}/V_\mathrm{box}$, where N is the number of halos above threshold mass M_h and V_box is the co-moving volume of the simulation. The purple, blue, and black solid lines show the LR, HR and SR results, respectively. The shaded areas give the Poisson error estimates. Due to their large particle mass and low force resolution, the LR simulations can only resolve the most massive halos up to $M_\mathrm{h} \gtrsim 10^{13}\,M_\odot$. They therefore do not contain any halos at $z = 4$ and only predict the most massive ones at $z = 2$.

Using our GAN model, the SR field generated from the LR input has the same mass resolution as the HR field, resolving halos over the whole mass range down to $M_\mathrm{h} \gtrsim 10^{11}\,M_\odot$. The generated SR fields predict halo populations that agree well with the HR results, with a $\sim 50\%$ deficit only at $z = 2$ on the lowest mass end $M_\mathrm{h} \sim 10^{11}\,M_\odot$. This underprediction of the smallest halo populations can be viewed as the result of the lack of clumpiness at the small(est) scale, as also shown in Fig. 5 and associated with the deficit of power at $k \gtrsim 10\,h^{-1}$Mpc. We suspect this lack of sharpness on small scales is due to our formulation of the loss function in pure Lagrangian description, and plan to improve on this in our future work.

4 APPLICATION TO A LARGE VOLUME

As described in Sec. 2.5, we train our GAN model on fields cropped from $(100\,h^{-1}\mathrm{Mpc})^3$ volume simulations. In the previous section, our trained model has proven to work well on a new SR realization of the same volume. Moreover, translational symmetry has been carefully preserved by our cropping and padding schemes. This implies that our method can be applied to any cosmological volumes. The additional long-wavelength modes in the large volume lead to rare massive structures that the model has not encountered in the training set. We apply our generative model to a $(1\ h^{-1}\mathrm{Gpc})^3$ LR simulation with 640^3 particles, a volume 1000 times larger than the training set simulations, and obtain an SR realization with 5120^3 particles. This generating process takes only about 16 hours on 1 GPU, including time for I/O. The processes consumes significantly less computing resources than PDE based N-body solvers.

In Fig. 7, we give an illustration of the generated $(1\ h^{-1}\mathrm{Gpc})^3$ volume SR result at $z = 2$. The large panel shows a slab of the full SR field of 1 h^{-1}Gpc in length and 20 h^{-1}Mpc in thickness. The upper left inset shows the same field of the LR field. In the two red inset panels, we also show a zoom-in region of volume $(20\,h^{-1}\mathrm{Mpc})^3$ around a massive halo in the SR and LR fields. The central halo shown in the red panels has a halo mass of $M_\mathrm{h} = 2 \times 10^{14}\,M_\odot$, larger than the most massive halo in the $(100\,h^{-1}\mathrm{Mpc})^3$ training sets. Remarkably, even though our GAN model has not been trained upon such massive halos, it performs well, and generates them with reasonable morphology, apparent substructure and correct population density (see below).

To quantitatively investigate the halo population of the SR field at $z = 2$, we use the FOF halo finder to identify dark matter halos in the input LR field and the generated SR field. The resultant halo mass functions are shown in Fig. 8. The blue solid line gives the result of the input LR field, which forms only massive halo with $M_h \gtrsim 10^{13}\,M_\odot$ at $z = 2$. The black solid line shows the corresponding SR halo abundance, agreeing remarkably well with both the $(1\ h^{-1}\mathrm{Gpc})^3$ LR and $(1\,h^{-1}\mathrm{Mpc})^3$ HR predictions. It is only slightly suppressed at the lowest mass $\sim 10^{11}\,M_\odot$ as we have already seen in Fig. 6. The SR prediction even has the right abundance for halos above $10^{14}\,M_\odot$, suggesting that our model actually learns some structure formation physics relevant at scales over which it has not been directly trained.

This exercise demonstrates the potential of our newly developed method to tackle the fundamental challenge of constructing large volume cosmological simulations at otherwise unfeasibly high resolution, with a dramatically lower footprint in computing resources. It opens up an avenue for constructing large volume cosmological (eventually hydrodynamical) simulations and creating large ensembles of associated mocks at scales comparable to current surveys, and so maximizing their scientific return.

5 DISCUSSION

We have shown how a GAN super-resolution architecture inspired by StyleGAN2 can be used to enhance cosmological simulations so that they reproduce the appearance and statistics of much higher-resolution models. Our approach has similarities to the work of Ramanah et al. (2020), who showed how SR dark matter density fields can be generated. Among the differences with this work are our use of particle displacements as the inputs and outputs of our modeling. By generating a field of SR particle displacements, we effectively create a whole simulation, which can be analysed (for example by carrying out halo finding) in the same manner as a full simulation

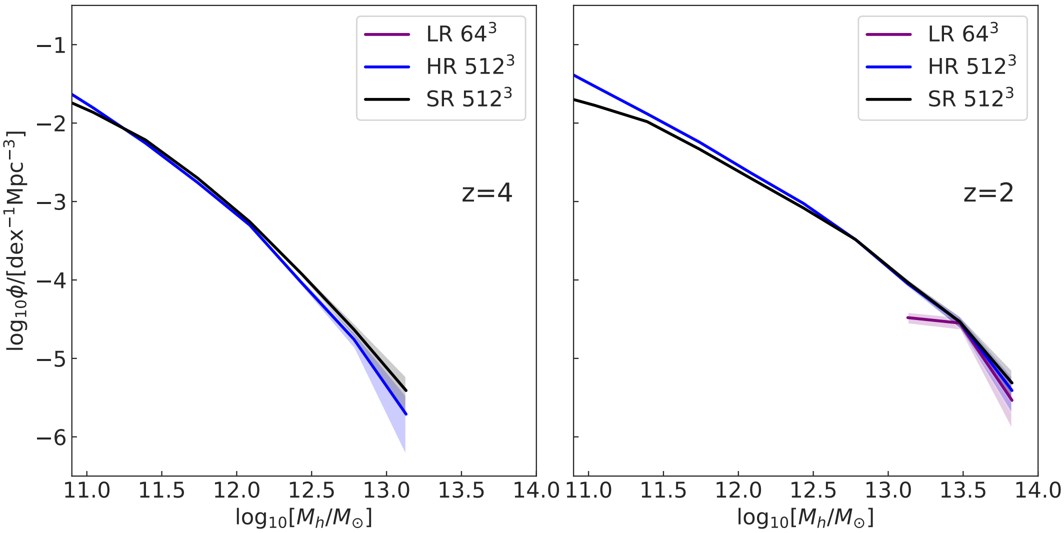


Figure 6. Halo mass function comparison at $z = 4$ (left) and $z = 2$ (right) for LR, HR and SR simulations shown by purple, blue and black solid lines, respectively. The halos are identified by the FOF halo finder with linking length parameter $b = 0.2$ and contain at least 32 particles. Due to the limited mass resolution, the LR simulations has not resolve any halos at $z = 4$, and only forms the most massive halos ($M_\mathrm{h} \gtrsim 10^{13}\ M_\odot$) at $z = 2$. Using our GAN model, the SR field generated from the LR input has the same mass resolution as the HR field, resolving a halo mass range all the way down to $M_\mathrm{h} \gtrsim 10^{11}\ M_\odot$. It predicts halo populations that closely match the HR results, only with a $\sim 50\%$ deficit for the lowest mass halos ($M_\mathrm{h} \sim 10^{11}\ M_\odot$) at $z = 2$.

run at the higher resolution. This in contrast with methods such as Ramanah et al. (2020) or Rodríguez et al. (2018), which generate a density field.

One can ask whether there is fundamental limit to the level of SR enhancement that our approach is able to produce. In principle, one should be able to generate simulations completely using a GAN, without a low-resolution model for conditioning (e.g, Rodríguez et al. 2018, or List & Lewis 2020). Logically, such an approach would need proportionally more training data to produce satisfactory results, as it would be generating large scale modes as well. At the opposite end, Ramanah et al. (2020) have added super-resolution details to their simulations by enhancing by a factor of 2 in length scale. In this paper we have shown that our SR modeling works well to produce small-scale structures a factor of 8 in length (and 512 in mass) below the LR scale. At these scales one can begin to find bound structures such as subhalos.

The main advantage in the use of SR simulations over full HR runs is the potentially huge reduction in the computational resources involved. As an example, for the $(100\,h^{-1}\mathrm{Mpc})^3$ volume simulations of our test set, it takes approximately 560 core hours to run a 512^3 particle HR model to redshift $z = 2$, while the 64^3 particle LR run takes only about 0.6 core hours, a factor of $\sim$ 1000 faster. We can achieve even more dramatic speed up when applying the SR model to a larger volume, as we show in Sec. 4. For a $(1\ \ h^{-1}\mathrm{Gpc})^3$ cosmological volume, we have found that only 500 core hours are needed to run a 640^3 particle LR simulation to $z = 2$. To run a 5120^3 HR counterpart with an N-body code would be daunting, and require dedicated supercomputing resources. Our trained GAN model, on the other hand, only takes 16 hours with 1 GPU (including the I/O time) to generate a 5120^3 SR field for the $(1\ \ h^{-1}\mathrm{Gpc})^3$ volume, a tiny fraction of the cost of the HR counterpart. An additional advantage, made apparent by our processing the simulations in distinct chunks is that the SR enhancement can be applied where it is needed, e.g. in the vicinity of a specific galaxy cluster or supermassive black hole. The data storage required can therefore also be much less than for a full HR run.

The stochastic nature of the StyleGAN implementation is also an interesting feature which can be exploited. As we have seen in Fig. 4, it is possible to sample multiple "realizations" of the small-scale clustering, conditioning on the large scale modes, by varying the input noise component. This opens up the possibility to improve the statistical inference of cosmology from the small-scale clustering of galaxies, by jointly sampling the small-scale modes with the cosmological parameters.

To train and test our model, we have used simulations of the same cosmology, and this simplifies the learning task. It is expected that the LR to HR mapping should depend weakly on cosmology, and introducing such dependence should make the model adaptable and more accurate for a range of parameters spanned by the training set. Furthermore, to fully capture the state of a simulation, our model can be used to learn particle velocities as well. We leave those cosmology dependence and velocity improvements for future work.

A central question for GANs (and DL in general) is how to determine if enough training data is being used. Different diagnostics can be employed, and the dependence of accuracy on training set size can be evaluated. We have seen that our SR simulations are at least able to reproduce the power spectrum and halo mass function well with a very limited set of training data. A related issue is the fact that in astrophysics we often deal with rare objects (e.g., quasars or galaxy clusters), and there may not even be a single example in the training data. In the future, one should examine how best to assemble a training dataset that sufficiently covers all cases that will be studied. As an example, separate universe or constrained simulations (e.g., Li et al. 2014, Barreira et al. 2019, Huang et al. 2020) could be used to enhance the number of high mass objects in the training data.

So far though, we have demonstrated that our method is capable of generating SR simulations one thousand times larger than the training sets. Our large SR simulation does successfully reproduce the halo mass function and power spectrum over several orders of magnitude. It also leads to the formation of visually reasonable large and rare galaxy clusters. Future work will further evaluate the quality of generated rare objects using other measures.

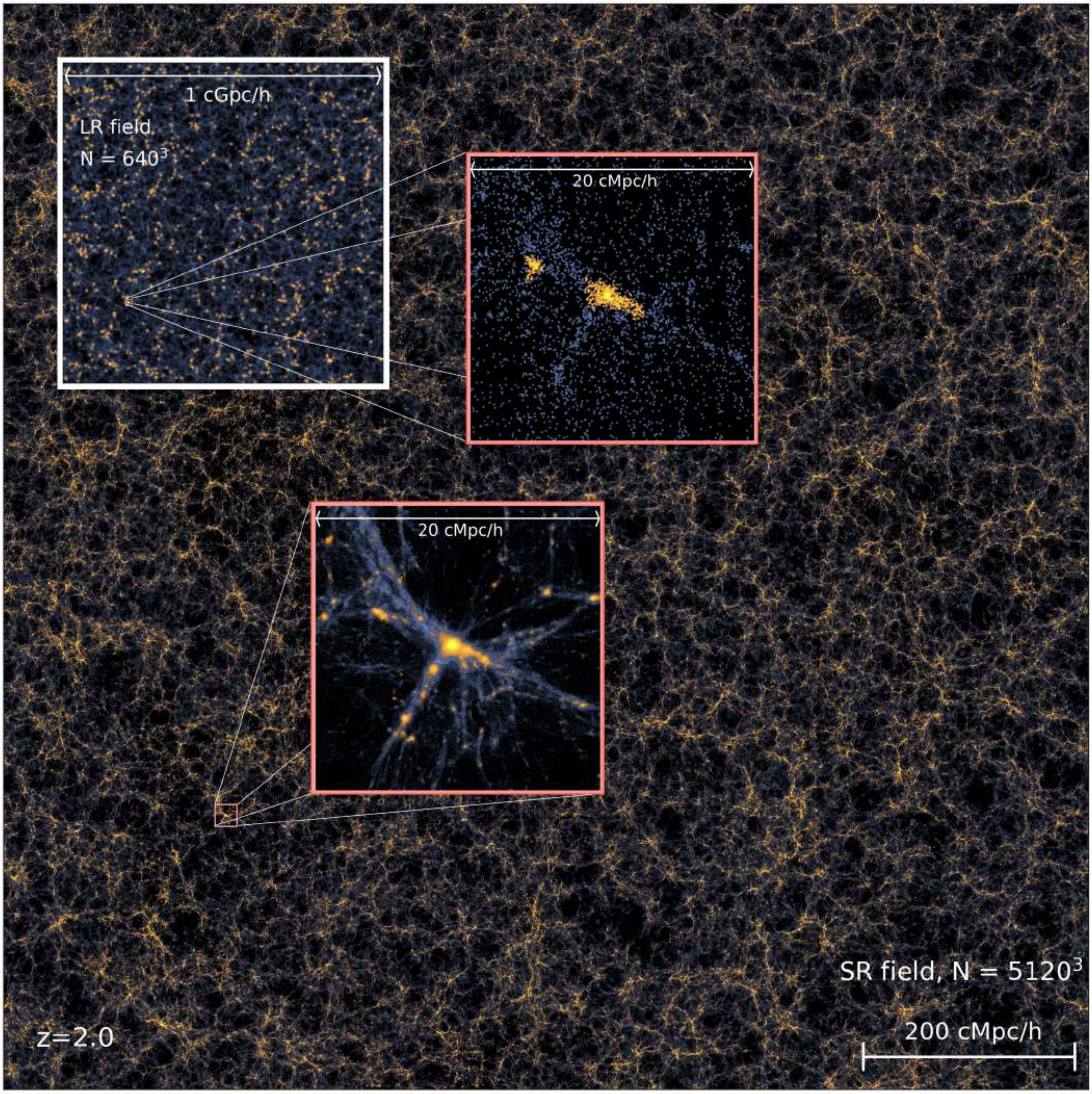


Figure 7. Illustration of the $(1\ h^{-1}\mathrm{Gpc})^3$ volume SR with 5120^3 particles generated from the input 640^3 LR field using our GAN model. Colors are the same as in Fig. 3. The large panel shows a slice through the full box, 1 h^{-1}Gpc in length and 20 h^{-1}Mpc in thickness. The left top inset in the white box shows the same field from the LR simulation. In the two red inset panels, we show a $(20\,h^{-1}\mathrm{Mpc})^3$ zoom-in region around a massive halo in both the SR and LR field. That central halo has a mass of $2 \times 10^{14}\ M_\odot$, larger than the most massive halo in the $(100\,h^{-1}\mathrm{Mpc})^3$ volume training set. The generating process only takes about 16 hours on 1 GPU.

Particularly in the way we have generated them (as full particle position datasets), the SR simulations could be used for many purposes that would have needed HR simulations. These include mock galaxy catalogues for large-scale structure studies (e.g., Lin et al. 2020). Our current SR models are dark matter only, but could be used for mocks in the same manner as HR models with resolved halos. As we discuss below, extensions of this work will mean that it will be possible to include hydrodynamic and star formation effects, through training using HR models which have these physics incorporated. SR mock catalogues could then be made which are more complex, for example including reionization (Satpathy et al. 2020). This type of mock making would have some similarities with the "painting" of galaxies onto dark matter simulations by Agarwal et al. (2018), except that unlike that paper our method is conditioned on the entire density distribution rather than a few halo properties, and it would also add SR structure to the model.

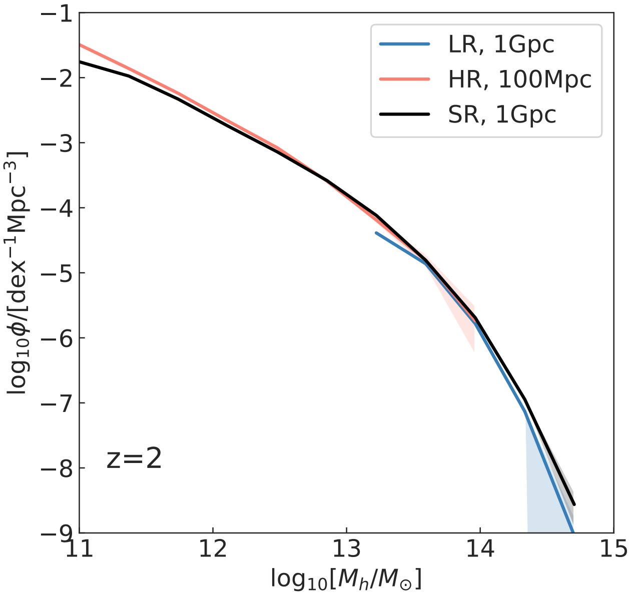


Figure 8. Halo mass function comparison of the $(1\ h^{-1}\mathrm{Gpc})^3$ LR (blue), the $(1\ h^{-1}\mathrm{Gpc})^3$ SR (black and generated from the former), and a $(100\,h^{-1}\mathrm{Mpc})^3$ HR (orange) simulations at $z = 2$. Limited by mass resolution, the LR run only resolves the most massive halos with $M_\mathrm{h} \gtrsim 10^{13.5}\ M_\odot$. Also, limited by volume, the HR run lacks halos above $M_\mathrm{h} \gtrsim 10^{14}\ M_\odot$. Our SR realization is able to match both the resolved LR and HR abundances over the whole mass range, except for a slight deficit at $10^{11}\ M_\odot$. Again we emphasize that our model successfully predicts the abundance for massive halos that it has not been trained on, suggesting it has learned some structure formation physics.

Straightforward extensions of this work can be imagined, with different levels of sophistication. Particle data can be used to generate super-resolution three dimensional gas, star and dark matter particle distributions. Because the prediction will be of SR particles, these would effectively function as full simulations, as we have mentioned previously. In order to achieve this, training will have to be carried out on full physics models (such as e.g., Feng et al. 2015; Marshall et al. 2019). It should also be possible to train on HR models run with different hydrodynamics algorithms (e.g. ENZO by Brummel-Smith et al. 2019) than are used in the LR models (as long as the initial conditions are identical). In such SR calculations, we will still be adding SR details to simulations after the LR models have been run, as in this paper.

Another extension is to replace the LR scheme, currently a full N-body solver, with faster alternatives, such as another DL model (e.g. D3M, He et al. 2019), or a pseudo N-body solver (e.g. FastPM, Feng et al. 2016b, or FlowPM, Modi et al. 2020). The computational cost of a SR realization can be further reduced. We will also be able to compute the full sensitivity matrix of the LR + SR model from back-propagation, which in turn allows us to use the method in Hamilton Monte-Carlo or similar sampling methods (e.g., Seljak et al. 2017; Leclercq et al. 2019) for a joint study of the initial condition and the cosmology parameters.

Beyond this, an improvement will be to run AI algorithms “on the fly”, alongside the hydrodynamic simulation codes. This will allow coupling and feedback between the physical processes at super resolved scales and the LR scales. For the most accurate modeling, this will be important, as for example star formation driven winds or metal pollution from supernovae can spread far from a galaxy. This means that the SR structure on galaxy scales should ideally be able to affect the properties of the LR simulation. To achieve this, it may also become necessary to carry out training on the fly as HR and LR simulations run together. We leave development of these methods to future work.

The ongoing revolution in Artificial Intelligence has already changed many fields of science. Applications of these techniques are becoming widespread in cosmology also. Super-resolution enhancement, that we have explored here, has many attractive features: Conventionally generated large-scale Fourier modes are fully linked to rapidly generated small-scale structures. This will allow consistent full volume simulations of the Universe to be produced, covering an unprecedented dynamic range.

ACKNOWLEDGEMENTS

We acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this paper. TDM acknowledges funding from NSF ACI-1614853, NSF AST-1517593, NSF AST-1616168 and NASA ATP 19-ATP19-0084. TDM and RACC also acknowledge funding from NASA ATP 80NSSC18K101, and NASA ATP NNX17AK56G. SB was supported by NSF grant AST-1817256. This work was also supported by the NSF AI Institute: Physics of the Future, NSF PHY- 2020295.

REFERENCES

Agarwal S., Davé R., Bassett B. A., 2018, MNRAS, 478, 3410
Arjovsky M., Chintala S., Bottou L., 2017, in Proceedings of the 34th International Conference on Machine Learning - Volume 70. ICML’17. JMLR.org, Sydney, NSW, Australia, pp 214–223, http://arxiv.org/abs/1701.07875
Barreira A., Nelson D., Pillepich A., Springel V., Schmidt F., Pakmor R., Hernquist L., Vogelsberger M., 2019, MNRAS, 488, 2079
Berger P., Stein G., 2019, Monthly Notices of the Royal Astronomical Society, 482, 2861
Brummel-Smith C., et al., 2019, The Journal of Open Source Software, 4, 1636
Dai B., Seljak U., 2020, arXiv e-prints, p. arXiv:2010.02926
Dai B., Feng Y., Seljak U., 2018, J. Cosmology Astropart. Phys., 2018, 009
Dai B., Feng Y., Seljak U., Singh S., 2020, J. Cosmology Astropart. Phys., 2020, 002
Davis M., Efstathiou G., Frenk C. S., White S. D. M., 1985, ApJ, 292, 371
Dong C., Loy C. C., He K., Tang X., 2015, IEEE transactions on pattern analysis and machine intelligence, 38, 295
Feng Y., Di Matteo T., Croft R., Tenneti A., Bird S., Battaglia N., Wilkins S., 2015, ApJ, 808, L17
Feng Y., Di-Matteo T., Croft R. A., Bird S., Battaglia N., Wilkins S., 2016a, MNRAS, 455, 2778
Feng Y., Chu M.-Y., Seljak U., McDonald P., 2016b, MNRAS, 463, 2273
Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y., 2014, in Advances in neural information processing systems. pp 2672–2680
Goodfellow I., Bengio Y., Courville A., 2016, Deep learning. MIT press
Gulrajani I., Ahmed F., Arjovsky M., Dumoulin V., Courville A. C., 2017, in Guyon I., Luxburg U. V., Bengio S., Wallach H., Fergus R., Vishwanathan S., Garnett R., eds, Advances in Neural Information Processing Systems 30. Curran Associates, Inc., pp 5767–5777, http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf
He K., Zhang X., Ren S., Sun J., 2016, in Proceedings of the IEEE conference on computer vision and pattern recognition. pp 770–778
He S., Li Y., Feng Y., Ho S., Ravanbakhsh S., Chen W., Póczos B., 2019, Proceedings of the National Academy of Sciences, p. 201821458
Huang K.-W., Ni Y., Feng Y., Di Matteo T., 2020, MNRAS, 496, 1

Isola P., Zhu J.-Y., Zhou T., Efros A. A., 2017, in Proceedings of the IEEE conference on computer vision and pattern recognition. pp 1125–1134

Jing Y. P., 2005, ApJ, 620, 559

Johnson J., Alahi A., Fei-Fei L., 2016, in European conference on computer vision. pp 694–711

Kamdar H. M., Turk M. J., Brunner R. J., 2016, Monthly Notices of the Royal Astronomical Society, 457, 1162

Karras T., Laine S., Aittala M., Hellsten J., Lehtinen J., Aila T., 2019a, arXiv preprint arXiv:1912.04958

Karras T., Laine S., Aila T., 2019b, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp 4401–4410

Leclercq F., Enzi W., Jasche J., Heavens A., 2019, MNRAS, 490, 4237

Ledig C., et al., 2017, in Proceedings of the IEEE conference on computer vision and pattern recognition. pp 4681–4690

Li Y., Hu W., Takada M., 2014, Physical Review D, 89, 083519

Lin S., et al., 2020, MNRAS,

List F., Lewis G. F., 2020, MNRAS, 493, 5913

Marshall M. A., Ni Y., Di Matteo T., Wyithe J. S. B., Wilkins S., Croft R. A. C., Kuusisto J. K., 2019, arXiv e-prints, p. arXiv:1912.03428

Mirza M., Osindero S., 2014, arXiv preprint arXiv:1411.1784

Modi C., Feng Y., Seljak U., 2018, J. Cosmology Astropart. Phys., 2018, 028

Modi C., Lanusse F., Mustafa M., Seljak U., 2020, Simulating the Universe in TensorFlow, `https://blog.tensorflow.org/2020/03/simulating-universe-in-tensorflow.html`

Ramanah D. K., Charnock T., Lavaux G., 2019, Physical Review D, 100, 043515

Ramanah D. K., Charnock T., Villaescusa-Navarro F., Wandelt B. D., 2020, MNRAS, 495, 4227

Rodríguez A. C., Kacprzak T., Lucchi A., Amara A., Sgier R., Fluri J., Hofmann T., Réfrégier A., 2018, Computational Astrophysics and Cosmology, 5, 4

Russell S. J., Norvig P., 2020, Artificial Intelligence: a modern approach, 4 edn. Pearson

Satpathy S., An Z., Croft R. A. C., Di Matteo T., Tenneti A., Feng Y., Heitmann K., Rossi G., 2020, MNRAS,

Schawinski K., Zhang C., Zhang H., Fowler L., Santhanam G. K., 2017, MNRAS, 467, L110

Seljak U., Aslanyan G., Feng Y., Modi C., 2017, J. Cosmology Astropart. Phys., 2017, 009

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

Wadekar D., Villaescusa-Navarro F., Ho S., Perreault-Levasseur L., 2020, arXiv preprint arXiv:2007.10340

Wang Z., Chen J., Hoi S. C., 2020, IEEE Transactions on Pattern Analysis and Machine Intelligence

Zhang X., Wang Y., Zhang W., Sun Y., He S., Contardo G., Villaescusa-Navarro F., Ho S., 2019, arXiv preprint arXiv:1902.05965

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