Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **545,** 1–15 (2026) https://doi.org/10.1093/mnras/staf2071
Advance Access publication 2025 November 21

Large, fast, and accurate H I intensity maps with latent overlap diffusion

Satvik Mishra,[1]★ Roberto Trotta [1,2,3,4] and Matteo Viel[1,2,3,5,6,7]

[1]*Theoretical and Scientific Data Science, SISSA, Via Bonomea 265, I-34136 Trieste, Italy*
[2]*INFN – National Institute for Nuclear Physics, Via Valerio 2, I-34127 Trieste, Italy*
[3]*ICSC – Centro Nazionale di Ricerca in High Performance Computing, Big Data e Quantum Computing, Via Magnanelli 2, I-40033 Bologna, Italy*
[4]*Astrophysics Group, Physics Department, Blackett Lab, Imperial College London, Prince Consort Road, London SW7 2AZ, UK*
[5]*Astroparticle and Gravitational Physics Group, SISSA, Via Bonomea 265, I-34136 Trieste, Italy*
[6]*INAF – Osservatorio Astronomico di Trieste, Via G. B. Tiepolo 11, I-34143 Trieste, Italy*
[7]*IFPU – Institute for Fundamental Physics of the Universe, Via Beirut 2, I-34151 Trieste, Italy*

Accepted 2025 November 19. Received 2025 October 10; in original form 2025 June 12

ABSTRACT

The distribution of 21 cm emission from neutral hydrogen is a powerful cosmological and astrophysical probe, as it traces the underlying dark matter and cold gas distributions throughout cosmic times. However, the prediction of observable signals is hindered by the large computational costs of the required hydrodynamic simulations. We introduce a novel machine learning pipeline that, once trained on a hydrodynamical simulation, is able to generate both halo mass density maps and the three-dimensional 21 cm brightness temperature signal, starting from a dark matter-only simulation. We use an attention-based ResUNet (`HALOgen`) to predict dark matter halo maps, which are then processed through a trained conditional variational diffusion model (`LODI`) to produce 21 cm brightness temperature maps. `LODI` is trained on smaller sub-volumes that are then seamlessly combined in 512-times larger volume using a new method, called 'latent overlap'. We demonstrate that, once trained on 25^3 $(\mathrm{Mpc}/h)^3$ volume simulations, we are able to predict the 21 cm power spectrum on an unseen dark matter map (with the same cosmology) to within 10 per cent for wavenumbers $k \leq 10\ h\ \mathrm{Mpc}^{-1}$, deep inside the non-linear regime, with a computational effort of the order of two minutes. While demonstrated on this specific volume, our approach is designed to be scalable to arbitrarily large simulations.

Key words: software: machine learning – dark matter – large-scale structure of Universe – galaxies: haloes.

1 INTRODUCTION

Neutral (atomic) hydrogen (H I) plays an important role in cosmology and structure formation processes: its distribution follows the underlying matter density field, and for most of the cosmic history it constitutes the reservoir of baryons to fuel star formation. This makes it a powerful and novel tracer of the large-scale structure (LSS) of the Universe (e.g. R. Ansari et al. 2012; J. R. Pritchard & A. Loeb 2012; F. Villaescusa-Navarro et al. 2014; M. G. Santos et al. 2015). While Cosmic Microwave Background observations (G. Hinshaw et al. 2013; Planck Collaboration VI 2020; M. Mallaby-Kay et al. 2021) and galaxy redshift surveys (e.g. S. Alam et al. 2017) have significantly constrained the parameters of the standard Lambda cold dark matter cosmological model, key questions are still unresolved. In particular, the fundamental nature of dark matter and dark energy is still unknown, and persistent tensions between different cosmological measurements have yet to be fully understood, including the recent tentative evidence for evolving dark energy (e.g. A. G. Riess et al. 2019; L. Verde, T. Treu & A. G. Riess 2019; K. C. Wong et al. 2020; DESI Collaboration 2025). Mapping the large-scale distribution of H I and tracking its evolution over cosmic time offer a complementary approach to traditional galaxy surveys, by probing large volumes at high redshifts, and thus potentially providing stress tests for many cosmological models in new regimes (e.g. F. Villaescusa-Navarro, P. Bull & M. Viel 2015; P. Bull et al. 2016; F. Villaescusa-Navarro, D. Alonso & M. Viel 2017; A. Obuljen et al. 2018; M. Berti et al. 2022).

The 21 cm line, which results from the spin-flip transition within the hyperfine structure of the ground state of neutral hydrogen (see S. Furlanetto, S. P. Oh & F. Briggs 2006), is redshifted by the cosmological expansion and can be observed at radio wavelengths. Many experiments, including interferometric arrays like CHIME (K. Bandura et al. 2014; CHIME Collaboration 2023), CHORD, and HIRAX (L. B. Newburgh et al. 2016), and single-dish instruments such as GBT (K. W. Masui et al. 2013; L. Wolz et al. 2022) and FAST (W. Hu et al. 2020), aim to detect this signal using intensity mapping (IM) techniques (S. Bharadwaj et al. 2001; R. A. Battye, R. D. Davies & J. Weller 2004; M. McQuinn et al. 2006; T.-C. Chang et al. 2008; H.-J. Seo et al. 2010; R. A. Battye et al. 2013; E. D. Kovetz et al. 2017). Several of these efforts have already achieved detections through cross-correlation with optical galaxy surveys (T.-C. Chang et al. 2010; K. W. Masui et al. 2013; C. J. Anderson et al. 2018; S. Cunnington 2022; L. Wolz et al. 2022; S.

★ E-mail: samishr@sissa.it

Paul et al. 2023). Radio cosmology is also a key science objective for the SKA Observatory (SKAO),[1] which will comprise two major arrays: SKA-Low in Australia and SKA-Mid in South Africa. In particular, SKA-Mid, when operated in single-dish mode (e.g. M. G. Santos et al. 2015; SKA Cosmology SWG 2020), will be capable of conducting 21 cm IM surveys across cosmologically relevant scales out to redshift $z \sim 3$. Currently under construction, the SKAO has a working precursor, MeerKAT, which is already contributing through its cosmological IM survey, MeerKLASS (M. G. Santos et al. 2018). Early results from MeerKAT data have been encouraging (J. Wang et al. 2021; M. O. Irfan et al. 2022), including a first cross-correlation detection with WiggleZ galaxy data (S. Cunnington et al. 2023).

Alongside several technical efforts mainly focused on foreground modelling and mitigation, refining the forecast capabilities of 21 cm IM, both as a standalone probe and in combination with other cosmological observables, is crucial (e.g I. P. Carucci, F. Villaescusa-Navarro & M. Viel 2017; M. Berti, M. Spinelli & M. Viel 2024). These forecasts are essential not only for strengthening the scientific case for 21 cm IM radio cosmology, but also in order to optimize the design and strategy of upcoming surveys. The theoretical modelling of the 21 cm signal follows three different approaches: halo models (see H. Padmanabhan et al. 2023), perturbation theory (e.g. A. Obuljen et al. 2023) or hydrodynamical simulations of structure formation incorporating the relevant physical processes (e.g. F. Villaescusa-Navarro et al. 2018). Each of these methods has both advantages and disadvantages. For example, modelling the IM signal requires both large volumes and high resolution to fully resolve the physics of the small mass ($\sim 10^{10}\,\mathrm{M}_\odot/h$) dark matter haloes that host H I, something that is difficult to achieve with full hydrodynamical simulations due to the range of scales. In the context of halo models, extensive work has been expended to study the mass limit for haloes to retain H I, for example by looking at the damped Lyman-α systems statistics, both from the point of view of halo models (H. Padmanabhan & A. Refregier 2016) and observationally (A. Obuljen et al. 2019; A. Dev et al. 2023). In addition to these semi-analytical and perturbative frameworks, several forward-modelling approaches have been developed to predict the H I field directly from dark-matter simulations by prescribing or calibrating the $M_{\rm HI}(M)$ relation and other parameters on observations or hydrodynamical runs (e.g. M. Spinelli et al. 2020; Z. Li, H. Guo & Y. Mao 2022; P. Hitz et al. 2025). These methods however extend predictions to large volumes by applying calibrated prescriptions for the H I–halo connection, rather than learning the full field-level mapping from simulations. Thus, a method capable of learning the small scale physics from state-of-the-art simulations while at the same time reaching large scales and volumes would be of great importance.

In the context of machine learning (ML) application, progress in neural-based modelling has been recently achieved at large scales and for large volumes (e.g. T. Nguyen et al. 2024; S. Pandey et al. 2023, 2024). In the context of 21 cm IM theoretical modelling, there have been several recent efforts aim at predicting the brightness temperature signature at the field level, including fast generative models of H I maps using normalizing flows (S. Hassan et al. 2022), the neural approach of D. Wadekar et al. (2021), and a generative adversarial framework introduced by S. Andrianomena, F. Villaescusa-Navarro & S. Hassan (2024). Diffusion models (e.g. D. P. Kingma et al. 2021) have thus far found very limited applications (V. Ono et al. 2024), and none in the realm of 21 cm IM. In this work, we develop a novel, physically motivated ML pipeline, which in a first step uses a U-Net to learn the mapping between dark matter and haloes; in a second step, it employs a custom-designed diffusion model to predict the 21 cm signal, down to very small, non-linear scale. Predicting both haloes and 21 cm signal is crucial in the context of using such machinery for cross-correlation studies of the IM signal with other LSS tracers (like lensing, galaxies, etc.), which is the ultimate of aim of our work. This work should therefore be viewed as the first demonstration of a diffusion-based, halo-conditioned approach to 21 cm IM, complementary to existing forward modelling methods.

The paper is organized as follows: in Section 2, we describe in the the methodology used, including the ML pipeline, U-Net architecture, diffusion model with the novel latent overlap method, simulations used, loss function, networks training and validation of the pipeline. In Section 3, we present our results in terms of illustrative halo and H I maps, and quantitative comparison of power spectra predicted for held-out dark matter maps. We conclude in Section 4, where we also indicate promising avenues for future work and applications of our method.

2 METHODOLOGY

In this section, we explain the ML technique, the mock data sets used, and the training methods implemented.

Our strategy is the following: starting from a dark matter only map, we use a U-Net architecture to predict the halo map, as described in Section 2.1.

We then use the halo map as input to a variational diffusion model (VDM) trained to inpaint the H I brightness temperature, as described in Section 2.2.

An overview of the full pipeline is presented in Fig. 1.

2.1 `HALOgen`: from dark matter to halo maps

2.1.1 U-Net architecture

A U-Net, introduced by O. Ronneberger, P. Fischer & T. Brox (2015) is a deep hierarchical convolutional neural network with encoder and decoder layers, used to perform image-to-image translation tasks – an ideal architecture for our aim of predicting halo maps from $\boldsymbol{\rho}_{\rm DM}$ maps.

We use ResNet blocks (K. He et al. 2016) as the fundamental blocks for the U-Net architecture, as well as an attention block at the bottleneck motivated by O. Petit et al. (2021). The attention block enhances the ability of the model to capture and highlight complex structure and long-range dependencies within the data. The architecture, depicted in Fig. 2, consists of three encoding blocks with grouped convolutions, which process a 3D $\boldsymbol{\rho}_{\rm DM}$ input of size 64^3 voxels. The input is progressively downsampled through three stages, reaching a bottleneck representation of size of 8^3 voxels, with $2 \times 2 \times 2$ kernel max-pooling layers employed for downsampling to reduce spatial resolution while preserving dominant features. For the up-sampling leg of the architecture, we use trilinear interpolation instead of the more commonly used transposed convolutions to avoid checkered board artefacts – grid-like patterns that can arise when upsampling images. Trilinear interpolation estimates voxel values by computing a weighted average of the eight nearest neighbours, ensuring a smoother reconstruction. To enhance high-level feature preservation, we incorporate skip connections (represented as horizontal grey lines in Fig. 2) after upsampling, whose aim is to transfer directly feature maps from corresponding encoder layers to decoder layers, thus bypassing the bottleneck.

[1] https://www.skao.int/

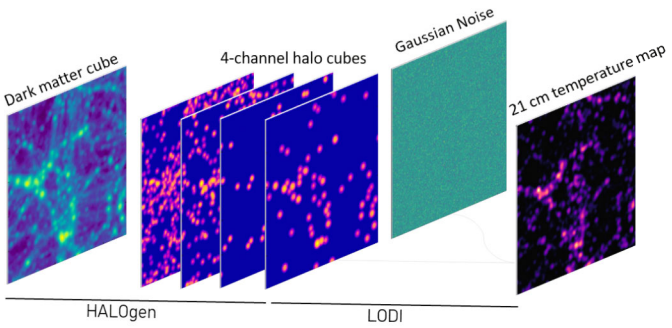

Figure 1. An overview of our generative pipeline, starting from dark matter particle distribution as produced by an *N*-body code to the final 21 cm intensity map: in the first step, a ResNet with attention bottleneck (HALOgen) is used to predict the dark matter halo mass density in four mass channels; subsequently, a VDM with latent overlap (LODI) generates the 21 cm brightness temperature map.

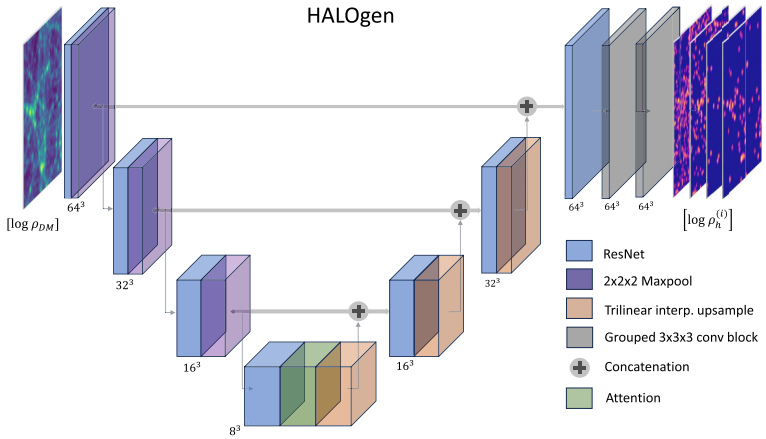

Figure 2. Overview of the HALOgen (Halo Assignment and Generation with U-Net) architecture used for halo assignment from an input DM-only map, employing grouped 3D convolutions (grey blocks) and an attention mechanism at the bottleneck. Upsampling is done via trilinear interpolation to avoid artefacts in the upsampled map. Horizontal lines represent skip connections.The input data has shape of 64^3 voxels while the bottleneck has a spatial shape of 8^3.

The last decoder block leads to a set of grouped convolution layers that transform the output into four channels, each corresponding to a halo map density field $\boldsymbol{\rho}_{\rm h}^{(i)}$, where $i = 1, \ldots, 4$ is the channel number. Each channel describes haloes within a top-hat mass range (a bin). Attributing haloes to different mass bins is necessary to reduce the dynamic range of the relationship between halo mass and H I mass, and to account for its strong dependence on halo mass: haloes with mass $M_{\rm halo} < 3 \times 10^{10}\,{\rm M}_\odot$ contribute ≈95 per cent of the total number count of haloes, but their combined H I mass is only ≈5 per cent of the total H I in the simulation. At the other end of the mass distribution, haloes of mass $\geq 10^{12}\,{\rm M}_\odot$ populate only about ≈0.5 per cent of the simulation but account for more than 25 per cent of the H I mass. Our binning both reduces the dynamic range spanned by the $M_{\rm HI}$ to $M_{\rm halo}$ relationship (thus facilitating learning) and also accounts for the halo-mass dependency of the relation. The H I mass function, as determined by M. G. Jones et al. (2018) and W. Ma et al. (2025), features a knee H I mass of $\approx 7.5 \times 10^9\,{\rm M}_\odot$, with the power-law slope for low mass being $\alpha = -1.3$, indicating that contribution from H I regions smaller than the knee H I mass is non-negligible. Following F. Villaescusa-Navarro et al. (2018), the H I cut-off mass for our case is $\approx 10^8\,{\rm M}_\odot$, which is almost two orders of magnitude smaller than the knee mass determined observationally. This provides reassurance that we are accounting for most of the H I regions.

The (fixed) boundaries of the top-hat mass ranges are chosen to obtain similar H I mass within each bin,[2] and are given by:

$$\left[3 \times 10^{10}, 10^{11}, 5 \times 10^{11}, 10^{12}, \max\left(M_{\rm halo}\right)\right]\ {\rm M}_\odot, \tag{1}$$

where the maximum value of $M_{\rm halo}$ is $\approx 1.7 \times 10^{14}\,{\rm M}_\odot$. We use group convolution and group normalization layers to enforce that output channels be independent; experimentation with mixing the information between different groups (4 or multiples of 4) has shown that this degrades performance. The output of the network has a voxel dimension of 4×64^3. Compared to larger simulations such as TNG300, which host haloes up to $\approx 10^{15}\,{\rm M}_\odot$, the smaller CAMELS volumes do not sample such large systems. While this restricts the direct modelling of the rarest, most massive clusters, CAMELS provides a more machine-learning–friendly baseline, and our approach can in the future be extended to larger simulations or large cluster regions.

To create the full 256^3 voxels map, we take individual 64^3 voxels sub-volume dark matter fields and feed them to our model. Then we

[2]This remains true, up to a few per cent, for all maps in the cross-validation set of the simulation, which share the same astrophysical parameters. Generalizing this approach across different parameter values will be the focus of future work.

use a stride of 16 pixels in every cartesian direction to pick our next 64^3 box. The overlapping pixels are averaged over to get the final output.

2.1.2 Training dark matter and halo mass density maps

To obtain the input DM mass density map, we take the positions for each DM particle from the DM-only simulations with side $25h^{-1}$ Mpc and convert them to three-dimensional $\boldsymbol{\rho}_{\rm DM}$ fields of size 256^3 voxels, by using the cloud in cell (CIC) algorithm. The CIC algorithm assigns particle masses to a regular grid by distributing each particle's mass to the eight nearest grid points, weighted linearly by the particle's relative distance from each grid point. This produces a continuous density field while conserving total mass. Our field resolution is thus $\approx 0.1h^{-1}$ Mpc. We use for our model training the CAMELS data set from F. Villaescusa-Navarro et al. (2022), a suite of hydrodynamical and N-body cosmological simulations. More detail related to the simulations we use is provided in Section 2.3.

To determine the halo mass density $\boldsymbol{\rho}_{\rm h}$, we use the SUBFIND algorithm (V. Springel et al. 2001), available in the simulation suite to extract the halo centre-of-mass positions and then use the same CIC algorithm, weighted by the mass of the haloes. We manually create the four halo channels using the bins given in equation (1), and apply the CIC algorithm on each bin separately.

Maps are smoothed with a Gaussian kernel of fixed radius $R = 0.2$ Mpc h^{-1} to increase the signal to noise ratio (SNR) and remove irrelevant small-scale features to help training.

2.1.3 `HALOgen` loss function

The loss function needs to deal with the sparsity of the $\boldsymbol{\rho}_{\rm h}^{(i)}$ maps. Furthermore, a given DM voxel may be associated with multiple haloes of different masses (and falling into different halo mass bins), making it challenging for the model to disentangle their respective contributions. We call this phenomenon 'channel mixing', i.e. the fact that halo information from one channel leaks into adjacent channels. Moreover, since the binning scheme is not explicitly encoded within the neural network architecture, the model must learn to distinguish haloes across adjacent bins without misclassifications. Ensuring proper bin differentiation is crucial to maintaining the physical consistency of the model predictions.

To address the above issues, we create a custom-weighted mask, m_w^i $(i = 1, \ldots, 4)$, associated to each halo channel i. The purpose of the mask is to give higher weight to pixels with haloes in channel i, while enforcing a negligible contribution to the loss from haloes in neighbouring channels, thus helping both with sparsity and channel separation.

For a given halo channel i, with halo mass density field $\boldsymbol{\rho}_{\rm h}^{(i)}$, we define its weight as:

$$w^i = \frac{N_{\rm mesh}^3}{\sum_{j=\max(1,i-1)}^{\min(4,i+1)} \sum_{k=1}^{N_{\rm mesh}} \boldsymbol{\theta}(\rho_{\rm h}^{(j,k)} - T_{\rm c}^{(j)})} \geq 1\,, \tag{2}$$

where $\rho_{\rm h}^{(j,k)}$ is the halo density for channel j at pixel k and $\boldsymbol{\theta}(x)$ is the Heaviside function. The threshold $T_{\rm c}^{(j)}$ is determined from the empirical CDF of $\rho_{\rm h}^{(j,k)}$, in such a way that

$$P(\rho_{\rm h}^{(j,k)} > T_{\rm c}^{(j)}) = 0.05\,. \tag{3}$$

In other words, the threshold eliminates pixels that contain low-density values, and only retains the highest 5 per cent of the log-density distribution in each channel. Note that for a given channel j, the pixels used for the computation are taken not only from the channel j, but also from the immediate neighbouring channels, $j+1$ and $j-1$. For the boundary channels (i.e. $j = 1$ and $j = 4$), we include only the next or previous channel, respectively. Now we define a pixel- and channel-specific mask $m_w^{(i,k)}$ by incorporating these weights:

$$m_w^{i,k} = \begin{cases} w^i & \text{if } \rho_{\rm h}^{(i,k)} > T_{\rm c}^{(i)} \\ 1 & \text{if } \rho_{\rm h}^{(i,k)} \leq T_{\rm c}^{(i)}\,. \end{cases} \tag{4}$$

We normalize the mask such that $\sum_{i,k} m_w^{i,k} = 64$, or each training mask of shape 4×64^3 adds to 1 on average, since each simulation is made of 64 sub-volumes.

An illustration of the construction of $m_w^{(i,k)}$ is shown in Fig. 3. By construction, pixels corresponding to halo sites for channel j (beige) have high mask values; the mask also has large values for adjacent channel's halo positions (golden), in order to enforce the absence of haloes there in the particular channel under consideration, since for these regions, the target voxel value corresponds to the small background values.

The masked loss function is then written as:

$$\mathcal{L} = \sum_{j=1}^{4} \left(\alpha L_2^j \odot m_w^j \beta^j + (1-\alpha)\, L_q^j \right) + \lambda_w L_1^{\rm reg} \tag{5}$$

with α, $\beta^{j=1,\ldots,4}$, λ_w being hyper-parameters of the model. The loss incorporates both the mean squared error L_2^j and the quantile loss L_q^j between the pixel-level target density, $\rho_{\rm h}^{(j,k)}$, and its predicted value, $\rho_{\rm pred}^{(j,k)}$, weighed by the mask for the L_2 loss, with the direct product running over pixel values k:

$$\begin{aligned} L_q^j &= \frac{1}{N_{\rm mesh}} \sum_{k=1}^{N_{\rm mesh}} \max\Big(q \left(\rho_{\rm h}^{(j,k)} - \rho_{\rm pred}^{(j,k)} \right), \\ &\quad (q-1) \left(\rho_{\rm h}^{(j,k)} - \rho_{\rm pred}^{(j,k)} \right) \Big), \\ L_2^j &= \frac{1}{N_{\rm mesh}} \sum_{k=1}^{N_{\rm mesh}} \left(\rho_{\rm pred}^{(j,k)} - \rho_{\rm h}^{(j,k)} \right)^2 . \end{aligned} \tag{6}$$

Here, $q \in (0, 1)$ is the chosen quantile. The weights β^j set the relative scaling of the contribution of a particular channel to the loss. To further deal with sparsity, we add the $L_1^{\rm reg}$ regularization term, defined as:

$$L_1^{\rm reg} = \sum_{i=1} \left| w_{\boldsymbol{\theta}}^i \right|, \tag{7}$$

where $\left| w_{\boldsymbol{\theta}}^i \right|$ are the neural network weights, and i runs over the network's parameters.

2.2 `LODI`: H I intensity maps with latent overlap diffusion

2.2.1 Diffusion model architecture

Diffusion models are a class of generative models, proposed by J. Sohl-Dickstein et al. (2015) and J. Ho, A. Jain & P. Abbeel (2020) that start from a Gaussian random field and denoise it in steps to produce a data point sampled from the underlying data distribution. A diffusion model has two parts: in the forward noising process, Gaussian noise is progressively added to a data point to convert it into white noise. This forward process can be written as,

$$q(\boldsymbol{x}_{\rm t} \mid \boldsymbol{x}_0) = \mathcal{N}\left(\alpha_{\rm t} \boldsymbol{x}_0,\, \sigma_{\rm t}^2 \mathbf{I}\right), \tag{8}$$

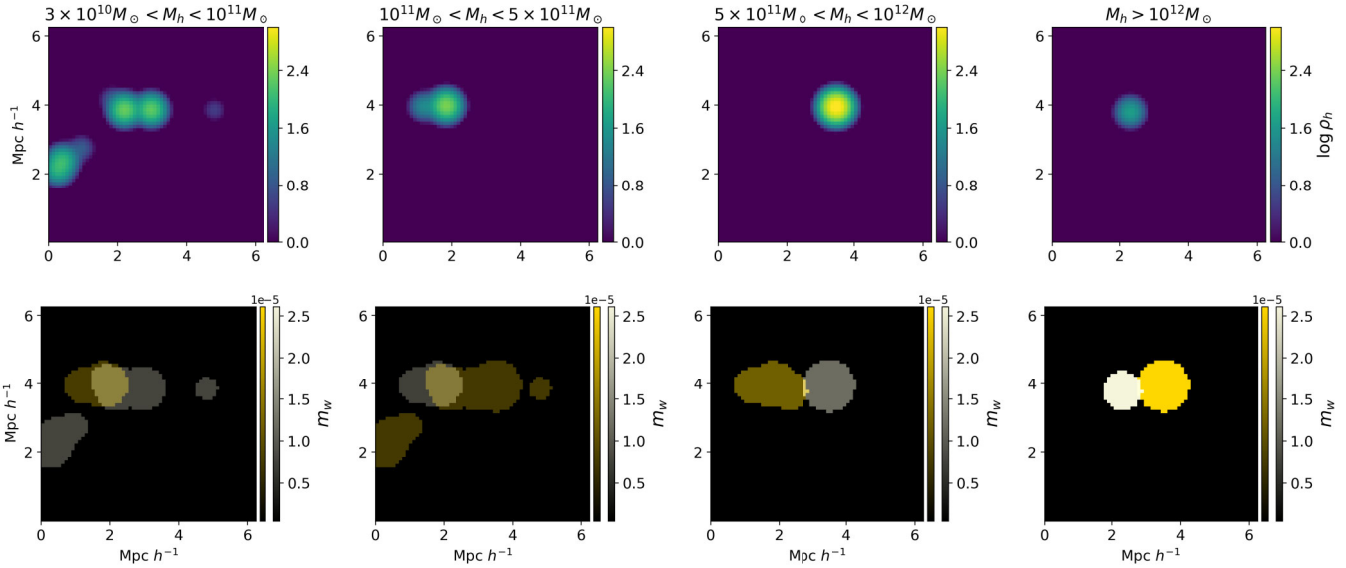

Figure 3. Illustration of the halo density channels masking prescription. The first row of four images shows a 2D slice of the training halo maps $\rho_{\rm h}^{(j)}$ for each halo mass channel j. The second row shows the corresponding 2D weighted mask maps, $m_w^{j,k}$. In each masking map, the contribution from the halo j is shown in beige, and the contribution from the neighbouring channel's halo location is shown in golden.

where $\boldsymbol{x}_0$ is the original data point and $\boldsymbol{x}_{\rm t}$ is the noisy version of $\boldsymbol{x}_0$ at time-step t, where $t \in \{1, 2, \ldots, {\rm T}\}$ and T is the total number of noising steps. Here, $q(\boldsymbol{x}_{\rm t} \mid \boldsymbol{x}_0)$ denotes the probability distribution of the noisy sample $\boldsymbol{x}_{\rm t}$, given an original data point $\boldsymbol{x}_0$, The notation $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicates a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The noise is added according to a schedule defined by $\alpha_{\rm t}$ and $\sigma_{\rm t}^2$. In the reverse process, also known as the generation process, the model learns to conditionally denoise this added noise in T steps with a general form:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{\rm t-1} \mid \boldsymbol{x}_{\rm t}) = \mathcal{N}\big(\boldsymbol{x}_{\rm t-1};\, \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_{\rm t}, {\rm t}),\, \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\boldsymbol{x}_{\rm t}, {\rm t})\big)\,. \tag{9}$$

Here, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{\rm t-1} \mid \boldsymbol{x}_{\rm t})$ denotes the learned reverse (generative) distribution parametrized by the denoising neural network parameters, $\boldsymbol{\theta}$. The distribution can also be parametrized as a multivariate Gaussian with covariance $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ and mean $\boldsymbol{\mu}_{\boldsymbol{\theta}}$, obtained by the same denoising neural network that was trained to predict $\boldsymbol{x}_{\rm t-1}$ given $\boldsymbol{x}_{\rm t}$. This formulation holds generally for both denoising diffusion probabilistic models (DDPMs) and VDMs, the latter being the framework adopted in this work. The loss function for this task minimizes an objective of the form:

$$\mathcal{L}_{\rm T}(\mathbf{x}) = \mathbb{E}_{\epsilon,{\rm t}}\left[(f_{\boldsymbol{\theta}}(\sigma_{\rm t}, \alpha_{\rm t})\, \|\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{x}_{\rm t}; t)\|_2^2\right]\,, \tag{10}$$

where $\boldsymbol{\epsilon}$ and $\hat{\boldsymbol{\epsilon}}$ are the added and predicted noise, respectively. Here, $\mathbb{E}_{\boldsymbol{\epsilon},{\rm t}}[\cdot]$ is the expectation value with respect to the Gaussian distribution of the noise $\boldsymbol{\epsilon}$ and time-step t sampled from the diffusion process, and $\|\cdot\|_2^2$ denotes the squared l_2 norm. The term $f_{\boldsymbol{\theta}}(\sigma_{\rm t}, \alpha_{\rm t})$ is a weighing function which depends on the noising variance and scaling factors. It weighs the relative contribution of a given denoising step at time t to the total loss, and in our case its explicit form is given in equation (13).

To inpaint the H I brightness temperature, $T_{\rm b}$, on to the halo distribution obtained in the previous step, we use the VDM of D. P. Kingma et al. (2021). The neural network architecture is modified to work with three-dimensional simulation boxes and follows a similar architecture as the one provided in V. Ono et al. (2024): 2D grouped convolutions are replaced with their 3D counterpart, and we add an extra convolutional layer to the residual blocks of the network. We set up the denoising architecture as a U-Net with residual networks and an attention block at the bottleneck, as illustrated in Fig. 4. This is because it has been shown that attention blocks at the bottleneck facilitate faster convergence and achieve the same quality of cross-correlation results with fewer training steps (V. Ono et al. 2024).

2.2.2 *Building the 21 cm temperature training map*

To produce the 21 cm temperature map for training, we proceed as follows: we obtain the neutral hydrogen fraction $x_{{\rm HI},i}$ and the gas particle mass $m_{g,i}$ [$\rm M_\odot$], for each gas particle i in the hydrodynamic simulation, from which we first compute the H I mass per particle, $M_{{\rm HI},i}$, as:

$$M_{{\rm HI},i} = m_{g,i} x_{{\rm HI},i} X\,, \tag{11}$$

where X is the hydrogen mass fraction, which in this CAMELS simulations is set to 0.76. In addition, since the simulations (as detailed below) do not include self-shielding effects for star-forming particles, we manually set $x_{{\rm HI},i} = 1$ for particles with star formation rate larger than zero. The reason for having a separate star formation rate parameter on top of the star particles is that in the underlying multiphase ISM model, self-shielded gas is represented as a gas element until stochastically converted to a star particle, where the conversion probability depends on the star formation time-scale (V. Springel & L. Hernquist 2003). This simple and effective prescription, also employed in the post-processing of the CAMELS simulations (F. Villaescusa-Navarro et al. 2022), assumes that all the star-forming particles are fully self-shielded against ionizing radiation. As shown in F. Villaescusa-Navarro et al. (2018), this assumption results in a good match to observational data sets like abundance of H I and column density distribution of Damped Lyman-α systems.

We then use the same CIC procedure described above to interpolate $M_{{\rm HI},i}$ to a continuous mass field, $M_{\rm HI}$, after which we divide each voxel by its volume to obtain the density field $\boldsymbol{\rho}_{\rm HI}$. The 21 cm brightness temperature in the post-reionization universe is given to

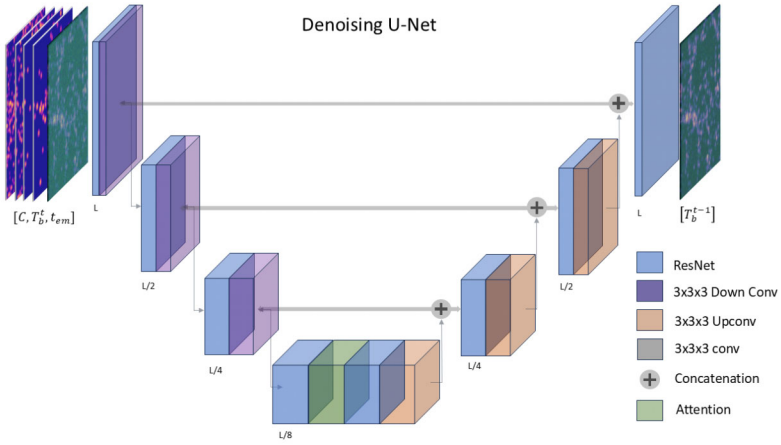

Figure 4. The architecture of the denoising model: a 3D U-net with residual network blocks and an attention layer at the bottleneck. The input to the network is the noisy temperature map at time step t, the time step embedding and the conditional halo maps. The output is a less noisy version of the temperature map.

a good approximation by the following expression in F. Villaescusa-Navarro et al. (2018):

$$T_{\rm b}(\mathbf{x}) = 189h \left(\frac{H_0(1+z)^2}{H(z)}\right) \frac{\rho_{\rm HI}(\mathbf{x})}{\rho_{\rm c}}\,[{\rm mK}]\,, \tag{12}$$

where H_0 is the Hubble–Lemaître constant today, G is the gravitational constant, $\rho_{\rm HI}(\mathbf{x})$ is the interpolated neutral hydrogen density at a given cell location $\mathbf{x}$, z is the redshift, $H(z)$ is the Hubble function at redshift z, $T_{\rm b}$ is expressed in units of mK, and $H_0 = 100h\,{\rm km\,s^{-1}\,Mpc^{-1}}$. The quantity $\rho_{\rm c} = \frac{3H_0^2}{8\pi G}$ denotes the critical density of the Universe today, i.e. the density required for a spatially flat Universe.

We will work in the plane-parallel approximation and assume the redshift of the box as a constant, given the fact that redshift evolution within the simulated volume would require a realistic set of lightcones mock and this is beyond the scope of the paper. Once the maps are created, we smooth them with a Gaussian kernel of fixed radius $R = 0.2$ Mpc h^{-1}. This choice balances two considerations: on the one hand, smoothing helps to stabilize training by reducing sharp fluctuations and noise in the target fields; on the other hand, excessive smoothing would wash out the small-scale structures that are physically important for our task. Given that the voxel resolution is $\approx$0.1Mpc h^{-1}, we chose a slightly larger smoothing radius that facilitates improved training of the network while preserving most of the relevant small-scale information.

2.2.3 *Loss function*

When training `LODI`, we first sample $\boldsymbol{T}_{\rm b}$ map from the training set and then apply to it the forward diffusion process of equation (8), setting $\boldsymbol{x}_0 = \boldsymbol{T}_{\rm b}$, $\boldsymbol{x}_{\rm t} = \boldsymbol{T}_{\rm b}^{\rm t}$, $\alpha_{\rm t}^2 = {\rm sigmoid}(-\gamma_{\boldsymbol{\eta}}(t))$ and $\sigma_{\rm t}^2 = {\rm sigmoid}(\gamma_{\boldsymbol{\eta}}(t))$, where the noising schedule $\gamma_{\boldsymbol{\eta}}(t) = b + w\,t$ with learnable parameters $\boldsymbol{\eta} = \{w, b\}$. For our work, we use $T = 50$.

During learning, the network predicts the noise that was added at step t by minimizing the diffusion loss:

$$\mathcal{L}_{\rm T}(\mathbf{T}_{\rm b}) = \frac{T}{2}\,{\rm E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),\,{\rm t}\sim\mathcal{U}(1,T)}\Big[\big(\exp(\gamma_\eta({\rm t}-1) - \gamma_\eta({\rm t})) - 1\big) \times \Big\|\epsilon - \hat{\epsilon}_\theta\big(\mathbf{T}_{\rm b}^{\rm t};\,{\rm t}\big)\Big\|_2^2\Big]. \tag{13}$$

where $\mathcal{L}_{\rm T}(\boldsymbol{T}_{\rm b})$ is the diffusion loss, which measures the error in noise prediction; where the expectation is taken over the noise distribution $\boldsymbol{\epsilon} \sim \mathcal{N}\big(\mathbf{0},\,\mathbf{I}\big)$ and ${\rm t} \sim \mathcal{U}[1, T]$, denoting the uniform discrete distribution. The noise predicted by the de-noising UNet is $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}\big(\boldsymbol{T}_{\rm b}^{\rm t};\,{\rm t}\big)$. Thus, in the forward process, noise is added until we obtain pure Gaussian noise at time-step T, when $\boldsymbol{T}_{\rm b}^{\rm T} \sim \mathcal{N}(\mathbf{0},\,\mathbf{I})$. During the generation process, we sample from a standard normal and denoise it in T steps using the learned distribution $p_{\boldsymbol{\theta}}\big(\boldsymbol{T}_{\rm b}^{\rm t-1} \mid \boldsymbol{T}_{\rm b}^{\rm t},\,\boldsymbol{\rho}_{\rm h},\,{\rm t}\big)$ to generate a sample $\boldsymbol{T}_{\rm b}^0 \sim p\big(\boldsymbol{T}_{\rm b}^0 \mid \boldsymbol{\rho}_{\rm h}\big)$. Sampling from this learned distribution is equivalent to performing ancestral sampling (D. P. Kingma et al. 2021), which in this case simplifies to:

$$\mathbf{T}_{\rm b}^{\rm s} = \frac{\alpha_{\rm s}}{\alpha_{\rm t}}\left(\mathbf{T}_{\rm b}^{\rm t} - \sigma_{\rm t}\,c\,\hat{\epsilon}_\theta\big(\mathbf{T}_{\rm b}^{\rm t};\,{\rm t}\big)\right) + \sqrt{(1-\alpha_{\rm s}^2)\,c}\,\epsilon \tag{14}$$

where $c = \exp\big(\gamma_{\boldsymbol{\eta}}({\rm s}) - \gamma_{\boldsymbol{\eta}}({\rm t}) - 1\big)$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $0 < s < t < T$.

2.3 Data sets, training, and validation

We use the CAMELS data set (F. Villaescusa-Navarro et al. 2022), which is a suite of hydrodynamical simulations created by varying cosmological and astrophysical parameters, along with different initial seeds. This is especially useful to test the generalization performance of the model trained on 3D cosmological dark matter density and brightness temperature maps. Each simulation box has a comoving side of length $25h^{-1}$ Mpc and contains 256^3 gas cells and 256^3 dark matter particles. CAMELS is based on the AREPO (R. Weinberger, V. Springel & R. Pakmor 2020) and GIZMO (P. F. Hopkins 2015) codes and run the same subgrid physics and feedback effects of IllustrisTNG and SIMBA simulations, respectively. The hydrodynamics is evolved with the moving-mesh code AREPO which combines a TreePM gravity solver with a finite-volume Godunov scheme on an unstructured Voronoi mesh. Gas cells above a density threshold are treated with the multiphase star-formation model of V. Springel & L. Hernquist (2003). Stellar winds and black-hole seeding/growth with AGN feedback follow the IllustrisTNG prescriptions. The particles are evolved from a redshift of $z = 127$.

For this work, we train on the CV set of IllustrisTNG, which contains 256^3 gas and dark matter particles (which are later subdivided into 256^3 voxels), all run with the cosmological and astrophysical parameters set to: $\Omega_m = 0.3, \sigma_8 = 0.8, A_{\rm SN1} = A_{\rm SN2} = A_{\rm AGN1} = A_{\rm AGN2} = 1.0$, Here, Ω_m denotes the present-day matter density parameter and σ_8 is the root-mean-square amplitude of linear matter fluctuations on scales of $8h^{-1}$ Mpc. $A_{\rm SN1}$, $A_{\rm SN2}$ are the supernova

feedback process parameters of the model and $A_{\rm AGN1}$, $A_{\rm AGN2}$ are the active galactic nuclei feedback process parameters, with values of 1.0 corresponding to their fiducial calibration in IllustrisTNG. There are 27 different initial seeds for each of the 27 simulations.

2.3.1 *Training of* HALOgen

To train HALOgen, we use one single simulation from the CV set and test on other simulations in the same set, with different initial seeds, but same simulation parameters. Both dark matter and halo maps are divided into 64^3 sub-volumes, rotated, and augmented to get rotationally-equivariant training.

We input the 64^3 voxels $\boldsymbol{\rho}_{\rm DM}$ field in batches of 16 and apply the loss function in equation (5) to the predicted output, using the weighted mask in equation (2). The optimizer used for training is RMSprop (Root Mean Square Propagation), an adaptive gradient-based optimization algorithm that scales the learning rate for each parameter by a running average of its recent squared gradients, a standard technique in deep learning architectures. The learning rate schedule is set as follows: the first 10 epochs are run with a learning rate of 2×10^{-4}; after that, we use Cosine Annealing Warm Restarts with the maximum and minimum values set to 10^{-4} and 3×10^{-5}, respectively, with a repeat cycle of 50 epochs. The cosine annealing warm restarts changes the learning rate between the maximum and minimum value, with a cosine decay, while repeating this cycle every 50 epochs in our case. We run the model for 260 epochs, and store the model's weights at the point in which the validation loss is lowest. We subdivide the data into training and validation set, with a 75 per cent and 25 per cent split. In equation (5), the coefficient λ_w of the regularization term $L_1^{\rm reg}$ is set to 10^{-6}. For the results in the paper, we use $\alpha = 0.6, q = 0.7$ and $\beta^j = [1, 2, 5, 8]$. The above hyper-parameters have been chosen to achieve the best predictions, after extensive testing.

2.3.2 *Training of* LODI

We train LODI on five simulations of the CV set from CAMELS, each with a different initial seed. We process the five simulations as explained above, and divide them into 32^3 voxel sub-volumes. To enforce rotational equivariance, we augment each simulation cube with all symmetry-equivalent flips and 90° rotations. Training on more simulations is necessary to account for cosmic variance in the large scales comparable to the box size and also to train on more haloes of large masses, especially $M_{\rm halo} > 10^{12}\,{\rm M}_\odot$. This expands the raw training set substantially and would correspondingly increase training time. We therefore apply a targeted sub-sampling strategy: because H I emission is concentrated around dark matter haloes – particularly massive ones – and thus sparse, we preferentially draw sub-volumes that contain objects in the higher mass halo channels. We define important regions here as regions where $\log \rho_h \geq 10$, since this density value is about 10^3 times smaller than the minimum density value in our data set, prior to Gaussian smoothing. Values smaller than that are small enough that they do not contribute to any meaningful H I content. To maintain diversity we also include 50 per cent of sub-volumes centred on the more numerous lower mass haloes, the first two mass bins in this case. This balanced selection reduces the total training data sets from ≈410 000 to ≈250 000, almost halving the training time without degrading the representativeness of the data, while increases the relative importance of the larger mass haloes as it is seen more frequently by the neural network. We finally create a training and validation split of 75 per cent and 25 per cent, respectively.

2.4 Generation over larger volumes with latent overlap

It is not possible to input the entire $25^3\,({\rm Mpc}/h)^3$ simulation box, corresponding to 256^3 voxels to our network, because of memory constraints. Such data cube would require several terabytes of GPU memory to train. Instead, we divide the domain into 512 smaller 32^3 voxels sub-volumes, with the view of generating for each its H I density field with the trained the diffusion model, and then combine them at the end to generate the full volume.

2.4.1 *Latent overlap method*

The naive solution would be to generate the brightness temperature field in each 32^3 sub-volume, then tile them to create the target, 256^3 volume – we call this approach 'tiling'. However, tiling introduces artificial, periodic discontinuities at the boundaries of each 32^3 voxel cube, arising because the brightness temperature field $\boldsymbol{T}_{\rm b}$ is sampled from the conditional distribution $\boldsymbol{T}_{\rm b} \sim p\left(\boldsymbol{T}_{\rm b} \mid \boldsymbol{\rho}_{\rm h}\right)$, which depends solely on the 32^3 halo field $\boldsymbol{\rho}_{\rm h}$, with no reference to adjacent boxes.

To address this, we adapt the inpainting method of A. Lugmayr et al. (2022), which in our work we call 'latent overlap' approach. The first box, B_1, is generated by starting with a Gaussian noise $\boldsymbol{\epsilon_1}$, to obtain a sample $\boldsymbol{T}^0_{\rm b,\,B_1}$. While generating this box in $T = 50$ steps, a set of T intermediate latent maps are produced using equation (14): $\left[\boldsymbol{T}^1_{\rm b,\,B_1}, \boldsymbol{T}^2_{\rm b,\,B_1}, \ldots, \boldsymbol{T}^{\rm T}_{\rm b,\,B_1}\right]$. These intermediate latent images can be interpreted as discrete snapshots of the evolving $\boldsymbol{T}_{\rm b}$ distribution under the diffusion process, forming a trajectory from random noise to the final sampled field.

When we next consider an adjacent box, B_2, we want to enforce that its generation process follow a 'similar' trajectory as B_1. For this, we partially overlap boxes B_2 on to B_1 such that

$$\mathrm{Vol}(B_1 \cap B_2) = V_{\rm overlap},$$

where $V_{\rm overlap}$ is the overlapping volume and is defined as the fraction of one box (specified below). For our work, the overlap volume or fraction is specified in Section 2.4.2. Now, at every tth denoising step of B_2, we replace $\boldsymbol{T}^{\rm t}_{\rm b,\,B_2}$ with the previously-generated $\mathbf{T}^t_{B_1}$ in the region $V_{\rm overlap}$, thus obtaining $\mathbf{T}^{t*}_{B_2}$, which is identical to $\mathbf{T}^t_{B_1}$ in the overlap volume. This new $\mathbf{T}^{t*}_{B2}$ is then denoised using the denoising U-net to obtain $\mathbf{T}^{t-1}_{B_2}$. Mathematically, the updated state of B_2 at the tth denoising step is given by:

$$\mathbf{T}^{\rm t*}_{\rm b,\,B_2} = M_{\rm overlap}\,\mathbf{T}^{\rm t}_{\rm b,\,B_2} + \left(1 - M_{\rm overlap}\right)\,\mathbf{T}^{\rm t}_{\rm b,\,B_1}.$$

where $M_{\rm overlap}$ is the overlap mask defined as:

$$M_{\rm overlap}(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in V_{\rm overlap} \\ 0, & \text{otherwise} \end{cases}$$

and

$$\boldsymbol{T}^{\rm t}_{\rm b,\,B_1} = \alpha_{\rm t}\,\boldsymbol{T}^0_{\rm b,\,B_1} + \sigma_{\rm t}\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0},\,\mathbf{I}).$$

where α_i and σ_i are the noise schedule parameters and are learnt by the diffusion model as explained in Section 2.2.3, and $\boldsymbol{T}^0_{B_1}$ was produced by the diffusion model in a previous backward pass.

Performing this latent overlap step ensures that the entire image $\boldsymbol{T}^{i*}_{B_2}$ is denoised harmoniously, thereby eliminating boundary artefacts, while producing a non-overlapping region that fits seamlessly with the overlapping region. We carry out this procedure T times, and at each step B_2 is pushed closer to the distribution of B_1 by forcing it to align with the overlapping region of B_1 while moving along the trajectory of $\boldsymbol{T}^i_{B_1}$. This latent overlap method between boxes B_1 and B_2 is shown in Algorithm 1, while Fig. 5 provides a comparison

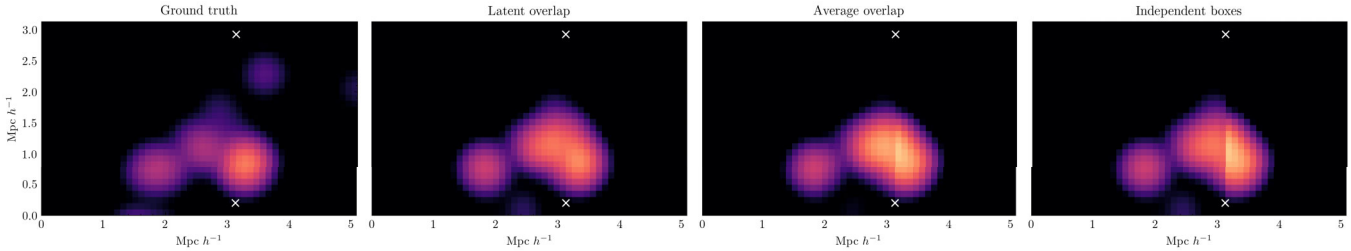

Figure 5. A comparison of different overlapping methods are shown, where the line of discontinuity is vertical and passes through the two marked crosses in the images. The left image is the ground truth. The latent overlap method is shown in the left centre, compared with two other cases. The centre right is the case where for the overlap region, we take the average pixel values of the two boxes and the right image shows the other case when we generate two separate boxes and just concatenate them, while keeping the the overlapping pixel region from the first box. The images shown are averaged in the third axis over a distance of about 0.9 Mpch^{-1}.

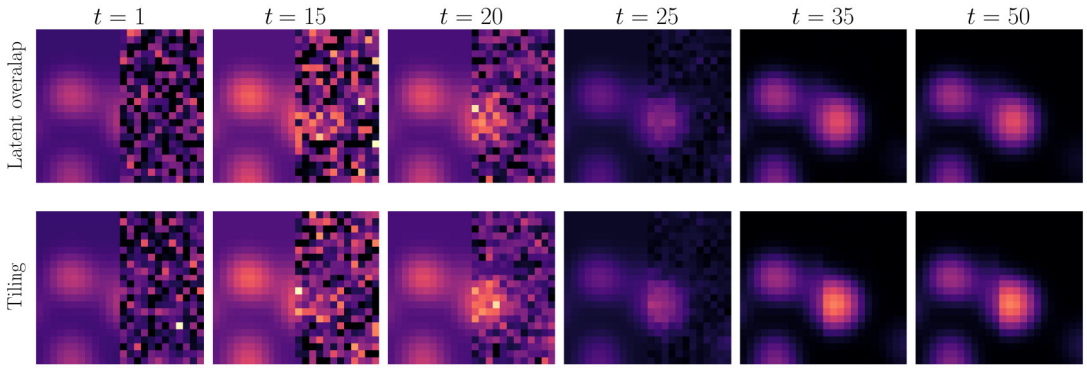

Figure 6. Illustration of `LODI`, using the latent overlap method: output of the diffusion process at different steps in the denoising process (top row, left to right) when using the latent overlap method on a 2D illustrative slice. At each time step, the left half of the field has already been generated. The bottom row shows the same time steps but with simple tiling and no latent overlap: the discontinuity at the boundary is apparent.

of the results obtained with latent overlap against simple tiling of independent boxes. In principle, as T is increased, the alignment should improve. Fig. 6 shows the diffusion process implementing this algorithm on a 2D patch of overlapping boxes. In this case, the left box has already been produced and does not change throughout the entire process.[3] We call this approach `LODI`, for Latent Overlap Diffusion for IM.

Algorithm 1 Latent overlap method

$\boldsymbol{T}^{\mathrm{T}}_{\mathrm{b,\,B_2}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$\boldsymbol{\epsilon}_{\mathrm{B_1}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Compute M_{overlap}
for $i = \mathrm{T}-1, \ldots, 0$ **do**
  $s(i) = i$ **and** $t(i) = i+1$
  $\boldsymbol{T}^{\mathrm{t}}_{\mathrm{b,\,B_1}} = \alpha^{\mathrm{t}} \boldsymbol{T}^{0}_{\mathrm{b,\,B_1}} + \sigma^{\mathrm{t}} \boldsymbol{\epsilon}_{\mathrm{B_1}}$
  $\boldsymbol{T}^{\mathrm{t*}}_{\mathrm{b,\,B_2}} = M_{\mathrm{overlap}}\, \boldsymbol{T}^{\mathrm{t}}_{\mathrm{b,\,B_2}} + (1 - M_{\mathrm{overlap}})\, \boldsymbol{T}^{\mathrm{t}}_{\mathrm{b,\,B_1}}$
  $\boldsymbol{\epsilon}_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\boldsymbol{T}^{\mathrm{s}}_{\mathrm{b,\,B_2}} = \frac{\alpha_{\mathrm{s}}}{\alpha_{\mathrm{t}}} \left(\boldsymbol{T}^{\mathrm{t*}}_{\mathrm{b,\,B_2}} - \sigma_{\mathrm{t}}\, c\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}} \left(\boldsymbol{T}^{\mathrm{t*}}_{\mathrm{b,\,B_2}}; \mathrm{t} \right) \right) + \sqrt{(1 - \alpha_{\mathrm{s}}^2)\, c}\, \boldsymbol{\epsilon}_2$
end for
return $\boldsymbol{T}^{0}_{\mathrm{b,\,B_2}}$

[3]Note that left half of the image does not change during the process, despite the misleading impression given by the colourbar boundaries being adjusted from left to right to accommodate the different dynamic range of pixel values.

2.4.2 *Patching 21 cm fields*

To generate the entire 25^3 (Mpc/h)^3 volume, we need to define the sequence in which the sub-volumes B_i cubes are processed. Ideally, we aim for an algorithm that enables parallelization of the process, for faster generation, while using the largest possible number of starting sub-volumes (i.e. the ones that are generated conditionally on the learnt diffusion process, but without reference to adjacent regions), because these are the ones that are truly in-distribution w.r.t. to the diffusion generative process.

To execute this, we create a mesh of 125 sub-volumes B_i of size 32^3 voxels, equally spaced from each other, arranged in a 3D grid (Stage 1, top panels in Fig. 7, showing from left to right the three orthogonal planes, $X-Y$, $X-Z$, $Y-Z$; maps are averaged along the third axis). In Stage 2, 3, and 4, we generate nearest-neighbour boxes along each of the three cartesian direction using the latent generation method with an overlap of $4 \times 32 \times 32$ voxels. This greatly reduces the out-of-distribution error, and helps with parallelizing the generation process since most of the boxes can be generated independently of each other. Consequently, the configuration in Stage 4 exhibits an alternating pattern where every other B_i is present along each axis. In each of these three stages, 100 new B_is are generated per axis, amounting to 300 new generations in total.

In the subsequent Stages (5, 6, and 7), additional sub-volumes are generated adjacent to the ones generated in stages 2, 3, and 4 along the cartesian axes, overlapping with the original sub-volume with the same overlap fraction of $4 \times 32 \times 32$. This procedure is performed sequentially for the three cartesian directions, requiring 80 cubes per axis (240 in total). Finally, the remaining 64 sub-volumes are

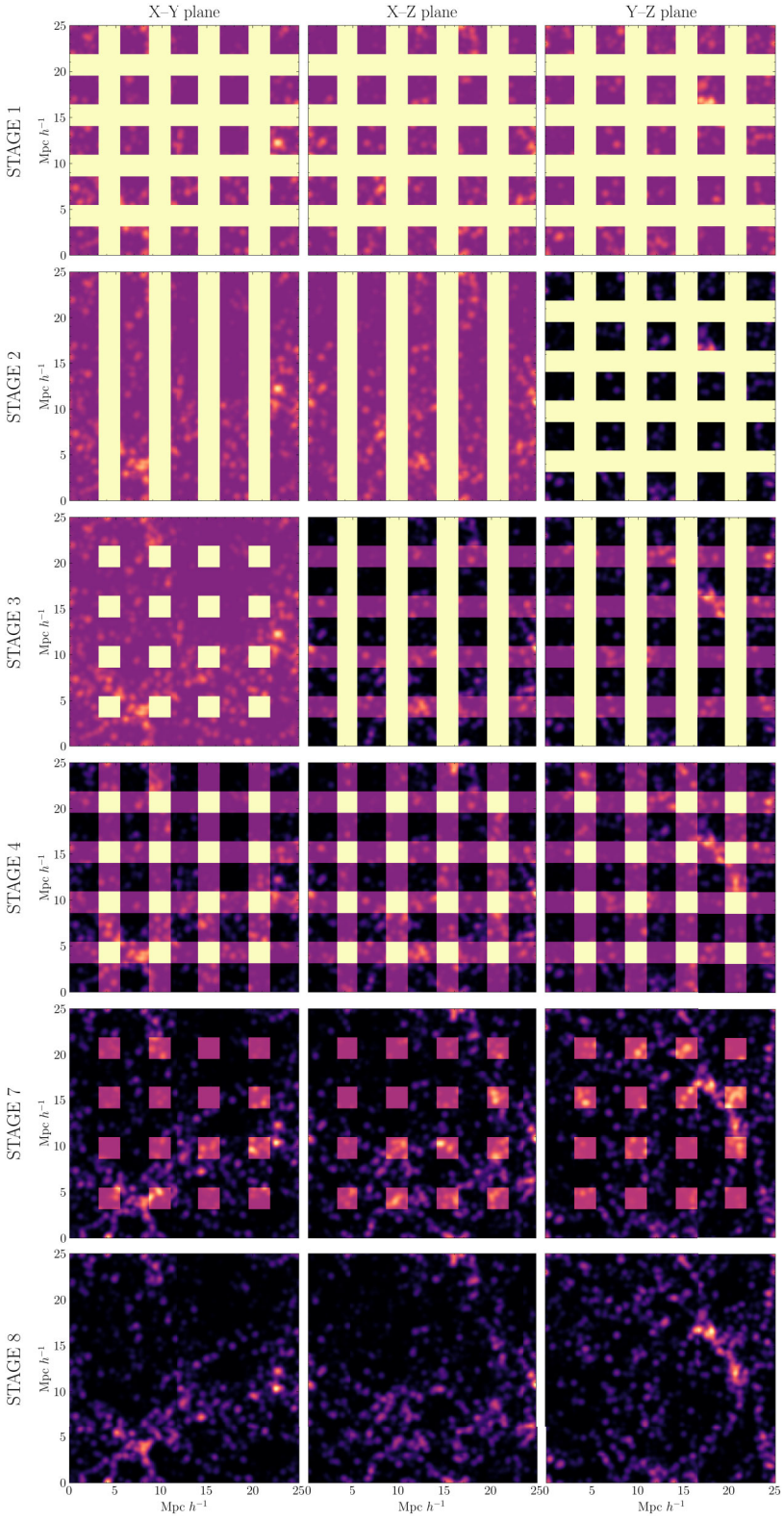

Figure 7. Assembly of the full volume cube of voxel size 256^3 from $9 \times 9 \times 9$ sub-volumes using the latent overlap method, for a simulation from the validation set. Each row corresponds to a different stage (top to bottom) of the latent generation process and each column displays one of the three orthogonal 3D cartesian planes of the simulation volume (the maps are shown by projecting along the third spatial axis across the full 25Mpc h^{-1} extent of the simulation box). Although the figure displays six rows, the complete process consists of eight sequential steps. The fifth row represents the net result of three consecutive sub-steps, which have been aggregated for clarity.

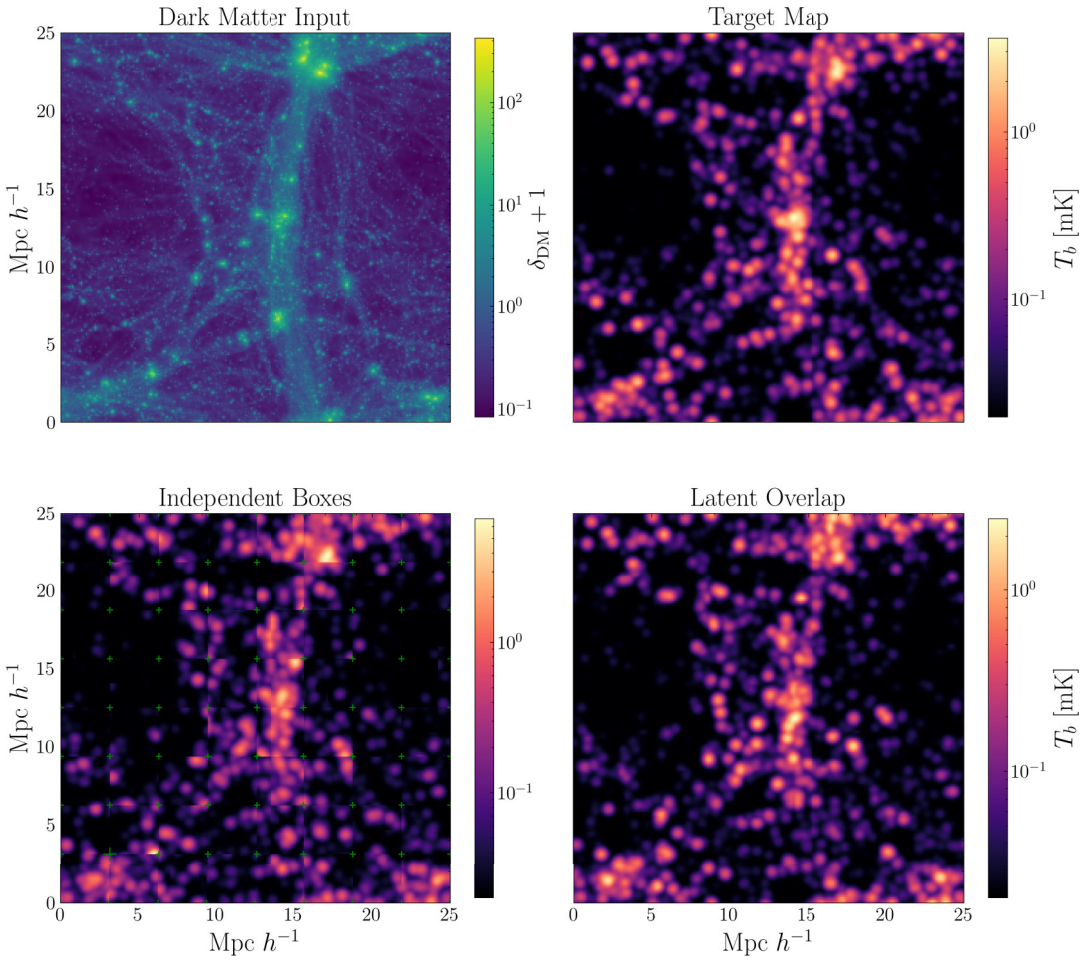

Figure 8. Top left: The dark matter density map, not seen during training, used as input for the generative pipeline. Top right: the ground truth (target) H I brightness temperature map from the simulation. Bottom left: The generated output using the Tiling method, with green markers highlighting the sub-volumes boundaries where discontinuities arise. Bottom right: brightness temperature map generated using the latent overlap method of `LODI`, which significantly reduces the discontinuities seen in the tiling approach. The brightness temperature map is conditioned on the output provided by `HALOgen`. All the maps are shown by projecting along the third spatial axis across the full 25Mpc h^{-1} extent of the simulation box.

generated at stage 8, each of which overlaps with the neighbouring six voxels with the same overlap volume of $4 \times 32 \times 32$ to complete the configuration, resulting in a final assembly of 729 B_is arranged in a $9 \times 9 \times 9$ grid resulting in a total volume of $25\,(\mathrm{Mpc}/h)^3$. This boosts the speed of generation by a factor of $\approx$6, compared to generating sub-volumes sequentially. In principle, we could further parallelize the generation process by combining the different stages into one, but this is limited by the available GPU memory.

The overall output of our generative pipeline is shown in Fig. 8. We start with an unseen dark matter density field, shown in the top left panel, run it through `HALOgen` (whose intermediate halo mass density maps are shown in the second column of Fig. 9), and then produce the brightness temperature map with `LODI` (bottom right panel of Fig. 8). Using the naive Tiling approach (bottom left panel of Fig. 8), the discontinuity between training sub-volumes is evident (corners of boundaries are indicated by green crosses for better visualization).

3 RESULTS

In this section, we demonstrate the performance of the entire generative pipeline, producing H I intensity maps from previously unseen dark matter simulation maps. We validate our results using the CV set of the IllustrisTNG simulations at $z = 0$, with the same parameters for the simulations as the training maps, but with different initial seeds. We postpone the study of generalization capabilities to other cosmologies to a future, dedicated work.

3.1 Performance of `HALOgen`

The output of `HALOgen` is shown in Fig. 9, split in the four mass channels (top to bottom), with the input dark matter map shown on the top left of Fig. 8. We reiterate that this input map was unseen during training, as it was generated with a different seed (but same cosmological and astrophysical parameters as the training map).

Visually, the predicted pixel values for the halo mass density (left column) look very close to the target maps (centre column), except for the smallest mass range (first row). This can be more precisely quantified by comparing the distribution of log-density values between the target and predicted $25^3\,(\mathrm{Mpc/h})^3$ box (right column). While we observe a consistent underprediction in the mean value of the smallest halo mass range (albeit still within the 16–84 percentile band), the agreement for the other mass ranges is excellent. We trace back the relatively poorer performance for the smallest

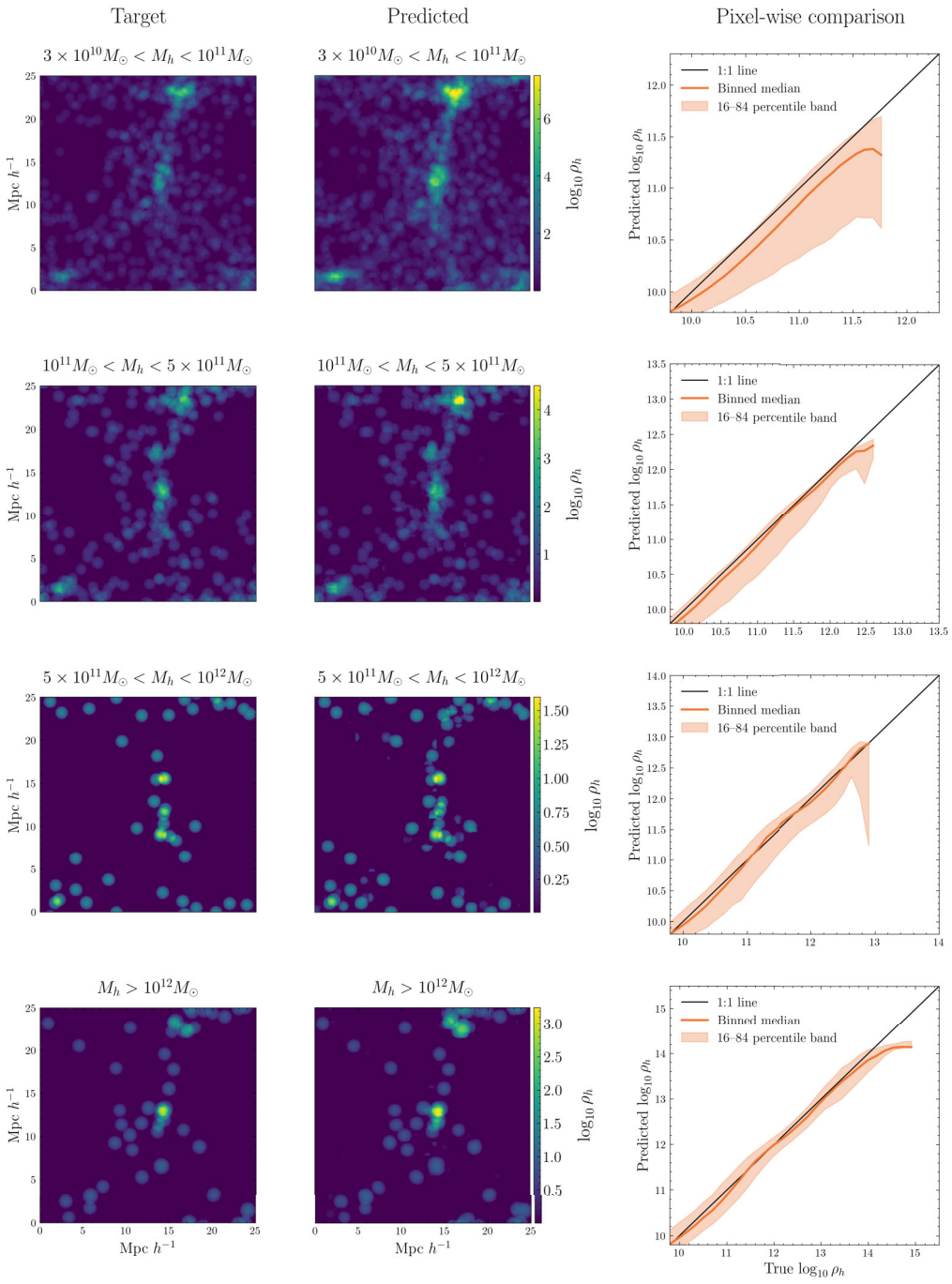

Figure 9. Output of HALOgen from a previously unseen dark matter simulation. The rows show the comparison between the target (left) and predicted (centre) halo mass density maps for each of the four mass channels (top to bottom). The halo maps are visualized by projecting along the third spatial axis across the entire 25Mpc h^{-1} of the simulation. The right column gives a pixel-wise comparison between the distribution of log-densities for the target and predicted map, with the mean and 1σ band computed from the 256^3 voxels.

mass halo density to our definition of masked loss: as discussed above, a lower weight is given to pixels of the smallest haloes, since they are the most abundant ones. Therefore, we can expect that the loss function puts greater attention to faithfully reconstructing the density of larger mass haloes. We plot the median instead of the mean since in the post-processing of the output halo maps, we set pixel values below a threshold of $\log \rho_h = 9.5$ to zero, which in the pixel comparison plot skews the mean value of the pixel distribution. The median values are agnostic to these outlier near zero values. The reason for setting this cut-off is because such low predicted values are not possible given the training data and lie below the lower density limit.

3.2 Performance of LODI

We define the dimensionless brightness temperature field by normalizing each spatial point $T_b(\mathbf{x})$ by its global mean $\langle T_b \rangle$:

$$\hat{T}_b(\boldsymbol{x}) = \frac{T_b(\boldsymbol{x})}{\langle T_b \rangle} . \tag{15}$$

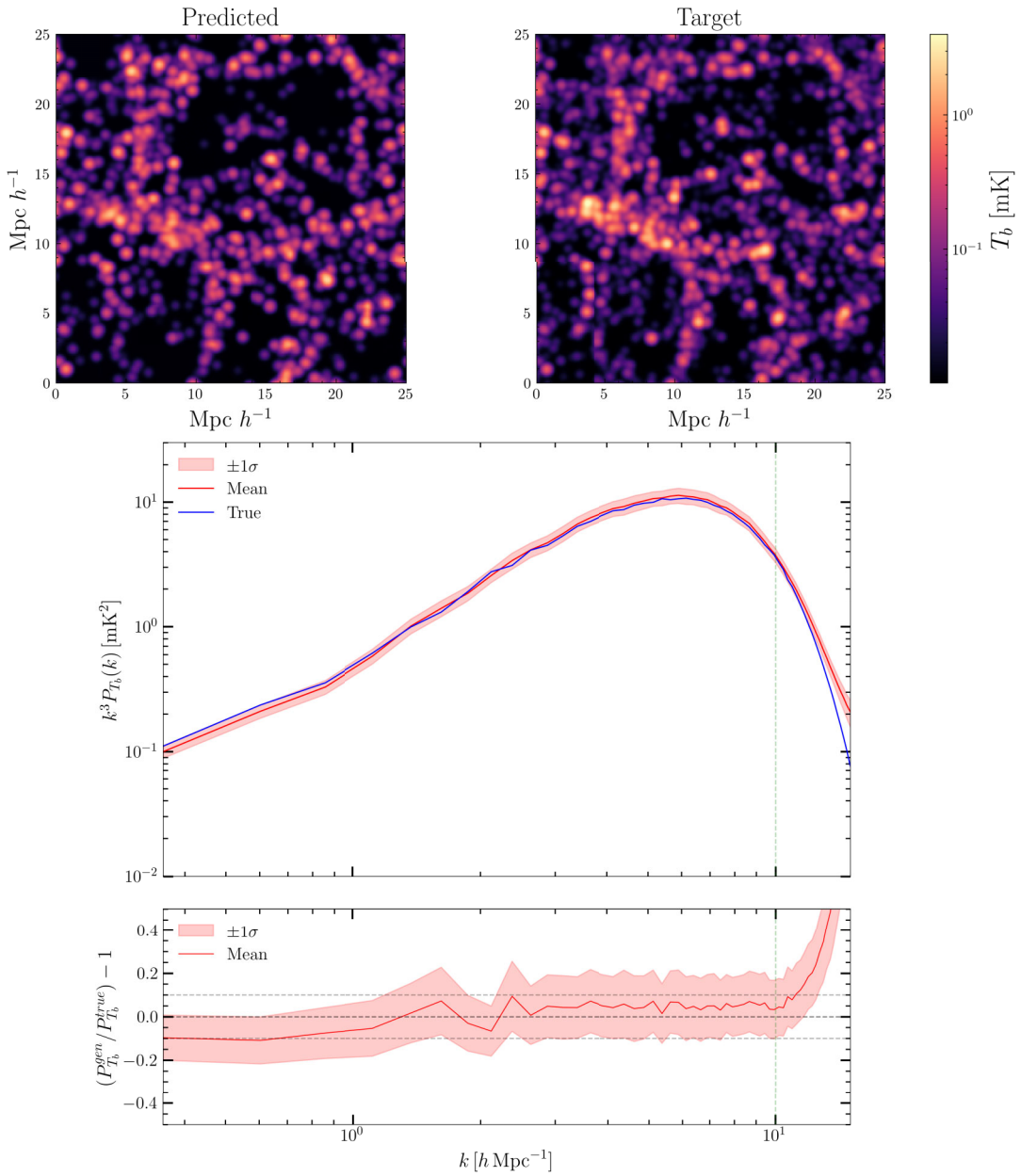

Figure 10. Generation of brightness temperature maps with LODI: one sample of diffusion-generated map (top left) compared to the unseen ground truth (top right) visualized by averaging over the third spatial axis, along the entire 25Mpc h^{-1} length of the simulation box. The bottom panel compares the mean power spectrum and its standard deviation obtained from 20 diffusion realizations, each produced with a different seed for the diffusion model, showing good agreement well into the non-linear regime, up to scales $k \simeq 10h$ Mpc^{-1}, also visible in the residual spectra shown in the bottom panel. The maps are smoothed with a Gaussian kernel of radius of 0.2 h Mpc^{-1}.

The dimensionless power spectrum is accordingly defined as:

$$P_{\hat{T}_b}(k) \;=\; \left\langle \hat{T}_b(\mathbf{k}) \cdot \hat{T}_b^*(\mathbf{k}) \right\rangle, \tag{16}$$

where $\hat{T}_b(\mathbf{k})$ is the Fourier transform of $\hat{T}_b(\mathbf{x})$. This dimensionless formulation allows for a direct comparison of small-scale fluctuations in the brightness temperature field across different simulations.

To assess the performance of LODI, we analyse the 21cm power spectrum produced by our pipeline and its residual,

$$P_{\hat{T}_b}(k) = \left\langle \hat{T}_b(\boldsymbol{k})\, \hat{T}_b^*(\boldsymbol{k}) \right\rangle .$$

where $P^{\rm gen}_{\hat{T}_b}(k)$ and $P^{\rm true}_{\hat{T}_b}(k)$ denote the LODI-generated and ground truth (from the simulation) 21 cm dimensionless power spectrum, respectively. Since the diffusion model was trained on Gaussian smoothed maps, the comparison plots and power spectra calculations are also done by Gaussian smoothing the ground truth maps with the same filter radius of $R = 0.2$ Mpc h^{-1}. Note that when testing our model, we compare against maps that include H I from all the haloes across the full halo-H I mass range. This allows us to verify that the specific halo-mass threshold used during training does not significantly affect the recovered 21cm signal.

Fig. 11 illustrates the end-to-end performance by showing the residual power spectrum obtained by applying the whole pipeline on

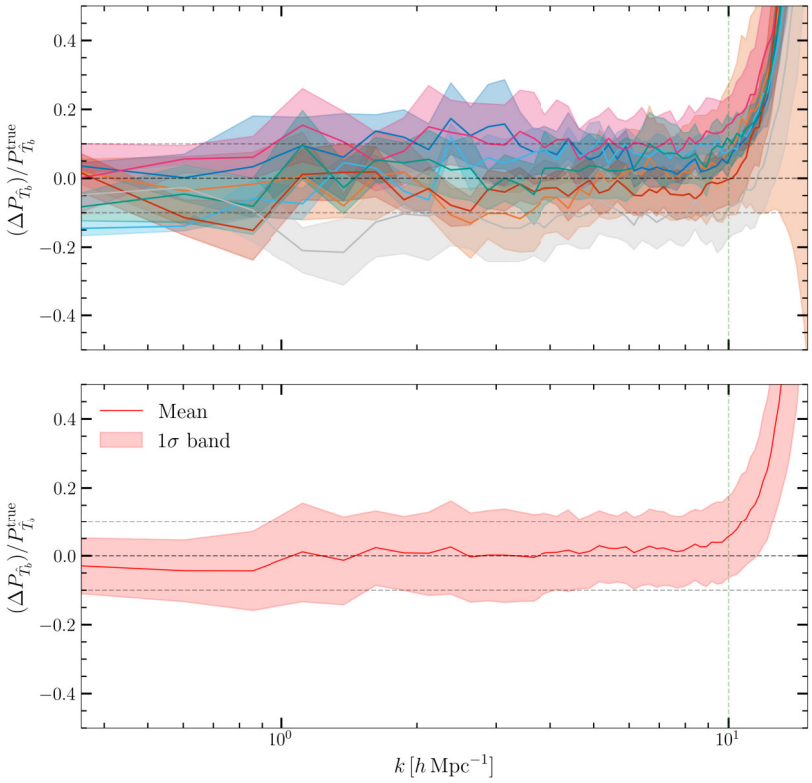

Figure 11. Top: mean (lines) and standard deviation (shaded bands) of the residual $T_{\rm b}$ power spectrum for seven dark matter simulations (each corresponding to a different random seed), with five diffusion realizations each. The bottom plot shows the result when combining all the realizations.

seven previously unseen dark matter density fields (i.e. with different initial seeds). For each of the seven dark matter fields, we first produce the halo maps for each using `HALOgen`, and then generate five realizations of brightness temperature maps.

In Fig. 10, we present an example $T_{\rm b}$ map generated using the full pipeline, alongside its corresponding ground-truth map, unseen before during training, each averaged along the third axis. We also plot the dimensional power spectrum $P_{T_{\rm b}}$, computed from 20 `LODI` realizations conditioned on the same halo maps produced by `HALOgen`. Visually, the generated and true maps show considerable matching, and the power spectra remain in good agreement up to $k \simeq 10h$ Mpc^{-1}, as can also been in the residual values shown in the bottom panel. This result showcases the intrinsic variance of the generative model `LODI`, as seen in the spread of the power spectra residuals. At high wavenumbers the predicted power spectrum exhibits excess power relative to the truth. This arises because the generated 21 cm maps display more compact radial profiles than the true maps, which are comparatively more extended and diffuse. Since `LODI` is conditioned on halo maps only, the network tends to concentrate 21 cm emission within halo regions, producing higher central intensities and a steeper radial decline. This enhanced contrast amplifies small-scale variance and leads to the observed excess power at higher wave modes.

To generate these results, `HALOgen` was trained for about 21 h on a single A100 GPU. The training of `LODI` took 600 000 iterations, using a batch size of 16. The total training time was around 30 h, using a single A100 GPU.

When generating, the total time taken by the pipeline to produce one temperature map for the whole 256^3 volume starting from the dark matter field is 106 s (16 s for `HALOgen`, 90 s for `HALOgen`).

4 CONCLUSIONS

In this work, we presented a two-steps generative pipeline to generate and inpaint large, three-dimensional 21 cm intensity maps, starting from dark matter only simulations. Although this work applied the method to 25^3 $({\rm Mpc}/h)^3$ simulations, it can be extended to inpaint simulations of any size, provided the resolution is the same as the training set simulation. In the first step, the `HALOgen` algorithm uses a U-Net to produce high-fidelity halo maps, subdivided in mass channels (bins); in the second step, `LODI` uses a conditional VDM coupled with a latent overlap method to paint the 21 cm signal on top of the halo field. We also developed a method to parallelize the generation of sub-volumes of intensity maps coherently into a larger volume simulation, without having to modify the diffusion model. Thus, this method is agnostic to the type of diffusion model used. A more powerful and general method is being investigated and will be demonstrated in a dedicated future work.

To create the halo maps, we use an attention ResUnet along with a weighted masking technique, to enforce halo separation based on halo mass. In this work, we create four halo maps from a single dark matter density field, which we then use as a conditional in our diffusion model – a VDM that learns to denoise a Gaussian field, conditioned on the four halo maps, to produce the 21 cm intensity maps at redshift $z = 0$.

The pipeline produces statistically unbiased estimates of the 21 cm power spectrum, and achieves a $\leq$10 per cent accuracy on average, up to $k \simeq 10\,h$ Mpc^{-1}, as can be seen in Fig. 11. We test the model on multiple unseen simulations, with different initial seeds, showcasing the robustness of the model to capture and reproduce features well into the non-linear small-scale regime.

Although the model performs well on simulations that share the same redshift, cosmology, and astrophysical prescriptions, it is essential to test its ability to generalize across various cosmological parameters and redshift. Such a robust model would enable parameter inference and simulation-based, likelihood-free analyses, and would allow us to build full past-light-cone mocks, that could be compared with data, potentially increasing the information retrieval many-folds compared with simpler (and lossy) summary statistics, like the power spectrum. In addition, forthcoming large-volume H I surveys will enable high-signal-to-noise cross-correlations between 21 cm intensity maps and galaxy catalogs (S. Cunnington 2022), an approach that mitigates foreground systematics and can tighten constraints on both cosmology and galaxy astrophysics. Our pipeline can therefore be extended to produce realistic H I mocks for these surveys, providing a test-bed for cross-correlation analyses and improved joint constraints on cosmological and astrophysical parameters. We also foresee to extend the pipeline presented here by coupling `HALOgen` and `LODI` with `JERALD`, a Lagrangian deep learning method that produces high-resolution dark matter, stellar mass and neutral hydrogen maps from lower resolution approximate *N*-body simulations (M. Rigo, R. Trotta & M. Viel 2025), thereby further accelerating the end-to-end analysis.

Furthermore, it will be interesting to investigate adapting the diffusion model to cope with realistic foregrounds and instrumental nuisances (like the role of the beam, wedge reconstruction, etc.), both in terms of cross-correlation with galaxies and auto-correlation of the 21 cm brightness temperature. Finally, from the fundamental physics perspective, the excellent performance of `LODI` even at very small scales makes it an ideal tool to explore non-standard dark matter scenarios, in which the dark matter could have thermal velocities or interactions with baryons and radiation.

ACKNOWLEDGEMENTS

The authors thank Danijel Skočaj for helpful suggestions that inspired the methods adopted, as well as Bruce Bassett, Daniela Breitman, Kosio Karchev, David Prelogovič, Mauro Rigo, Giulio Scelfo, Francisco Villaescusa-Navarro, and Gabrijela Zaharijas for discussions. SM thanks the Flatiron CCA and University of Nova Gorica for hospitality.

SM is supported by the National Recovery and Resilience Plan (PNRR), Dottorati Green/Innovazione under DM 351 and also acknowledges support from the SISSA-Flatiron Exchange Programme. RT acknowledges co-funding from Next Generation EU, in the context of the National Recovery and Resilience Plan, Investment PE1 Project FAIR 'Future Artificial Intelligence Research'. This resource was co-financed by the Next Generation EU [DM 1555 del 11.10.22]. RT and MV are partially supported by the Fondazione ICSC, Spoke 3 'Astrophysics and Cosmos Observations', Piano Nazionale di Ripresa e Resilienza Project ID CN00000013 'Italian Research Center on High-Performance Computing, Big Data and Quantum Computing' funded by MUR Missione 4 Componente 2 Investimento 1.4: Potenziamento strutture di ricerca e creazione di 'campioni nazionali di R&S (M4C2-19)'– Next Generation EU (NGEU). Part of the simulations were postprocessed on the Ulysses supercomputer at SISSA. MV and RT are also partially supported by the INFN INDARK grant.

No generative AI was used in the writing of this article.

DATA AVAILABILITY

The full source code, implemented in PYTHON using PyTORCH, along with usage instructions and notebook scripts is publicly available at https://github.com/satvik-97/LODI. All data used to produce the results in this paper are available upon request.

REFERENCES

Alam S. et al., 2017, MNRAS, 470, 2617
Anderson C. J. et al., 2018, MNRAS, 476, 3382
Andrianomena S., Villaescusa-Navarro F., Hassan S., 2024, Cosmological multifield emulator. https://arxiv.org/abs/2402.10997
Ansari R. et al., 2012, A&A, 540, A129
Bandura K. et al., 2014, Proc. SPIE Conf. Ser. Vol. 9145, Ground-based and Airborne Telescopes V. SPIE, Bellingham, p. 914522
Battye R. A., Browne I. W. A., Dickinson C., Heron G., Maffei B., Pourtsidou A., 2013, MNRAS, 434, 1239
Battye R. A., Davies R. D., Weller J., 2004, MNRAS, 355, 1339
Berti M., Spinelli M., Haridasu B. S., Viel M., Silvestri A., 2022, J. Cosmol. Astropart. Phys., 2022, 018
Berti M., Spinelli M., Viel M., 2024, MNRAS, 529, 4803
Berti M., Spinelli M., Viel M., 2024, MNRAS, 529, 4803, http://arxiv.org/abs/2309.00710
Bharadwaj S., Nath B. B., Nath B. B., Sethi S. K., 2001, J. Astrophys. Astron., 22, 21
Bull P. et al., 2016, Phys. Dark Univ., 12, 56
Carucci I. P., Villaescusa-Navarro F., Viel M., 2017, J. Cosmol. Astropart. Phys., 2017, 001, http://arxiv.org/abs/1611.07527
Chang T.-C., Pen U.-L., Bandura K., Peterson J. B., 2010, Nature, 466, 463
Chang T.-C., Pen U.-L., Peterson J. B., McDonald P., 2008, Phys. Rev. Lett., 100, 091303
CHIME Collaboration, 2023, ApJ, 947, 16
Cunnington S. et al., 2023, MNRAS, 518, 6262
Cunnington S., 2022, MNRAS, 512, 2408
DESI Collaboration, 2025, Phys. Rev. D, 112, 083515
Dev A. et al., 2023, MNRAS, 523, 2693
Furlanetto S., Oh S. P., Briggs F., 2006, Phys. Rept., 433, 181
Hassan S. et al., 2022, ApJ, 937, 83
He K., Zhang X., Ren S., Sun J., 2016, Deep Residual Learning for Image Recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, p. 770
Hinshaw G. et al., 2013, ApJS, 208, 19
Hitz P., Berner P., Crichton D., Hennig J., Refregier A., 2025, Fast Simulation of Cosmological Neutral Hydrogen based on the Halo Model,Journal of Cosmology and Astroparticle Physics, p. 04
Ho J., Jain A., Abbeel P., 2020, Denoising Diffusion Probabilistic Models, Advances in Neural Information Processing Systems, p. 33
Hopkins P. F., 2015, MNRAS, 450, 53
Hu W., Wang X., Wu F., Wang Y., Zhang P., Chen X., 2020, MNRAS, 493, 5854
Irfan M. O. et al., 2022, MNRAS, 509, 4923
Jones M. G., Haynes M. P., Giovanelli R., Moorman C., 2018, MNRAS, 477, 2
Kingma D. P., Salimans T., Poole B., Ho J., 2021, Variational Diffusion Models, Advances in Neural Information Processing Systems
Kovetz E. D. et al., 2017, Line-Intensity Mapping: 2017 Status Report, preprint (arXiv:1709.09066)
Li Z., Guo H., Mao Y., 2022, MNRAS, 516, 2548
Lugmayr A., Danelljan M., Romero A., Yu F., Timofte R., Van Gool L., 2022, RePaint: Inpainting using Denoising Diffusion Probabilistic Models, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition
Ma W. et al., 2025, A&A, 695, A241
Mallaby-Kay M. et al., 2021, ApJS, 255, 11
Masui K. W. et al., 2013, ApJ, 763, L20
McQuinn M., Zahn O., Zaldarriaga M., Hernquist L., Furlanetto S. R., 2006, ApJ, 653, 815
Newburgh L. B. et al., 2016, Proc. SPIE Conf. Ser. Vol. 9906, Ground-based and Airborne Telescopes VI. SPIE, Bellingham, p. 99065X

Nguyen T. et al., 2024, How DREAMS are made: Emulating Satellite Galaxy and Subhalo Populations with Diffusion Models and Point Clouds, preprint (arXiv:2409.02980), https://arxiv.org/abs/2409.02980
Obuljen A., Alonso D., Villaescusa-Navarro F., Yoon I., Jones M., 2019, MNRAS, 486, 5124
Obuljen A., Castorina E., Villaescusa-Navarro F., Viel M., 2018, J. Cosmol. Astropart. Phys., 2018, 004
Obuljen A., Simonović M., Schneider A., Feldmann R., 2023, Phys. Rev. D, 108, 083528
Ono V., Park C. F., Mudur N., Ni Y., Cuesta-Lazaro C., Villaescusa-Navarro F., 2024, ApJ, 970, 174
Padmanabhan H., Maartens R., Umeh O., Camera S., 2023, MNRAS, 524, 4778
Padmanabhan H., Refregier A., 2016, MNRAS, 464, 4008
Pandey S. et al., 2023, CHARM: Creating Haloes with Auto-Regressive Multi-stage networks, Advances in Neural Information Processing Systems. Vol. 36
Pandey S., Lanusse F., Modi C., Wandelt B. D., 2024, Teaching Dark Matter Simulations to Speak the Halo Language, preprint (arXiv:2409.11401), https://arxiv.org/abs/2409.11401
Paul S., Santos M. G., Chen Z., Wolz L., 2023, A first detection of neutral hydrogen intensity mapping on Mpc scales at z ≈ 0.32 and z ≈ 0.44. The Astrophysical Journal Letters, Vol. 946
Petit O., Thome N., Rambour C., Soler L., 2021, in U-Net Transformer: Self and Cross Attention for Medical Image Segmentation, Machine Learning in Medical Imaging. MLMI 2021. Lecture Notes in Computer Science, Vol. 12903, p. 267
Planck Collaboration VI, 2020, A&A, 641, A6
Pritchard J. R., Loeb A., 2012, Rep. Prog. Phys., 75, 086901
Riess A. G., Casertano S., Yuan W., Macri L. M., Scolnic D., 2019, ApJ, 876, 85
Rigo M., Trotta R., Viel M., 2025, MNRAS, 541, 166
Ronneberger O., Fischer P., Brox T., 2015, U-Net: Convolutional Networks for Biomedical Image Segmentation, Medical Image Computing and Computer-Assisted Intervention (MICCAI 2015), Lecture Notes in Computer Science. Vol. 9351, p. 234
Santos M. G. et al., 2015, Proc. Sci., Advancing Astrophysics with the Square Kilometre Array (AASKA14). SISSA, Trieste, PoS#019
Santos M. G. et al., 2018, MeerKAT Science: On the Pathway to the SKA, Proceedings of Science, Vol. MeerKAT2016
Seo H.-J., Dodelson S., Marriner J., Mcginnis D., Stebbins A., Stoughton C., Vallinotto A., 2010, ApJ, 721, 164
SKA Cosmology SWG, 2020, Publ. Astron. Soc. Aust., 37, e007
Sohl-Dickstein J., Weiss E. A., Maheswaranathan N., Ganguli S., 2015, Proceedings of the International Conference on Machine Learning, 37, 2256
Spinelli M., Zoldan A., De Lucia G., Xie L., Viel M., 2020, MNRAS, 493, 5434
Springel V., Hernquist L., 2003, MNRAS, 339, 289
Springel V., White S., Tormen G., Kauffmann G., 2001, MNRAS, 328, 726
Verde L., Treu T., Riess A. G., 2019, Nat. Astron., 3, 891
Villaescusa-Navarro F. et al., 2018, ApJ, 866, 135
Villaescusa-Navarro F. et al., 2022, ApJS, 259, 61
Villaescusa-Navarro F., Alonso D., Viel M., 2017, MNRAS, 466, 2736
Villaescusa-Navarro F., Bull P., Viel M., 2015, ApJ, 814, 146
Villaescusa-Navarro F., Viel M., Datta K. K., Choudhury T. R., 2014, J. Cosmol. Astropart. Phys., 2014, 050
Wadekar D., Villaescusa-Navarro F., Ho S., Perreault-Levasseur L., 2021, ApJ, 916, 42
Wang J. et al., 2021, MNRAS, 505, 3698
Weinberger R., Springel V., Pakmor R., 2020, ApJS, 248, 32
Wolz L. et al., 2022, MNRAS, 510, 3495
Wong K. C. et al., 2020, MNRAS, 498, 1420

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