The Astrophysical Journal, 991:152 (17pp), 2025 October 1 https://doi.org/10.3847/1538-4357/adfc49

The Wrath of KAN: Enabling Fast, Accurate, and Transparent Emulation of the Global 21 cm Cosmology Signal

J. Dorigo Jones[1], B. Reyes[2], D. Rapetti[1,3,4], Shah Mohammad Bahauddin[5,6], J. O. Burns[1], and D. W. Barker[1]

[1] Center for Astrophysics and Space Astronomy, Department of Astrophysical and Planetary Sciences, University of Colorado Boulder, CO 80309, USA; johnny.dorigojones@colorado.edu

[2] University of Colorado Research Computing, University of Colorado Boulder, CO 80309, USA

[3] NASA Ames Research Center, Moffett Field, CA 94035, USA

[4] Research Institute for Advanced Computer Science, Universities Space Research Association, Washington, DC 20024, USA

[5] Laboratory for Atmospheric and Space Physics, University of Colorado, Boulder, CO 80303, USA

[6] Center for Astronomy, Space Science and Astrophysics, Independent University, Bangladesh, Dhaka 1229, Bangladesh

Received 2025 July 2; revised 2025 July 30; accepted 2025 August 14; published 2025 September 24

Abstract

Based on the Kolmogorov–Arnold network (KAN), we present a novel emulator of the global 21 cm cosmology signal, 21cmKAN, that provides extremely fast training speed while achieving nearly equivalent accuracy to the most accurate emulator to date, 21cmLSTM. The combination of enhanced speed and accuracy facilitated by 21cmKAN enables rapid and highly accurate physical parameter estimation analyses of multiple 21 cm models, which is needed to fully characterize the complex feature space across models and produce robust constraints on the early Universe. Rather than using static functions to model complex relationships like traditional fully connected neural networks do, KANs learn expressive transformations that can perform significantly better for low-dimensional physical problems. 21cmKAN predicts a given signal for two well-known models in the community in 3.7 ms on average and trains about 75 times faster than 21cmLSTM, when utilizing the same typical GPU. In addition, 21cmKAN is able to achieve these speeds because of its learnable, data-driven transformations and its relatively small number of trainable parameters compared to a memory-based emulator. We show that 21cmKAN required less than 30 minutes to train and fit these simulated signals and obtain unbiased posterior distributions. We find that the transparent architecture of 21cmKAN allows us to conveniently interpret and further validate its emulation results in terms of the sensitivity of the 21 cm signal to each physical parameter. This work demonstrates the effectiveness of KANs and their ability to more quickly and accurately mimic expensive physical simulations in comparison to other types of neural networks.

Unified Astronomy Thesaurus concepts: Neural networks (1933); Astronomy software (1855); Early universe (435); Cosmology (343); Posterior distribution (1926); Nested sampling (1894)

1. Introduction

The physics of the early Universe is imprinted on the redshifted 21 cm cosmological signal emitted by neutral hydrogen (HI). Thus, measuring its brightness temperature relative to the radiation background may reveal key properties of the first stars and galaxies, and the intergalactic medium during the Cosmic Dawn and Dark Ages (see S. R. Furlanetto et al. 2006 for a review). Robust software and modeling tools are needed to achieve these constraints, in particular efficient and accurate simulations of the 21 cm signal that can be used to perform Bayesian inference and estimate the underlying physical parameters that shape its evolution. Seminumerical or semianalytical cosmological simulations of the 21 cm signal (e.g., A. Mesinger et al. 2011; A. Fialkov et al. 2014; J. Mirocha et al. 2017) offer faster alternatives to hydrodynamic codes that solve the governing partial differential equations (PDEs); however, it is computationally prohibitive to use these models directly in inference pipelines. This has motivated the development of emulators of the 21 cm signal—based on deep artificial neural networks (NNs)—that mimic the output of cosmological simulations for given sets of input physical parameters in just milliseconds. Replacing computationally expensive models with emulators is essential to efficiently sample parameter posteriors from a measured signal (e.g., E. de Lera Acedo et al. 2022; H. T. J. Bevins et al. 2024), assuming systematic effects are properly addressed (e.g., D. Rapetti et al. 2020; D. Anstey et al. 2023; A. Saxena et al. 2023).

A summary statistic of particular interest is the 1D isotropic (i.e., global; P. A. Shaver et al. 1999) 21 cm signal, which is observed at low radio frequencies of $\nu \lesssim 235$ MHz, corresponding to redshifts $z \gtrsim 5$. Existing emulators of the global 21 cm signal (A. Cohen et al. 2020; H. T. J. Bevins et al. 2021; C. H. Bye et al. 2022; D. Breitman et al. 2024; J. Dorigo Jones et al. 2024) differ in how they prioritize training speed, evaluation speed, and emulation accuracy. In this paper, we present 21cmKAN—a new publicly available global 21 cm signal emulator utilizing a Kolmogorov–Arnold network (KAN[7]; Z. Liu et al. 2025) that trains extremely quickly and achieves low emulation error comparable to the most accurate known emulator, 21cmLSTM (J. Dorigo Jones et al. 2024). It is beneficial to have an emulator that is very fast to train, while maintaining high accuracy similar to 21cmLSTM, to emulate different models of the global 21 cm signal and constrain their cosmological and astrophysical

[7] https://github.com/KindXiaoming/pykan

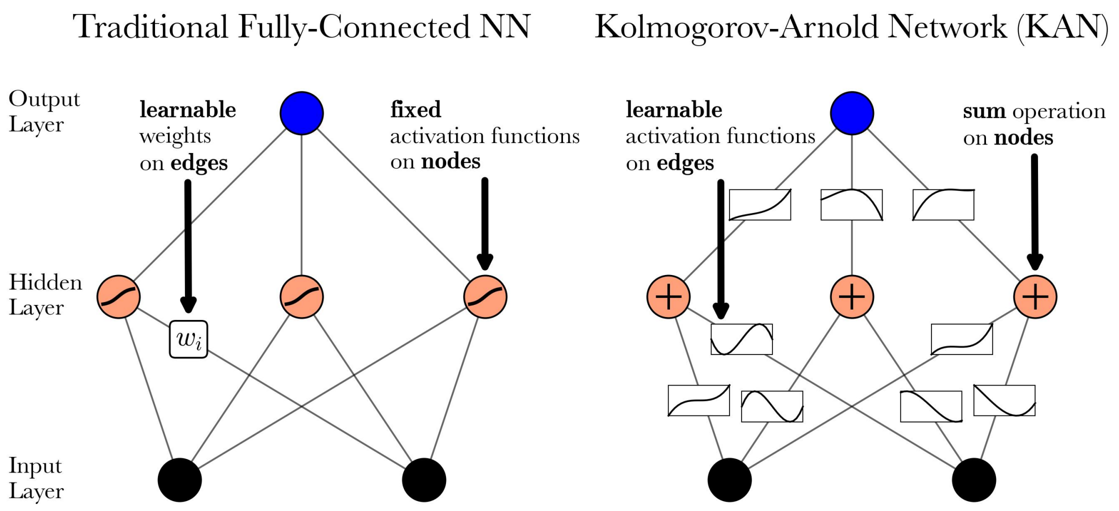
Traditional Fully-Connected NN
Kolmogorov-Arnold Network (KAN)
Output Layer
Hidden Layer
Input Layer
learnable weights on edges
fixed activation functions on nodes
w_i
learnable activation functions on edges
sum operation on nodes

Figure 1. Architecture diagrams of a single-hidden-layer traditional fully connected NN (left) and KAN (right). In the KAN, activation functions are learned and applied to parameters on the edge connections between nodes and summed at the nodes. For traditional fully connected NNs, the activations are predetermined and fixed on the nodes, the scalar weights of which are learned on the edges.

parameters. For upcoming observations, the speed–accuracy combination of 21cmKAN eliminates emulator training as a bottleneck by enabling rapid and accurate posterior sampling to comprehensively explore signal models.

KANs are a recently introduced NN architecture inspired by the Kolmogorov–Arnold representation theorem (A. N. Kolmogorov 1957; see Section 2.1). KANs learn complex relationships between input parameters and the desired output in a highly adaptive and expressive manner by stacking layers of data-driven functional transformations (see Figure 1). In contrast, traditional fully connected NNs (D. E. Rumelhart et al. 1986) transform input to output using fixed, predetermined functions with learned scalar weights. The flexibility of KANs makes them a promising alternative to traditional NNs and particularly well suited to modeling complex, lower-dimensional physical systems such as the global 21 cm signal (see also J. Cui et al. 2025; H. Shi et al. 2025). KANs provide an inherently transparent architecture with relatively few parameters that allows the user to see what it learns, which is desired in scientific applications of machine learning to further validate and explain predictions and enable transferability of knowledge between models (G. Montavon et al. 2018; A. Siemiginowska et al. 2019; H. Wang et al. 2023; F. Belfiore et al. 2025).

To demonstrate the benefits of using KANs to emulate the global 21 cm signal, this paper is organized as follows. In Section 2, we describe the architecture and training of 21cmKAN. In Section 3, we present the emulation accuracy and speed of 21cmKAN compared to other emulators and posterior constraints when using 21cmKAN to fit mock 21 cm signals, and we show examples of and interpret the learned activations. Finally, we summarize the conclusions in Section 4.

2. Methods

2.1. Neural Networks and the Architectural Benefits of KANs

At the core of an NN lies the activation function, which is inspired by the firing behavior of biological neurons. Each individual neuron—or "node" in an NN—processes the weighted sum of incoming signals by applying a nonlinear function, which determines whether and to what extent the node activates. This nonlinear response allows each node to contribute selectively to the network's representation of complex input patterns, enabling the construction of rich hierarchical features across layers. By performing a series of these simple mathematical transformations, NNs become powerful function approximators, as established by the universal approximation theorem (G. Cybenko 1989; K. Hornik et al. 1989).

A traditional fully connected NN, shown schematically in the left panel of Figure 1, is composed of layers of interconnected nodes, where each edge connection between nodes carries a scalar weight, w_i, and each node applies a fixed activation function (e.g., tanh) to a weighted sum of its inputs plus a constant term (known as bias). In this framework, the trainable parameters consist of the weights and biases, while the activation functions remain fixed throughout training. The expressive power of a traditional fully connected NN—also referred to as a multilayer perceptron (MLP)—arises from the composition of many simple nonlinear transformations. However, because these transformations are distributed across many layers and involve fixed nonlinearities, the resulting model is often difficult to interpret and lacks transparency in how it represents the underlying functional relationships (Z. C. Lipton 2018; G. Montavon et al. 2018; C. Rudin 2019), although there are methods to help explain the predictions and behavior of so-called "black box" machine learning models (e.g., K. Simonyan et al. 2014; P. W. Koh & P. Liang 2017; B. Kim et al. 2018).

In contrast to traditional NNs, KANs assign trainable functions to the edges, as illustrated in the right panel of Figure 1, while the nodes sum the transformed inputs. Each function, $\phi(x)$, is learned through a parameterized B-spline: a class of smooth, piecewise polynomial curves commonly used in approximation theory. Basis functions, $B(x)$, are weighted to form the B-splines at each edge that ultimately determine the final network output, a process that we depict in Figure 2 (see Section 3.4 for further detail). The KAN architecture is motivated by the Kolmogorov–Arnold representation theorem (A. N. Kolmogorov 1957), which states that any multivariate continuous function can be written as the finite composition of univariate continuous functions. By learning these univariate transformations and their summations, KANs can capture

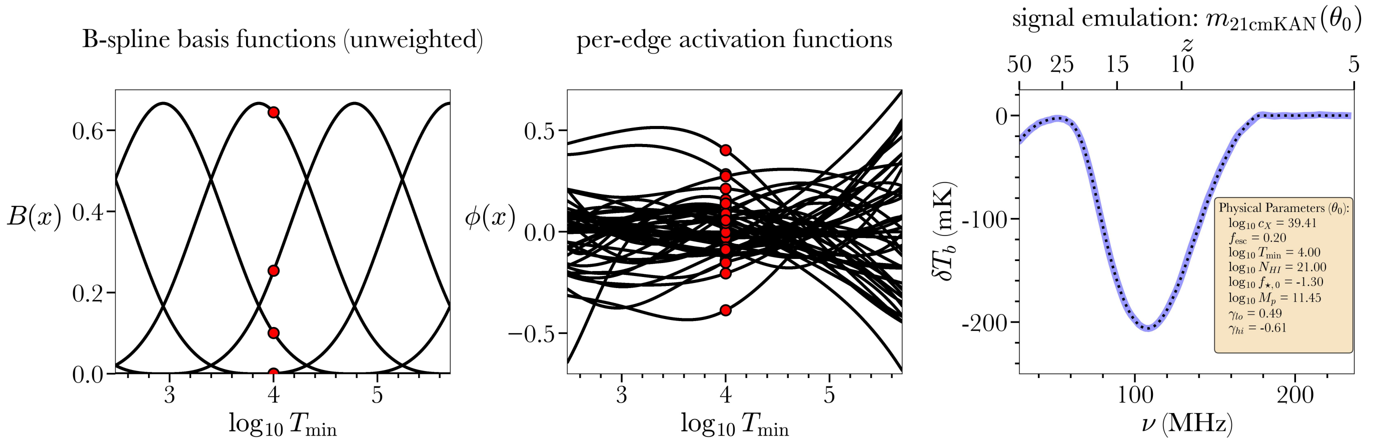

Figure 2. Example learned components and signal emulation of 21cmKAN when trained on the physical model ARES. Left: Unweighted B-spline basis functions, $B(x)$, that parameterize the trainable activation functions, $\phi(x)$. The basis functions are evaluated at a sample input (e.g., $\log_{10}T_{\min} = 4$, red markers) and linearly combined to yield the activation functions. Middle: Resulting per-edge activation functions, $\phi(x)$, each learned independently along the network edges (see Figure 1). Right: Emulated global 21 cm brightness temperature, $\delta T_b(\nu)$, as a function of frequency ν, in blue, output by 21cmKAN for input astrophysical parameters θ_0 (see Table 1 for parameter descriptions). The corresponding "true" signal from ARES is shown in dotted black. See Sections 2.2, 2.3, and 3.4 for further details on the data, training and architecture, and interpretation of 21cmKAN, respectively.

functional relationships more efficiently. As a result, training is often faster and convergence is robust, particularly in low-dimensional problems involving functional compositions, which are frequently encountered in astrophysics and cosmology. Since each transformation is explicitly represented as a learned function, the internal representations in KANs are easier to interpret and analyze when compared to traditional fully connected NNs (Z. Liu et al. 2025).

In prior work, we showed that the 21cmLSTM model provided the most accurate emulations of the global 21 cm signal to date (J. Dorigo Jones et al. 2024), owing to its inherent capability to capture the temporal evolution of the 21 cm brightness temperature across redshifts or low radio frequencies. Recurrent architectures, such as long short-term memory (LSTM) networks, are specifically designed to model sequential data. They achieve this by learning how to store and propagate information through time (or sequence) using gated mechanisms (e.g., input, forget, and output gates) that allow them to retain memories and capture temporal dependencies in data. However, LSTM NNs can be slower to train and often require more complex optimization.

In contrast to LSTM NNs, KANs capture the shape of the signal by directly learning the underlying functional transformations, without the need for memory gates or recurrent structures. This makes them well suited for low-dimensional, multivariate curve emulation where preserving the sequential structure is important. We have thus created the 21cmKAN emulator to leverage the functionality of KANs for the global 21 cm signal. As demonstrated in Section 3, while 21cmLSTM achieved the highest overall accuracy among conventional neural architectures, 21cmKAN attains comparable performance with significantly faster convergence and reduced architectural complexity.

The transparent nature of the KAN architecture, illustrated by its decomposable components in Figures 1 and 2, allows the user to directly visualize the functional transformations learned by 21cmKAN and conveniently infer feature importance (see Section 3.4). This builds further trust and reliability in the model's predictions beyond the reported emulation error and complements explainable machine learning techniques, which are benefits of interpretable architectures (Z. C. Lipton 2018; Z. Liu et al. 2025). Additionally, KANs are more geared toward transferability of knowledge between simulations or parameterizations (i.e., domain adaptation or transfer learning) because of their simplicity and flexibility compared to traditional NNs. Such functionality would help make global 21 cm signal inference pipelines more robust, generalizable, and model agnostic by unifying physical parameters across simulations and reducing the necessary sizes of training sets (e.g., S. Andrianomena & S. Hassan 2025; F. Belfiore et al. 2025; A. Wehenkel et al. 2025).

21cmKAN utilizes Ef-KAN,[8] which is an efficient implementation of KANs that significantly reduces the memory and evaluation cost, in comparison to the original implementation (see footnote 7) (Z. Liu et al. 2025), by reformulating the activation execution. 21cmKAN[9] is written in Python with a PyTorch (A. Paszke et al. 2019) backend and is publicly available (J. Dorigo Jones & B. Reyes 2025). These attributes make 21cmKAN simple to train, test, and apply to different models and data sets. In the following subsections, we describe how the 21cmKAN emulator can be trained, using popular and publicly available data sets.

2.2. Data

To train and test 21cmKAN, we utilize publicly available data sets (A. Cohen et al. 2021; J. Dorigo Jones 2024) of synthetic global 21 cm signals that allow for direct comparison of speed and accuracy against existing emulators that used the same data sets. The "21cmGEM" set was made from a large-volume seminumerical model (E. Visbal et al. 2012; A. Fialkov et al. 2013, 2014) similar to 21cmFAST (A. Mesinger et al. 2011; see A. Cohen et al. 2020), while the "ARES" set was created by a physically motivated, semianalytical model called accelerated reionization era simulations (ARES[10]; J. Mirocha et al. 2012; J. Mirocha 2014; J. Mirocha et al. 2017) that does not compute 3D volumes. See Figure 2 of J. Dorigo Jones et al. (2024) for representative subsets of each model data set. By training and

[8] https://github.com/Blealtan/efficient-kan
[9] https://github.com/jdorigojones/21cmKAN
[10] https://github.com/mirochaj/ares

Table 1
Astrophysical Parameters Varied in 21cmGEM and ARES Data Sets and Fit in Nested Sampling Analyses

Model	Parameter	Description	Range (with units)
21cmGEM	f_*	SFE	Log unif. $[10^{-4}, 5 \times 10^{-1}]$
	V_c	Minimum circular velocity of star-forming halos	Log unif. [4.2, 100] km s^{-1}
	f_X	X-ray efficiency of sources	Log unif. $[10^{-6}, 10^{3}]$
	τ	Cosmic microwave background optical depth	Uniform [0.04, 0.2]
	α	Slope of X-ray spectral energy distribution	Discrete [1, 1.5]
	$\nu_{\min}$	Low energy cut off of X-ray spectral energy distribution	Discrete [0.1, 3] keV
	$R_{\rm mfp}$	Mean free path of ionizing radiation	Uniform [10, 50] Mpc
ARES	$f_{\star,0}$	Peak SFE	Log unif. $[10^{-5}, 10^{0}]$
	$T_{\min}$	Minimum temperature of star-forming halos	Log unif. $[3 \times 10^{2}, 5 \times 10^{5}]$ K
	c_X	Normalization of X-ray luminosity–star formation rate relation	Log unif. $[10^{36}, 10^{44}]$ erg $s^{-1}(M_\odot\ yr^{-1})^{-1}$
	$\log N_{\rm H}I$	HI column density in galaxies	Uniform [18, 23]
	$M_{\rm p}$	Dark matter halo mass at $f_{\star,0}$	Log unif. $[10^{8}, 10^{15}]$ $M_\odot$
	$\gamma_{\rm lo}$	Low-mass slope of $f_\star(M_{\rm h})$	Uniform [0, 2]
	$\gamma_{\rm hi}$	High-mass slope of $f_\star(M_{\rm h})$	Uniform [-4, 0]
	$f_{\rm esc}$	Escape fraction of ionizing radiation	Uniform [0, 1]

testing on these data sets, we are able to make direct comparisons of 21cmKAN with the emulators 21cmGEM (A. Cohen et al. 2020), globalemu (H. T. J. Bevins et al. 2021), 21cmVAE (C. H. Bye et al. 2022), and 21cmLSTM (J. Dorigo Jones et al. 2024).

Although the two sets were made using different physical models with different parameters, the ARES set was made to be nearly physically equivalent to the 21cmGEM set (see J. Dorigo Jones et al. 2024). Seven astrophysical parameters were varied over wide ranges to create the 21cmGEM set, while eight parameters were varied to create the ARES set (see Table 1). The parameters in each set control the star formation efficiency (SFE) and ionizing photon production in galaxies, although via different parameterizations. The signals in the 21cmGEM set span $z = 5$–50, while the ARES signals span $z = 5.1$–49.9, all with resolution of $\Delta z = 0.1$. Each test set is the same size (1704), and the combined training+validation sets are similar in size to within 3% (27,292 for 21cmGEM and 26,552 for ARES, each split 90% for training and 10% for validation). Each set has a physical constraint on the HI fraction ($x_{\rm HI}$) at the end of the Epoch of Reionization, motivated by observations (see, e.g., X. Fan et al. 2006; I. D. McGreer et al. 2015; S. E. I. Bosman et al. 2022; X. Jin et al. 2023): the 21cmGEM set requires $x_{\rm HI} < 16\%$ at $z = 5.9$, and the ARES set requires $x_{\rm HI} < 5\%$ at $z = 5.3$.

Preprocessing or normalization of the data (i.e., physical parameter values) and labels (i.e., signal brightness temperatures, δT_b) is a common step before network training to facilitate performance. To preprocess the data, we first take the $\log_{10}$ of those parameters that are uniform only in $\log_{10}$-space: $f_\star$, V_c, and f_X for 21cmGEM and c_X, $T_{\min}$, $f_{\star,0}$, and $M_{\rm p}$ for ARES (see definitions of parameters in Table 1). The labels are flipped to train the network from high-z to low-z. Finally, a global (rather than bin-by-bin) min-max normalization is performed on each feature, x, in the data and labels:

$$\tilde{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \qquad (1)$$

2.3. Training and Architecture

For the results presented in this work, we trained and tested 21cmKAN using a single NVIDIA A100 GPU with 10 CPU cores on the Alpine heterogeneous supercomputing cluster operated by University of Colorado Boulder, Research Computing (CU RC; University of Colorado Boulder Research Computing 2023). The emulator is trained on the preprocessed training set and for each complete pass of the training set through the network (i.e., for each epoch) saves the mean squared error (MSE; Equation 2) loss function values between the true and emulated normalized signals in the training and validation sets:

$$\text{MSE} = \langle (\delta T_{\rm b},\ \text{true}(\nu) - \delta T_{\rm b},\ \text{emulated}(\nu))^2 \rangle. \qquad (2)$$

The training set errors are used during backpropagation to calculate the loss gradients and update the weights via stochastic gradient descent, which is optimized using the Adam method (D. Kingma & J. Ba 2015) with default learning rate of 10^{-3} and weight decay of 10^{-4}. The validation set is used to monitor for overfitting, ensuring the emulator can generalize to unseen signals. Finally, the test set is used to compute the reported emulation error of the saved trained instance of 21cmKAN, which loads the weights from the final epoch. As described below, we optimized the training and architecture of 21cmKAN to achieve the smallest reproducible emulation error and a validation loss that plateaus in the final epochs; as a result, we did not find it necessary to implement additional network convergence strategies such as learning rate decay, batch size scheduling, or early stopping.

We performed two separate Tune experiments (R. Liaw et al. 2018) using Optuna (T. Akiba et al. 2019), which is an automatic hyperparameter optimization software framework designed for machine learning. For these hyperparameter searches, we utilized the NVIDIA Grace Hopper (GH200) superchip through CU RC, which allowed us to train multiple models in parallel. Automatic hyperparameter searches are a common practice when designing NNs (e.g., C. H. Bye et al. 2022) and in our case do not significantly affect the resulting emulation accuracy of 21cmKAN. The hyperparameters searched were the components of the KAN architecture and the training batch size. The batch size corresponds to the number of signals whose loss values are averaged at a time to update the network weights during each epoch. Specifically, the architecture hyperparameters optimized were the number of

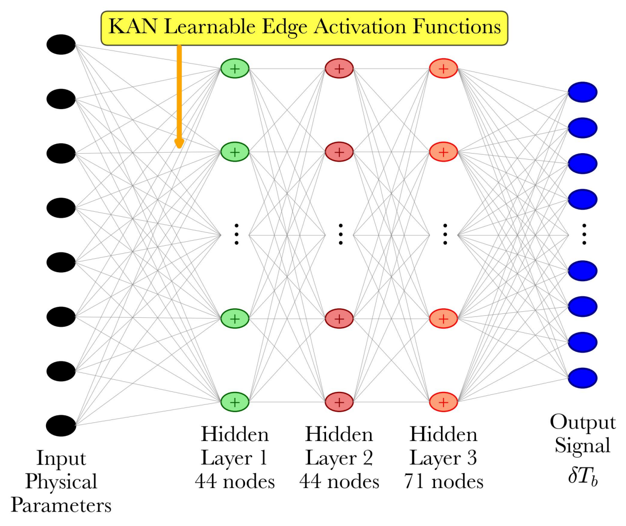
KAN Learnable Edge Activation Functions
Input Physical Parameters
Hidden Layer 1 44 nodes
Hidden Layer 2 44 nodes
Hidden Layer 3 71 nodes
Output Signal δT_b

Figure 3. Architecture diagram of 21cmKAN. As described in Section 2.1, 21cmKAN learns and applies flexible activation functions on the edges and sums them at the nodes to accurately and efficiently map the input physical parameters to the output global 21 cm signal. See Section 2.3 for a detailed description of the architecture, including optimization of the numbers of hidden layer nodes, parameterizations of the B-spline activations, and network training.

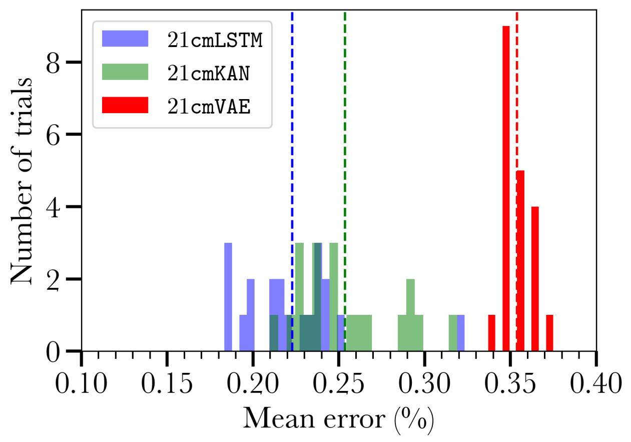

Figure 4. Histogram of the mean relative rms error (Equation 3) for 20 trials of 21cmKAN (in green) trained and tested on the 21cmGEM data set. The blue histogram is the error for 20 trials of 21cmLSTM trained and tested on the same data (J. Dorigo Jones et al. 2024). The red histogram is the approximate error for 20 trials of 21cmVAE trained and tested on the same data (adapted from Figure 6 of C. H. Bye et al. 2022). Dashed lines depict the average emulation errors.

nodes in each hidden layer, the number of grid intervals in the B-spline activation functions (i.e., the number of knots, or grid points, minus one; see Section 3.4), and the order of the individual splines. Each search trained 1,000 networks on the 21cmGEM set for 400 epochs and varied each network's hyperparameter values over the following wide ranges to find nearly optimal combinations: (1) number of nodes in hidden layer one: [10, 100], (2) number of nodes in hidden layer two: [10, 100], (3) number of nodes in hidden layer three: [10, 100], (4) grid intervals: [3, 15], (5) spline order: [3,7], and (6) batch size: [10, 1000]. We manually tested using one, two, three, or four hidden layers, rather than including the number of layers as a searched hyperparameter, and we found that three hidden layers resulted in the most accurate trained network on average.

Based on these hyperparameter searches, we determined a default architecture that achieves the lowest emulation error and has low model complexity (i.e., low number of B-spline basis functions; see Section 3.4) to facilitate fast training. We trained four different nearly optimal network architectures on the 21cmGEM set and repeated each for 20 trials to assess the stochasticity in the training algorithm (see Appendix A; Figure A1). We evaluated each trained network at the parameter values of the signals in the test set and compared the resulting emulated signals to their corresponding "true" signals, computing for each signal the relative rms error in percent across the full frequency range:

$$\text{Error} = \frac{\sqrt{\text{MSE}}}{\max(|\delta T_b(\nu)|)}, \tag{3}$$

where MSE is defined by Equation 2 and $\max(|\delta T_b(\nu)|)$ is the signal amplitude.

We found that the three-hidden-layer KAN generally prefers < 75 nodes per hidden layer, which is much less than what has been found for traditional fully connected NN emulators of the 21 cm signal (see C. H. Bye et al. 2022). This finding is consistent with the results of Z. Liu et al. (2025) and further highlights the transparency of the KAN architecture. The default architecture of 21cmKAN is three hidden layers of 44, 44, and 71 nodes, respectively, and the B-splines are parameterized by seven grid intervals and individual splines of order three (i.e., cubic splines, which is recommended by Z. Liu et al. 2025). The default batch size is 100 signals for each training epoch. We use the same default architecture when training 21cmKAN on either 21cmGEM or ARES; however, we train on the 21cmGEM set for 400 epochs and on the ARES set for 800 epochs to allow the validation loss to reach a stable value at the final epochs that is indicative of sufficient training without overfitting (see Appendix B; Figure B1; Section 3.4). The number of input layer nodes is the number of physical parameters for the data set being trained or tested on: seven for the 21cmGEM set and eight for the ARES set. The number of output layer nodes is the number of frequencies that compose each signal: 451 for the 21cmGEM set and 449 for the ARES set. Figure 3 depicts the default architecture of 21cmKAN.

3. Results

3.1. Accuracy

We report the emulation accuracy of 21cmKAN when trained and tested on the 21cmGEM set, as well as on the ARES set (see Section 2.2), to directly compare to the previous emulators 21cmGEM (A. Cohen et al. 2020), globalemu (H. T. J. Bevins et al. 2021), 21cmVAE (C. H. Bye et al. 2022), and 21cmLSTM (J. Dorigo Jones et al. 2024). We used the network architecture and training settings described in Section 2.3 and trained and tested 20 identical trials of 21cmKAN on each set, computing the relative (Equation (3)) and absolute (in mK) rms emulation errors.

When trained and tested on the 21cmGEM set, the distribution of mean relative rms error for all 20 trials is shown in Figure 4. Across the 20 trials, 21cmKAN has an average relative mean error of 0.254% ± 0.029% (corresponding to an average absolute error of 0.400 ± 0.046 mK), an average median error of 0.223% ± 0.027%, and an average maximum error of 1.09% ± 0.17%. The best trial has a mean

Table 2
Accuracy and Evaluation Speed Metrics of Global 21 cm Signal Emulators

Emulator	Mean Error (%)	Maximum Error (%)	Speed (ms)
21cmKAN	0.25	1.09	3.7
21cmLSTM	0.22	0.82	46
21cmVAE	0.35	1.84	41.4
globalemu	1.12	6.32	1.3
21cmGEM	1.59	10.55	160

Note. The information provided for each emulator are the average mean and maximum rms errors across the full frequency range of ≈1700 21cmGEM test set signals (see Section 3.1) and the average per signal evaluation speed (see Section 3.2). The per signal prediction speed quoted for 21cmKAN is the average speed from nested sampling analyses (see Section 3.3.2, Table 3), using the computational resources stated in Section 2.3. The emulation errors and evaluation speeds quoted for 21cmLSTM (J. Dorigo Jones et al. 2024), 21cmVAE (C. H. Bye et al. 2022), globalemu (H. T. J. Bevins et al. 2021), and 21cmGEM(A. Cohen et al. 2020) are from their respective original papers that utilized different computational resources. The evaluation speeds of 21cmLSTM and 21cmVAE were measured using a GPU, while the evaluation speed of globalemu was measured using a CPU (see the respective original papers for details).

relative error of 0.21% (corresponding to absolute error of 0.33 mK), a median error of 0.18% and a maximum error of 0.94%. Therefore, when trained and tested on the same data for the same number of trials, 21cmKAN has a 1.1× higher average error and 1.3× higher maximum error than those reported for 21cmLSTM (see Table 2), which currently is the most accurate emulator of the global 21 cm signal. Compared to 21cmVAE, 21cmKAN has a 1.4× lower average error and 1.7× lower maximum error.

When trained and tested on the physically equivalent ARES set for 20 trials, 21cmKAN has an average relative mean error of 0.858% ± 0.075% (corresponding to an average absolute error of 0.763 ± 0.061 mK), an average median error of 0.553% ± 0.046%, and an average maximum error of 5.31% ± 1.51%. The best trial has a mean relative error of 0.76% (corresponding to 0.68 mK), a median error of 0.50%, and a maximum error of 4.58%. The average emulation error of 21cmKAN is therefore ≈3× larger when trained and tested on the ARES set than on the 21cmGEM set, and the maximum emulation error is ≈5× larger. The difference in 21cmKAN's emulation accuracy when trained on the 21cmGEM set versus the ARES set is likely due to differences in model parameterizations, parameter ranges, and sampling uniformity (see Section 3.4).

3.2. Speed

Two different speeds are relevant when assessing an emulator's efficiency and usefulness for parameter estimation: the training speed and the per sample evaluation speed. We report the training speed of 21cmKAN as the average time to train one instance, and we report its evaluation speed as the average time to emulate one signal (i.e., predict δT_b for all frequencies from a set of input parameters) during nested sampling analyses, including the time for data preprocessing and signal denormalization (see Section 2.2; Equation (1)). Each reported speed of 21cmKAN was measured using the same computational resources (see number and type of CPUs and GPU in Section 2.3), which it naturally depends on.

Across 20 trials, the average time required to train 21cmKAN on the 21cmGEM set is only 10 minutes, which is very fast compared to existing emulators of the global 21 cm signal (see below). The average time required to train 21cmKAN on the ARES set (i.e., for 800 epochs rather than 400 epochs) is 19.2 minutes. When trained on the same set of 21cmGEM signals and using the same computational resources, we find that 21cmKAN trains ≈75× faster than 21cmLSTM. 21cmKAN trains significantly faster than 21cmLSTM because of its fully parallel feedforward architecture as opposed to the sequential nature of LSTM cells, and because standard backpropagation is much simpler and less memory intensive than the backpropagation through time required for recurrent NNs. We also find that 21cmKAN trains ≈3–4× faster than the current emulator with the fastest evaluation speed, globalemu (H. T. J. Bevins et al. 2021), when using the same computational resources. We trained the most recent version of globalemu with all preprocessing steps turned on, using three hidden layers of 32 nodes each. 21cmKAN converges more quickly during training because it learns expressive functional transformations that allow it to better model the global 21 cm signal shape.

Regarding the evaluation speed of 21cmKAN, we find that the average per signal evaluation time (i.e., the total time of nested sampling analyses performed in Section 3.3 divided by the total number of likelihood evaluations) is 3.7 ms. When using 21cmKAN and 21cmLSTM to perform the same nested sampling analyses with the same computational resources, we find that 21cmKAN evaluates ≈5× faster than 21cmLSTM. Because here we do not perform nested sampling analyses using other emulators, we do not directly compare with their evaluation speeds and instead report in Table 2 the speeds measured in their respective original papers that utilized different computational resources. Emulator training time is a more significant bottleneck in inference pipelines than evaluation speed, given the need to train an emulator with different parameterizations to constrain various combinations of physical parameters across multiple models. As might be expected, the order of the evaluation times for 21cmLSTM, 21cmKAN, and globalemu matches the order of the number of trainable parameters (i.e., weights and biases) in each emulator, with 21cmLSTM having the highest and globalemu the lowest; however, because of differences in network architectures and computational resources, the ratios of the emulator evaluation times do not directly match the ratios of their trainable parameters.

In summary, the extremely fast training and evaluation speed and low emulation error of 21cmKAN altogether enable rapid and highly accurate Bayesian multiparameter estimations of different parameterizations and models, which we carry out in Section 3.3. Efficient training is important for global 21 cm signal inference pipelines because emulator models with different parameterizations are needed to constrain the different cosmological and astrophysical parameters and their covariances across signal models. The speed and accuracy of 21cmKAN highlight the effectiveness of KANs in learning physical mappings between relatively low-dimensional input and smooth output spaces that are traditionally computed using expensive PDE solvers.

3.3. Posterior Emulation

3.3.1. Nested Sampling Analysis

We present results when using 21cmKAN, in place of the full astrophysical model it is trained on, in the likelihood of

Bayesian inference analyses to fit mock global 21 cm signals with added statistical noise and estimate its parameters. In doing so, we evaluate how well 21cmKAN constrains synthetic 21 cm signals. As mentioned in Section 1, because Monte Carlo sampling methods are computationally expensive, requiring 10^4–10^6 or more likelihood evaluations to explore multidimensional parameter spaces, emulators are desired to speed up or make feasible such analyses. Our analyses here assume that systematic uncertainties have been properly accounted for, such as those from the beam-weighted foreground (see, e.g., G. Bernardi et al. 2016; J. J. Hibbard et al. 2020; D. Anstey et al. 2023; J. J. Hibbard et al. 2023; P. H. Sims et al. 2023; M. Pagano et al. 2024; A. Saxena et al. 2024) and environment (see, e.g., S. Singh et al. 2018; N. S. Kern et al. 2020; N. Bassett et al. 2021; E. Shen et al. 2022; S. G. Murray et al. 2022).

We employ `MultiNest` (F. Feroz & M. P. Hobson 2008; F. Feroz et al. 2009, 2019), which is a nested sampling Bayesian parameter inference tool (J. Skilling 2004; for reviews see G. Ashton et al. 2022; J. Buchner 2023), to numerically sample posterior distributions, $P(\theta|\boldsymbol{D}, m)$, where θ denotes the parameters of the given physical model m, $\boldsymbol{D}$ denotes the mock data being fit, and π denotes the parameter prior distributions that we assume to be uniform or log-uniform. Bayes' theorem defines the posterior as:

$$P(\theta|\boldsymbol{D}, m) = \frac{\mathcal{L}(\theta)\pi(\theta)}{Z}, \tag{4}$$

where $\mathcal{L}$ is the likelihood function and Z is the Bayesian evidence, which can be used for model comparison. The log-likelihood function we sample from is multivariate and assumes a Gaussian probability density function for the residuals: $\log\mathcal{L}(\theta) \propto [\boldsymbol{D} - m(\theta)]^T \boldsymbol{C}^{-1}[\boldsymbol{D} - m(\theta)]$, where $\boldsymbol{C}$ is the (diagonal) noise covariance array containing the square of the noise estimate σ_{21}. Nested sampling methods iteratively remove prior volume regions with lower likelihood and simultaneously compute the evidence and posterior (see e.g., P. Lemos et al. 2023 for an in-depth description of nested sampling algorithms). Compared to Markov Chain Monte Carlo, nested sampling better constrains complex, multimodal posterior distributions (J. Buchner 2023), which have been found when fitting 21 cm data (e.g., H. T. J. Bevins et al. 2022; J. Dorigo Jones et al. 2023; D. Breitman et al. 2024; also see A. Saxena et al. 2024). For the `MultiNest` sampling parameter values, we use the default sampling efficiency (i.e., 0.8), evidence tolerance of 0.1, and 1200 initial "live" points (i.e., 3× the default), which we find result in consistent, converged posteriors.

We fit the same three mock 21 cm signals that were randomly selected from the 21cmGEM test set and fit using 21cmLSTM in J. Dorigo Jones et al. (2024), and we also fit the same `ARES` signal that was fit using `globalemu` in J. Dorigo Jones et al. (2023), to allow for direct comparisons to the nested sampling results from these works. The statistical noise added to each signal is Gaussian-distributed, white noise, ranging from σ_{21} = 5 mK or 10 mK (referred to as "optimistic"; see E. de Lera Acedo et al. 2022) to σ_{21} = 25 mK or 50 mK (referred to as "standard"). We note that σ_{21} = 5, 10, 25, and 50 mK correspond to integration times of about 7100, 1800, 300, and 70 hr, respectively, when assuming $\Delta\nu = 0.5$ MHz in the ideal radiometer sensitivity equation (e.g., J. Kraus 1966) and ν = 30 MHz to compute the antenna temperature from galactic synchrotron radiation (C. G. T. Haslam et al. 1982).

To fit the 21cmGEM signals, we used an instance of 21cmKAN trained on the 21cmGEM set that has test set mean rms error of 0.26% (corresponding to 0.42 mK) and maximum error of 1.27%. To fit the `ARES` signal, we used an instance of 21cmKAN trained on `ARES` that has test set mean rms error of 0.81% (corresponding to 0.72 mK) and maximum error of 6.41%. Each trained network is consistent with the respective average accuracy found in Section 3.1, and all fits used the same computational resources (see Section 2.3).

3.3.2. Posterior Results

In the top panels of Figures 5 and 6, we present sets of posterior signal realizations (shown in red), when using 21cmKAN to fit 21cmGEM and `ARES` mock 21 cm signals (shown in dark blue), respectively, with added noise levels (shown in light blue) of σ_{21} = 5, 10, and 25 mK, from left to right. The 1σ emulated posteriors are shown in red, which we define as the 68% of samples with the lowest relative rms error with respect to the true signal (Equation 3), although we note that the distribution of rms errors is not exactly Gaussian. The bottom panels show the residuals between the true signals and the 1σ posteriors (red), the mean posterior (solid black), and the emulator realization of the true signal (i.e., $m_{\texttt{21cmKAN}}(\theta_0)$; dashed black). Table 3 summarizes each fit, and the marginalized 1D and 2D posterior distributions for three of the fits are presented in Appendix C along with the physical parameter values, θ_0, used to generate each mock signal. We note that the fits using 21cmKAN to constrain the 21cmGEM signals completed in 3–8 minutes, and the fits using 21cmKAN to constrain the `ARES` signal completed in 5–18 minutes, depending on the noise level σ_{21} (see Table 3).

As expected, the fits obtained using 21cmKAN consistently improve for decreasing σ_{21}, following the increase in constraining power for longer assumed integration times, and this does not depend on the particular mock signal being fit. As seen visually in the bottom panels of Figures 5 and 6 and quantitatively in Table 3, we find that the mean and 1σ rms errors of the posteriors are consistently ≈3× less than σ_{21} and approach the emulator error (see Section 3.3.1) as σ_{21} decreases, which facilitates unbiased parameter posterior distributions (see below). For the 21cmGEM signal fits, the mean relative rms error (Equation 3) between all the emulated posteriors and the true signal is 0.91% (corresponding to 1.63 mK absolute error) for the fit done with σ_{21} = 5 mK, 1.42% (2.75 mK) for σ_{21} = 10 mK, and 4.15% (6.94 mK) for σ_{21} = 25 mK. For the `ARES` signal fits, the posterior mean relative rms error is 0.79% (corresponding to 1.62 mK absolute error) for σ_{21} = 5 mK, 1.38% (2.85 mK) for σ_{21} = 10 mK, and 2.93% (6.05 mK) for σ_{21} = 25 mK. We note that the `ARES` signal fits required ≈3× more likelihood evaluations to converge compared to the 21cmGEM signal fits with the same σ_{21} (see Table 3), which reflects the ≈3× higher average emulation error for 21cmKAN when trained on the `ARES` set than when trained on the 21cmGEM set (Sections 3.1 and 3.4).

In general, the 21cmGEM signal fits using 21cmKAN are extremely similar to the same fits performed using 21cmLSTM presented in J. Dorigo Jones et al. (2024). Furthermore, as described below, the `ARES` signal fits using 21cmKAN are slightly better than the same fits performed

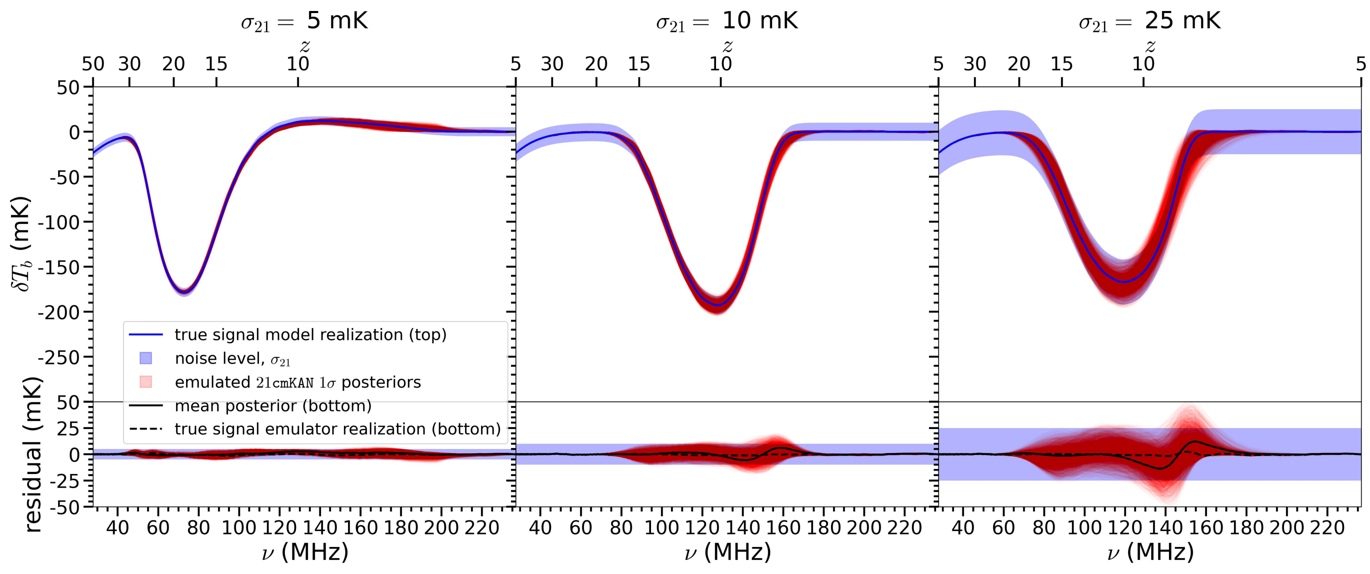

Figure 5. Top: Signal realizations of the 1σ posteriors (red, see Section 3.3) obtained from nested sampling analyses using 21cmKAN to fit three global 21 cm signals (dark blue) randomly selected from the 21cmGEM test set (see J. Dorigo Jones et al. 2024) with added noise (light blue bands) of 5 mK (left), 10 mK (middle), and 25 mK (right). Bottom: Residuals between the corresponding true signal and each 21cmKAN1σ posterior (red, see Table 3), the mean posterior (solid black), and the signal emulation (dashed black).

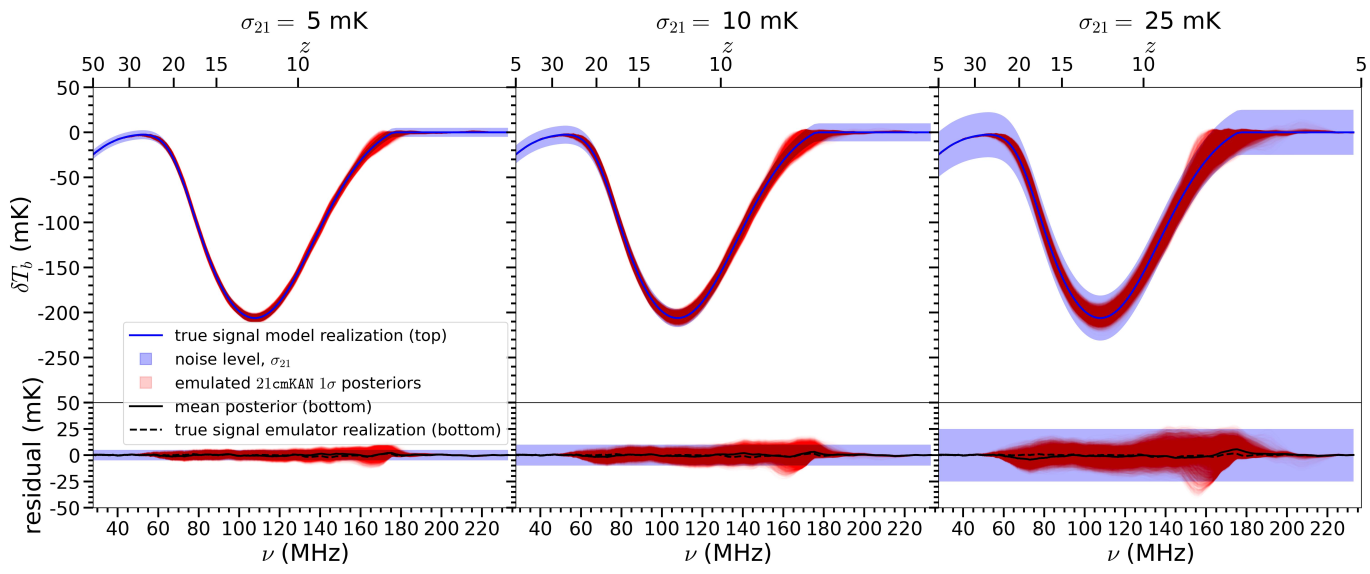

Figure 6. Top: Same as Figure 5 but when using 21cmKAN to fit the same ARES signal that was fit in J. Dorigo Jones et al. (2023).

using globalemu presented in J. Dorigo Jones et al. (2023). These results are expected, given that the average emulation error of 21cmKAN is only slightly higher than that of 21cmLSTM and is significantly lower than that of globalemu (see Section 3.1; Table 2). For the 21cmGEM signal fit with $\sigma_{21} = 25$ mK (Figure C1), 21cmKAN obtains well-constrained and unbiased posteriors (i.e., captures the true parameter values to within 2σ) for f_*, V_c, and τ, and for the $\sigma_{21} = 5$ mK fit (Figure C2) the constraints become unbiased for f_X and $\nu_{\min}$. The same fits performed using 21cmLSTM in J. Dorigo Jones et al. (2024) obtained nearly equivalent final evidence and 1σ posterior signal residuals (see their Table 3), and very similar but slightly better posterior distributions (see their Figures B1 and B2). For the ARES signal fit with $\sigma_{21} = 50$ mK (Figure C3), 21cmKAN obtains unbiased posteriors for c_X, $f_{\rm esc}$, $T_{\min}$, and $f_{\star,0}$, capturing their true values within 2σ. The posteriors obtained using globalemu in J. Dorigo Jones et al. (2023) were similar but slightly less centered on the true parameter values and fewer 2D posteriors captured the true values within 2σ (see their Figure 6), which is reflected in the final evidence being slightly worse (see their Table 2). These mock 21 cm posterior constraints support studies that have found it necessary to combine complementary summary statistics or data sets to break degeneracies between some physical parameters (e.g., Y. Qin et al. 2020; A. Chatterjee et al. 2021; J. Dorigo Jones et al. 2023; D. Breitman et al. 2024).

To summarize, 21cmKAN obtains unbiased physical parameter posteriors when used in Bayesian inference analyses to fit mock global 21 cm signals from two different simulations

Table 3
Summary of Nested Sampling Analyses

Signal Fit	σ_{21} (mK)	$n_{\rm eval}$	$f_{\rm acc}$	$\log Z$	$t_{\rm eval}$ (ms)	$\sigma_{\rm post}$ (mK)
21CMGEM	5	136,397	0.190	-256.6	3.7	1.7
	10	64,749	0.338	-253.4	4.2	2.9
	25	47,711	0.363	-250.9	4.4	7.8
ARES	5	335,760	0.090	-258.6	3.2	1.9
	10	208,161	0.126	-255.0	3.4	3.2
	25	132,493	0.157	-250.8	3.5	5.7
	50	82,675	0.208	-248.1	3.8	⋯

Note. The information provided for each fit includes the noise level of the mock 21 cm signal (σ_{21}), the total number of likelihood evaluations ($n_{\rm eval}$), the final acceptance rate ($f_{\rm acc}$), the final evidence ($\log Z$), the time per likelihood evaluation ($t_{\rm eval}$), and the 1σ rms error of all posterior samples ($\sigma_{\rm post}$). Each $\log Z$ has error of ± 0.1. The nested sampling analysis and results are described in Section 3.3. The 1σ posterior signal realizations and residuals with respect to the true 21CMGEM and ARES signals are shown in Figures 5 and 6, respectively. The full posterior distributions for the 21CMGEM fits with $\sigma_{21} = 25$ mK and $\sigma_{21} = 5$ mK are shown in Figures C1 and C2, respectively, and for the ARES fit with $\sigma_{21} = 50$ mK in Figure C3.

—21CMGEM and ARES—with standard or extremely optimistic noise levels. 21CMKAN has signal and posterior emulation accuracy on par with the most accurate emulator, 21CMLSTM, and requires only 15–38 minutes for the entire process from beginning of training to completing the multidimensional nested sampling analyses presented here, when using a typical GPU (see resources in Section 2.3). The combination of exceptionally high speed and accuracy makes 21CMKAN valuable for constraining real data of the 21 cm signal because different physical model parameterizations will need to be tested.

3.4. Visualizing and Interpreting Learned 21CMKAN Activation Functions when Trained on Different Models

We now describe the 21CMKAN architecture in more detail and present visualizations of the learned activation functions, which help interpret the emulator's behavior and the sensitivity of the 21 cm signal to different physical parameters. For a general overview of KAN and how it compares to other types of NNs, see Section 2.1 and Figures 1 and 2. By simply observing the components of the trained emulator, we demonstrate that 21CMKAN conveniently provides an intuitive understanding of its predictions that is complementary to the information gained from more rigorous techniques such as saliency mapping, feature importance or attribution, and activation maximization (Z. C. Lipton 2018; G. Montavon et al. 2018). The interpretability of 21CMKAN further validates its performance and motivates transferability to other models and tasks beyond the emulation results presented in Sections 3.1 and 3.3 (e.g., F. Belfiore et al. 2025; Z. Liu et al. 2025).

Let a given hidden layer have input dimensionality p and output dimensionality q, such that the default 21CMKAN trained on the 21CMGEM set has $p = 7$ and $q = 44$ for the first hidden layer. Each edge connecting input node p to output node q learns a univariate activation function, $\phi_{q,p}(x_p)$, and applies it to the corresponding input parameter x_p, such that there are pq different activations per layer. Each per-edge function, hereafter denoted $\phi(x)$, is a B-spline curve (C. de Boor 1980) that is the weighted sum of local basis functions $B_i(x)$ with learnable coefficients c_i (see Z. Liu et al. 2025):

$$\phi(x) \approx \sum c_i B_i(x). \tag{5}$$

For a B-spline parameterized by G grid intervals (i.e., $G + 1$ grid points) and individual splines of order k, there are $G + k$ basis functions, which sum to one at each x_p, $\sum B_i(x_p) = 1$, making KANs a form of nonlinear dimensionality reduction or feature extraction (see also multivariate adaptive regression splines; J. H. Friedman 1991; D. Denison et al. 1998; D. Francom & B. Sansó 2020). At the start of training, the basis function coefficients and thus B-splines are zero, and the splines become nonlinear and grow in magnitude as the c_i are learned. The input to each node in a subsequent layer is the sum of all the incoming postactivations across p, $\sum\phi(x)$ (see also multiplicative KANs; Z. Liu et al. 2024). In summary, KANs learn and apply flexible nonlinear activation functions on the edges and sum them at the nodes to map input to output, whereas traditional MLPs learn scalar weights on the edges and apply fixed, predetermined activations at the nodes.

To visualize the KAN activations, we present in Figures 7 and 8 the basis functions ($B(x)$, bottom panels), per-edge B-spline activations ($\phi(x)$, middle panels), and their node-wise sums ($\sum\phi(x)$, top panels) in the first hidden layer for each physical parameter that were learned by 21CMKAN when trained on the 21CMGEM and ARES data sets, respectively. The first hidden layer is more interpretable or explainable than subsequent layers (G. Montavon et al. 2018; Z. C. Lipton 2018) because the learned activations depict how each input physical parameter contributes to the output. To generate these plots, we evaluated the same trials of 21CMKAN used for posterior emulation in Section 3.3 at the parameter values of the respective training set (see ranges in Table 1).

The KAN activation functions for almost all physical parameters are nonlinear and vary significantly across the parameters and edges, indicating that 21CMKAN learns distinct contributions of each parameter to the output signal δT_b. The learned activations and basis function coefficients can change between different trials of 21CMKAN, which is expected because NNs often converge to different local minima, but this stochasticity may preclude direct connections between the activation functions and relevant physical equations. As seen in the middle panels of Figures 7 and 8, larger magnitude activations correlate with larger basis function coefficients (yellow), whereas smaller activations generally correspond to smaller c_i (purple), as expected from Equation 5. The peaks of the activations typically coincide with the local basis function that has the largest learned coefficient, highlighting physical parameter values with greater impact on the output 21 cm signal. 21CMKAN learns near-zero coefficients for the first three basis functions, and so the first three $B(x)$ are zero across all x_p and parameters.

We find that physical parameters with larger magnitude per-edge activations learned by 21CMKAN correspond to those with greater influence on the 21 cm signal prediction. This relationship between activation magnitude and feature importance was also found in Z. Liu et al. (2025) as an example of KANs being inherently interpretable and complementary to post-hoc explainable machine learning approaches (see their Section 4.3) and could be utilized for domain adaptation. Additionally, activations with greater variation along

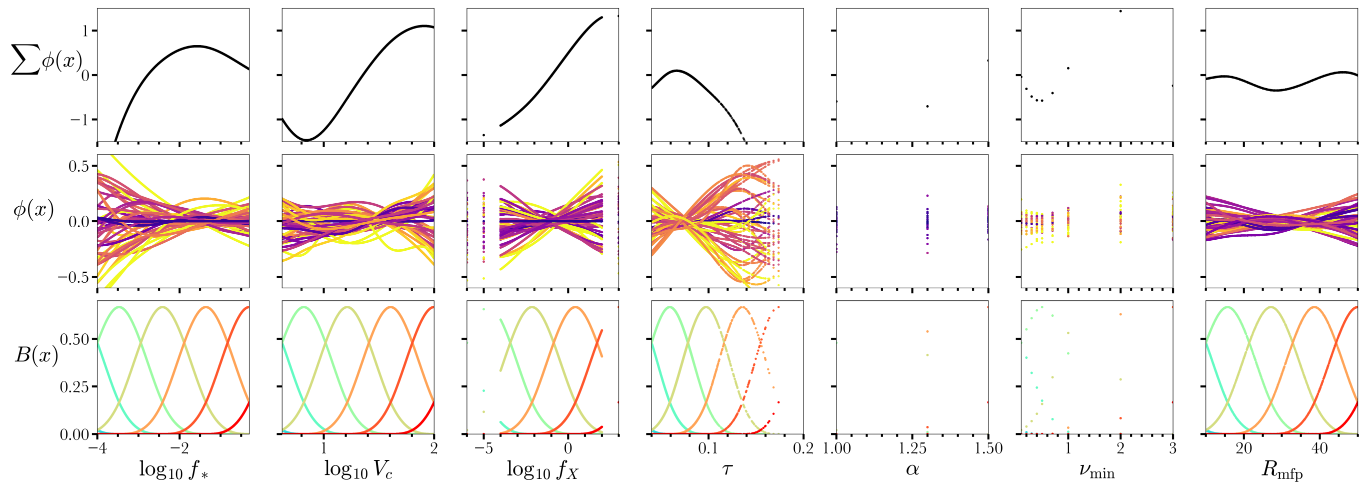

Figure 7. Univariate, nonlinear activation functions in the first hidden layer for each input physical parameter, learned by a trial of 21cmKAN trained on the 21cmGEM set and evaluated at the parameter values of the 24,562 training set signals (see Section 3.4). Top: Aggregated activation function output that is the input for the second hidden layer nodes. Middle: B-spline activation functions for all 44 edges, with color depicting the largest basis function coefficient value for each (see Equation (5)). Yellow activations have larger coefficients of their constituent basis functions, while purple activations have smaller, or less influential, basis functions. Bottom: Local B-spline basis functions.

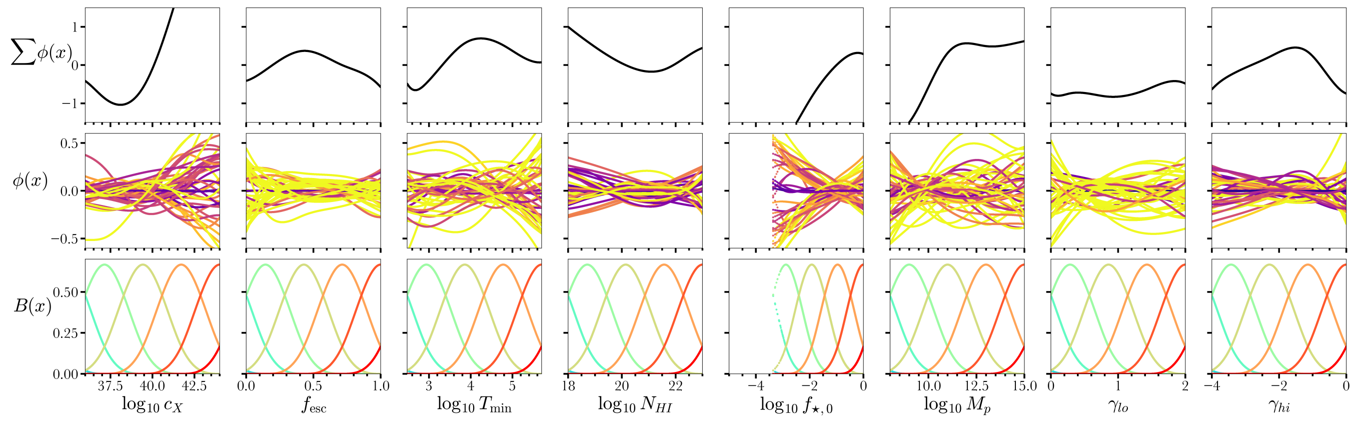

Figure 8. Same as Figure 7 but for 21cmKAN trained on the ARES set and evaluated at the parameter values of the 23,896 training set signals. The y-axis ranges are the same as in Figure 7 to show that the per-edge activations are generally higher magnitude when trained on the ARES set.

parameter values mean that small changes in the parameter meaningfully affect the output signal and imply the signal is strongly sensitive to that parameter, which is also indicated by more constrained posteriors (see Section 3.3.2). The learned activations across all 20 trials for 21cmGEM parameters $f_\star$, V_c, and f_X and ARES parameters c_X and $T_{\rm min}$ are larger in magnitude and present larger variations, and their posterior distributions are correspondingly well constrained in Figures C2 and C3, respectively. In contrast, the 21cmGEM parameter $R_{\rm mfp}$ as well as the ARES parameters $\log N_{\rm H}I$ and $\gamma_{\rm hi}$ have activations with less variation and magnitude, and they also have the least constrained posteriors. These results are consistent with previous analyses (see, e.g., Figure 4 of C. H. Bye et al. 2022) that find $R_{\rm mfp}$ to have the smallest impact on the 21cmGEM signals and $f_\star$, V_c, and f_X to have the largest impact.

Figures 7 and 8 also show that the sampling in the training sets is less uniform for certain parameters than others. For 21cmGEM, α and $\nu_{\rm min}$ are sparsely sampled, and the lower end of f_X and upper end of τ are less uniformly sampled, while the other parameters are uniformly sampled. For ARES, the lower end of $f_{\star,0}$ is less uniformly sampled, while all of the other parameters are uniformly sampled. As explained in Section 2 of A. Cohen et al. (2020), in creating the 21cmGEM set, a few values were chosen for α and $\nu_{\rm min}$ to construct the spectral energy distribution (SED) power-laws of high-z X-ray sources. The ARES model treats X-ray SEDs in a fundamentally different manner than the 21cmGEM model, as explained in J. Mirocha (2014) and J. Mirocha et al. (2017), by explicitly solving the cosmological radiative transfer equation rather than scaling the X-ray heating efficiency by f_X. The sparse sampling of α and $\nu_{\rm min}$ hinders the variation in the 21cmGEM training set and restricts 21cmKAN to learn from fewer parameter values, which effectively simplifies the mapping. Combined with the fact that the 21cmGEM set contains one less physical parameter than the ARES set, we infer that the nonuniform sampling of α and $\nu_{\rm min}$ contributes to the $\approx 3\times$ lower average emulation error of 21cmKAN and $2\times$ faster time for it to reach a stable validation loss when trained on the 21cmGEM set than the ARES set (see Section 3.1; Figure B1). This interpretation is consistent with the principal component analysis decompositions performed in J. Dorigo Jones et al. (2024), which identified lower statistical variation among the signals in the 21cmGEM set than those in the ARES set. We leave for future work a detailed comparison of the 21cmGEM and ARES models and a potential domain adaptation between them enabled by 21cmKAN.

4. Conclusions

In this paper, we presented 21cmKAN, which is a publicly available (see footnote 9) KAN-based emulator of the global 21 cm signal. KANs (see footnote 7) (Z. Liu et al. 2025) provide a fundamental shift in the architecture of fully connected NNs, by explicitly learning the nonlinear transformations, or activation functions, that best model complex relationships in data (Figures 1 and 2). The expressivity of KANs makes them an improvement in accuracy and speed over traditional fully connected NNs, which use fixed activation functions, when modeling certain structured, lower-dimensional functions or PDEs often found in science (e.g., K. Shukla et al. 2024; J. Cui et al. 2025; S. Cui et al. 2025; H. Shi et al. 2025). 21cmKAN has low emulation error similar to the most accurate existing emulator of the global 21 cm signal, 21cmLSTM (J. Dorigo Jones et al. 2024) and trains ≈75 × faster than 21cmLSTM (Figure 4; Table 2; Section 3.1; Section 3.2).

The combination of low emulation error and extremely fast training and evaluation speed of 21cmKAN enables rapid and highly accurate Bayesian inferences of multiple physical models. Emulators will need to be trained for various cosmological and astrophysical models of the global 21 cm signal to constrain their parameters. Posterior distributions generally change with the addition or removal of parameters or data sets based on their covariances and constraints (e.g., Y. Qin et al. 2020; A. Chatterjee et al. 2021; J. Dorigo Jones et al. 2023; D. Breitman et al. 2024), and so it will be important to explore different combinations of parameters to robustly fit and fully exploit measurements of the signal. The efficiency of 21cmKAN eliminates emulator training as a bottleneck for such comprehensive studies using inference pipelines. We trained and tested 21cmKAN on two popular physical models of the global 21 cm signal and employed 21cmKAN in Bayesian parameter estimation analyses to fit synthetic data with varying levels of added observational noise (Section 3.3). 21cmKAN obtained signal fits with rms errors ≈3× less than the assumed noise (Figures 5, 6; Table 3) as well as unbiased posterior distributions for multiple well-constrained parameters in the two models (Figures C2, C3), which are consistent with the results when using 21cmLSTM to fit the same signals (J. Dorigo Jones et al. 2024). 21cmKAN evaluated each signal prediction in just 3.7 ms on average, when utilizing a typical A100 GPU, which enabled it to perform end-to-end training and posterior emulation altogether in under 30 minutes.

We found that the transparent nature of the 21cmKAN architecture allows the user to conveniently infer the influence of different physical parameters on the global 21 cm signal by simply inspecting the learned activation functions (Section 3.4; Figures 7, 8). Physical parameters with larger magnitude and highly variable activation functions also have tightly constrained posterior distributions, which altogether imply high sensitivity of the 21 cm signal to these parameters. In contrast, the physical parameters with lower magnitude or less variable learned activations are also those with poorly constrained posterior distributions, both of which imply that the 21 cm signal is weakly sensitive to these parameters. Such intuitive understanding of the emulator's innerworkings and predictions complement standard explainable machine learning tests and build further trust in the model (Z. C. Lipton 2018; G. Montavon et al. 2018; Z. Liu et al. 2025). Beyond this, the simple and flexible architecture of 21cmKAN could enable mapping between different models of the global 21 cm signal to unify their physical assumptions, and thus improve the robustness and generalization of signal modeling in inference pipelines.

21cmKAN demonstrates the effectiveness and expressive nature of learnable neural activations for modeling the global 21 cm signal. While this work focused on a specific set of models and parameterizations, 21cmKAN can be readily trained on other models or adapted to approximate a range of functions that may benefit from its unique architecture. More broadly, KANs can complement existing recurrent or convolutional NNs (M. Cheon & C. Mun 2024; R. Genet & H. Inzirillo 2024) and may prove to be valuable for other astrophysical or cosmological applications where deep learning has already shown success (e.g., H. Liu et al. 2019; S. M. Bahauddin & M. P. Rast 2021; D. Piras & A. Spurio Mancini 2023; D. Breitman et al. 2024; I. Kamai & H. B. Perets 2025).

Acknowledgments

We thank the anonymous reviewer and Jordan Mirocha for their valuable feedback. This work utilized the Alpine high performance computing (HPC) cluster resource at the University of Colorado Boulder. Alpine is jointly funded by the University of Colorado Boulder, the University of Colorado Anschutz, Colorado State University, and the National Science Foundation (award #2201538 and #2322260). We acknowledge support by NASA APRA grant award 80NSSC23K0013, a subcontract from UC Berkeley (NASA award 80MSFC23CA015) to the University of Colorado (subcontract #00011385) for science investigations involving the LuSEE-Night lunar far side mission, and NASA award 80NSSC22K1264 to support radio astrophysics from the Moon.

Software: NUMPY (C. R. Harris et al. 2020), MATPLOTLIB (J. D. Hunter 2007), SCIPY (P. Virtanen et al. 2020), PYTORCH (A. Paszke et al. 2019), KAN (see footnote 7) (Z. Liu et al. 2025), Ef-KAN (see footnote 8), JUPYTER(T. Kluyver et al. 2016), MultiNest (F. Feroz et al. 2009).

Appendix A
Emulation Error for Different 21cmKAN Architectures

In Figure A1, we present the emulation error of 21cmKAN when trained and tested on the 21cmGEM set using four different architectures, each with 20 trials performed (see caption for details). For the green, red, blue, and purple histograms, the average training time using each architecture is 10.1, 13.6, 20.2, and 8.0 minutes, respectively. We decided to use the green architecture as the default for 21cmKAN because it results in the lowest average emulation error while still maintaining fast training time because of its relatively large batch size and low number of B-spline basis functions (see Section 3.4).

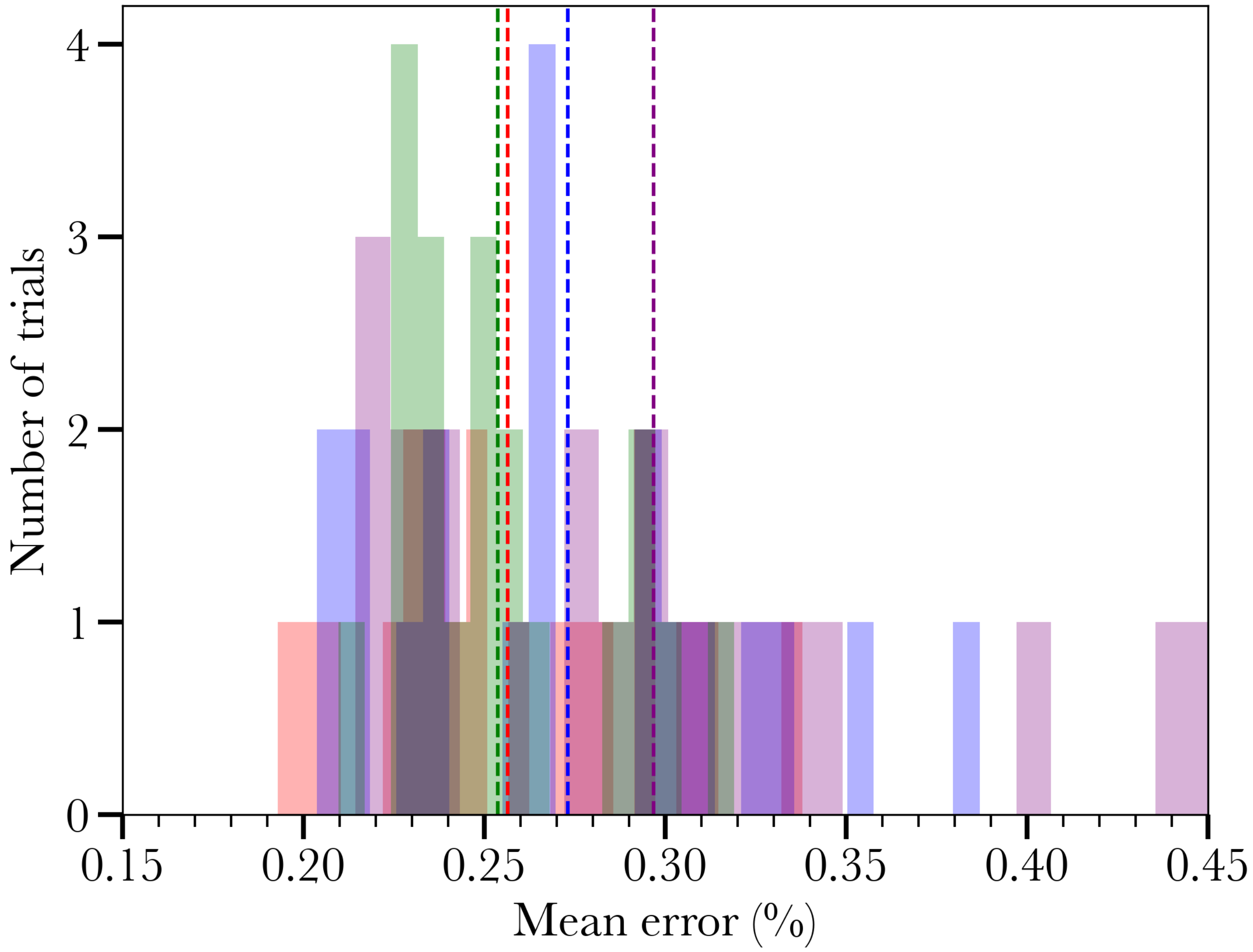

Figure A1. Emulation error (Equation 3) of 21cmKAN using four different architectures when trained and tested on the 21cmGEM set for 20 trials each. The average error for each architecture is indicated by the respective dashed line. The architecture hyperparameters that are varied are the numbers of nodes (N) per hidden layer (i.e., [N_1, N_2, N_3]), the B-spline grid size (G), the spline order (k), and the batch size (b). The green histogram used the default architecture (see Section 2.3) of [N_1, N_2, N_3] = [44, 44, 71], G = 7, k = 3, and b = 100. The red histogram used [N_1, N_2, N_3] = [20, 45, 74], G = 11, k = 6, b = 119, the blue histogram [N_1, N_2, N_3] = [55, 63, 71], G = 11, k = 7, b = 84, and the purple histogram [N_1, N_2, N_3] = [45, 95, 100], G = 11, k = 5, b = 195.

Appendix B
Loss Curves for Validation and Training Sets

Figure B1 shows the distribution of MSE loss (Equation 2) for the training and validation sets for 20 identical trials of the default 21cmKAN trained on the 21cmGEM data and also when trained on the ARES data (see Section 2.3). The mean validation losses (shown by silver curves) reach stable values at the final training epochs, which indicates sufficient training and that the network is still able to generalize to unseen signals and is not overfitting the training set. We note that we trained 21cmKAN on each set for more epochs and the average test set emulation errors did not significantly improve, which is also seen by the fact that the minimum validation loss does not occur at the end of the training, on average (vertical lines).

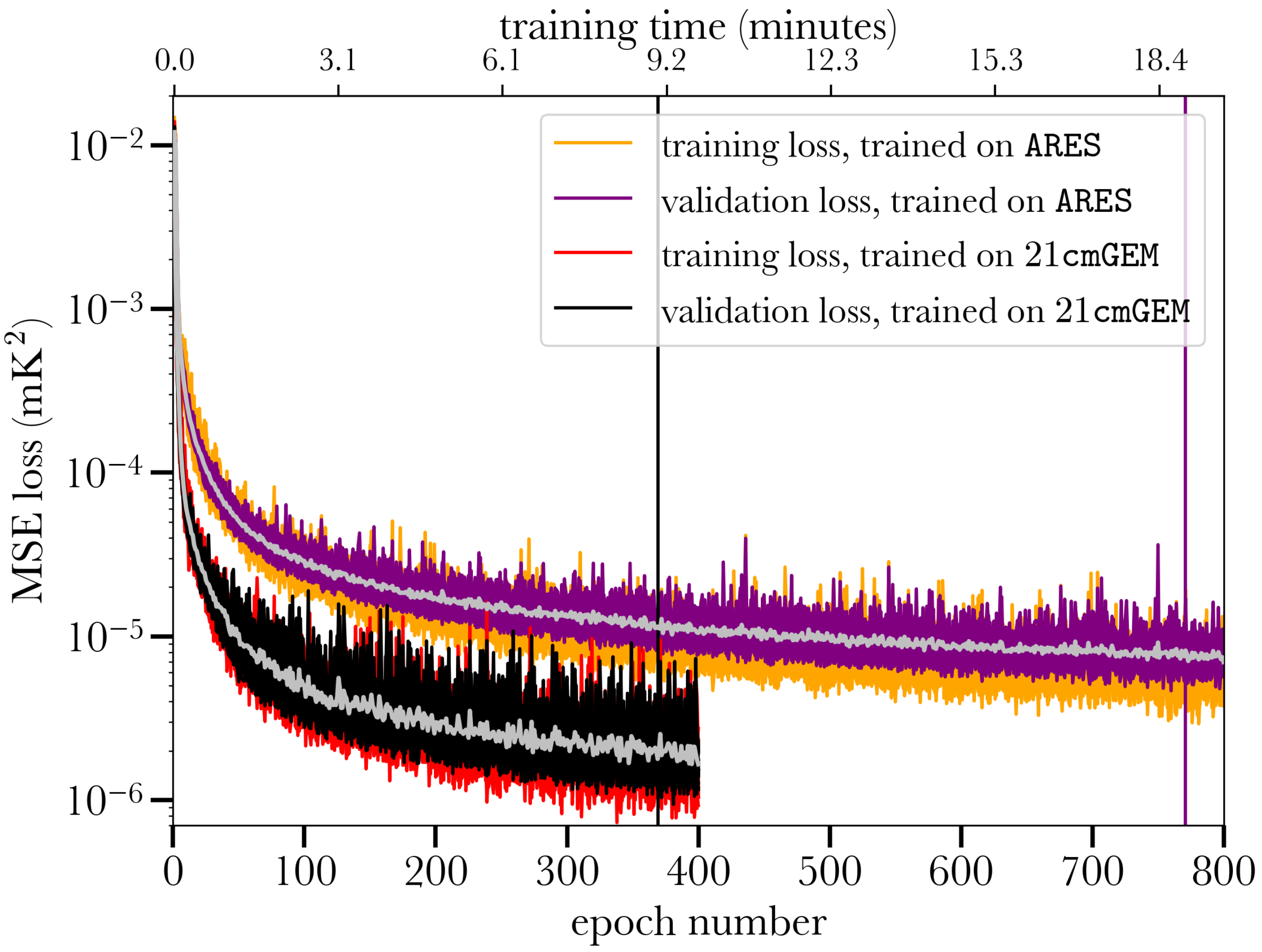

Figure B1. Loss vs. training epoch number for validation and training sets for 20 trials of 21cmKAN trained on the 21cmGEM (black and red, respectively) and ARES (purple and orange, respectively) sets (see Section 2). The top axis shows the approximate training time at each epoch. The vertical lines indicate the average epochs of the minimum validation losses, which are 369 for 21cmGEM and 771 for ARES. Note that 21cmKAN must be trained for 2 × longer on the ARES set to reach a stable mean validation loss (silver curves; see Section 3.4).

Appendix C
Posterior Distributions from Fitting Mock Global 21 cm Signals

We present the full 1D and 2D marginalized posterior distributions obtained when using 21cmKAN in nested sampling analyses to fit three different mock global 21 cm signals generated by two physical models, 21cmGEM and ARES, with different added noise levels, σ_{21}. See Table 1 for descriptions of the astrophysical parameters being fit, and see Section 3.3 and Table 3 for descriptions of the analyses and results and comparison to other works that used different emulators to fit the same signals with the same σ_{21}. Figures C1 and C2 show the posteriors when fitting two different signals from the 21cmGEM test set with σ_{21} = 25 mK and σ_{21} = 5 mK, respectively. Figure C3 shows the posteriors when fitting the ARES signal with σ_{21} = 50 mK. The 1σ posterior signal realizations for the 21cmGEM and ARES signal fits with σ_{21} = 5, 10, and 25 mK are shown in Figures 5 and 6, respectively.

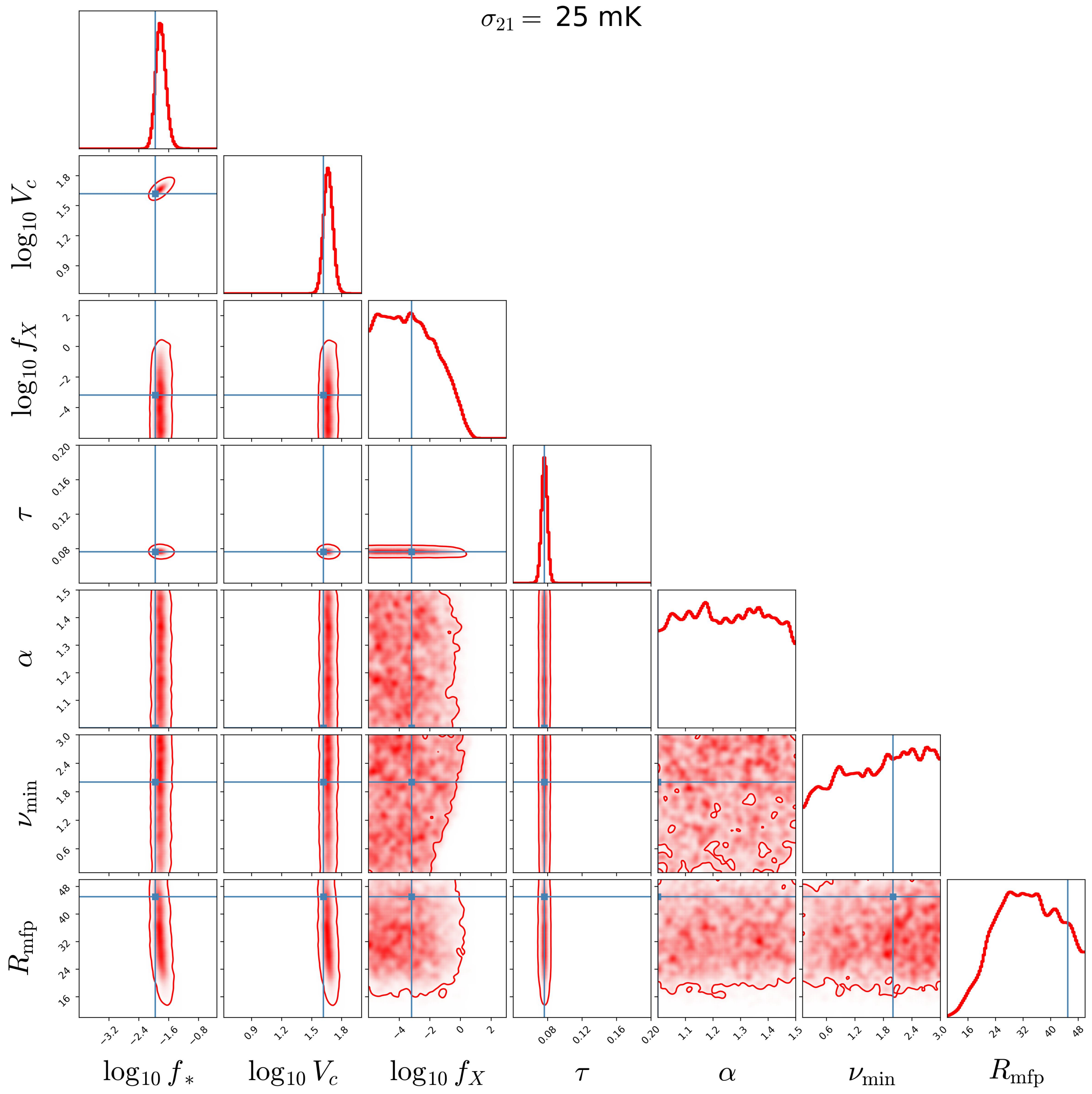

Figure C1. Marginalized 1D and 2D posteriors for seven astrophysical parameters obtained when using 21cmKAN to fit a mock global 21 cm signal from the 21cmGEM test set with the standard observational noise of $\sigma_{21} = 25$ mK (see the right panel of Figure 5; Section 3.3). These parameters control the SFE, emission of UV and X-ray photons, and cosmic microwave background optical depth (see Table 1). Blue vertical and horizontal lines indicate the parameter values used to generate the mock signal being fit: $\theta_0 = (f_*, V_c, f_X, \tau, \alpha, \nu_{\rm min}, R_{\rm mfp}) = (1.1020955 \times 10^{-2}, 41.533981, 6.4696016 \times 10^{-4}, 7.6195940 \times 10^{-2}, 1.0, 2.0, 45.0)$. Contour lines in the 2D posteriors represent the 95% confidence levels, and sample density colormaps are shown. Axis ranges are the full training set prior ranges given in Table 1.

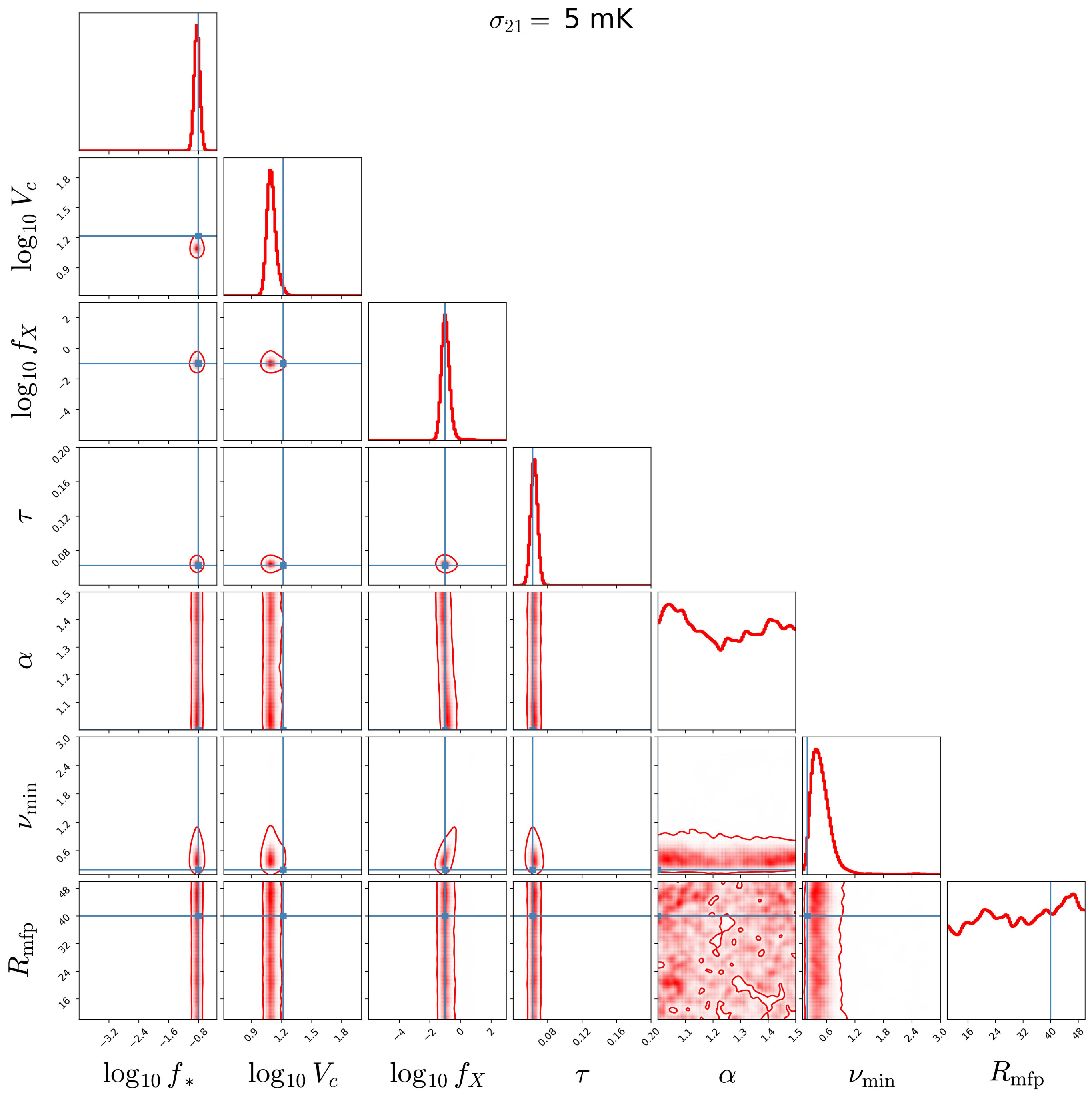

Figure C2. Same as Figure C1 but when using 21cmKAN to fit a different 21cmGEM signal with $\theta_0 = (f_*, V_c, f_X, \tau, \alpha, \nu_{\rm min}, R_{\rm mfp})$=(1.5811388 × 10^{-1}, 16.5, 0.1, 6.260315 × 10^{-2}, 1.0, 0.2, 40.0) and the optimistic noise level of $\sigma_{21} = 5$ mK added (see the left panel of Figure 5).

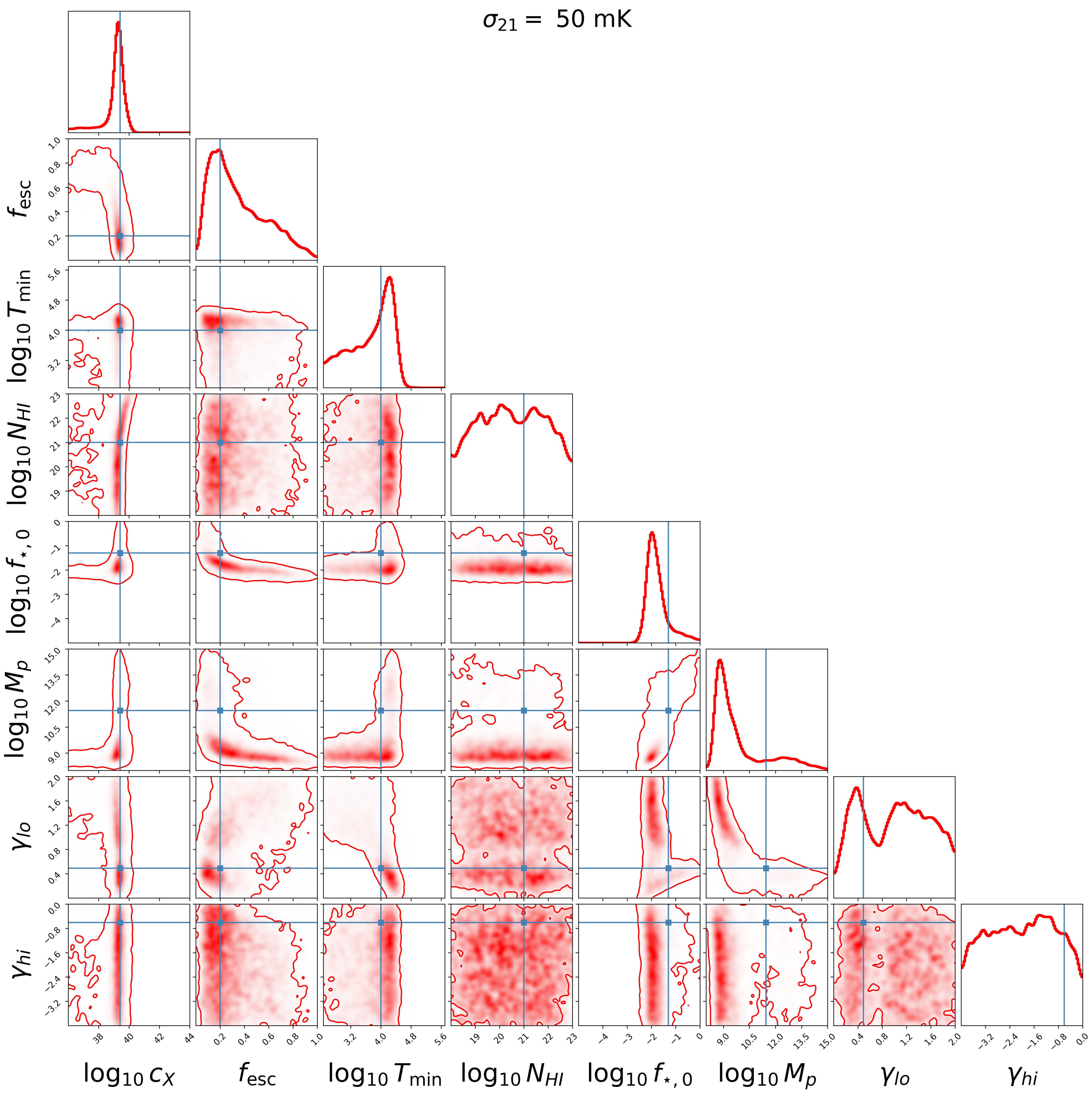

Figure C3. Same as Figure C1 but when using 21cmKAN to fit an ARES signal with $\theta_0 = (c_X, f_{\rm esc}, T_{\rm min}, \log N_{\rm H}I, f_{\star,0}, M_{\rm p}, \gamma_{\rm lo}, \gamma_{\rm hi}) = (2.6 \times 10^{39}, 0.2, 10^4, 21.0, 0.05, 2.8 \times 10^{11}, 0.49, -0.61)$ and the noise level of $\sigma_{21} = 50$ mK added. Figure 6 shows posterior signal realizations for fits to this signal with different σ_{21} levels. See Section 2.6 of J. Dorigo Jones et al. (2023) for a description of how the parameter values of this mock signal were chosen.

ORCID iDs

J. Dorigo Jones https://orcid.org/0000-0002-3292-9784
B. Reyes https://orcid.org/0000-0003-3496-7490
D. Rapetti https://orcid.org/0000-0003-2196-6675
Shah Mohammad Bahauddin https://orcid.org/0000-0003-0016-5377
J. O. Burns https://orcid.org/0000-0002-4468-2117
D. W. Barker https://orcid.org/0009-0002-4218-4840

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. 2019, in Proc. 25th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining, KDD '19 (New York: ACM), 2623
Andrianomena, S., & Hassan, S. 2025, Ap&SS, 370, 14
Anstey, D., de Lera Acedo, E., & Handley, W. 2023, MNRAS, 520, 850
Ashton, G., Bernstein, N., Buchner, J., et al. 2022, NRMP, 2, 39
Bahauddin, S. M., & Rast, M. P. 2021, ApJ, 915, 36
Bassett, N., Rapetti, D., Tauscher, K., et al. 2021, ApJ, 923, 33
Belfiore, F., Ginolfi, M., Blanc, G., et al. 2025, A&A, 694, A212
Bernardi, G., Zwart, J. T. L., Price, D., et al. 2016, MNRAS, 461, 2847
Bevins, H. T. J., Fialkov, A., de Lera Acedo, E., et al. 2022, NatAs, 6, 1473
Bevins, H. T. J., Handley, W. J., Fialkov, A., de Lera Acedo, E., & Javid, K. 2021, MNRAS, 508, 2923
Bevins, H. T. J., Heimersheim, S., Abril-Cabezas, I., et al. 2024, MNRAS, 527, 813
Bosman, S. E. I., Davies, F. B., Becker, G. D., et al. 2022, MNRAS, 514, 55
Breitman, D., Mesinger, A., Murray, S. G., et al. 2024, MNRAS, 527, 9833
Buchner, J. 2023, StSur, 17, 169
Bye, C. H., Portillo, S. K. N., & Fialkov, A. 2022, ApJ, 930, 79
Chatterjee, A., Choudhury, T. R., & Mitra, S. 2021, MNRAS, 507, 2405
Cheon, M., & Mun, C. 2024, RemS, 16, 3417
Cohen, A., Fialkov, A., Barkana, R., & Monsalve, R. 2021, Datset for 21cmVAE v1, Zenodo, doi:10.5281/zenodo.5084113
Cohen, A., Fialkov, A., Barkana, R., & Monsalve, R. A. 2020, MNRAS, 495, 4845
Cui, J., Biesiada, M., Liu, A., et al. 2025, ApJS, 280, 9
Cui, S., Cao, M., Liao, Y., & Wu, J. 2025, PhFl, 37, 037159
Cybenko, G. 1989, Math. Control Signal Systems, 2, 303
de Boor, C. 1980, MaCom, 34, 325, http://www.jstor.org/stable/2006241
de Lera Acedo, E., de Villiers, D. I. L., Razavi-Ghods, N., et al. 2022, NatAs, 6, 984
Denison, D., Mallick, B., & Smith, A. 1998, Statistics and Computing, 8, 337
Dorigo Jones, J. 2024, ARES dataset for 21cmLSTM, v1.0.0, Zenodo, doi:10.5281/zenodo.13840725
Dorigo Jones, J., Bahauddin, S. M., Rapetti, D., Mirocha, J., & Burns, J. O. 2024, ApJ, 977, 19
Dorigo Jones, J., Rapetti, D., Mirocha, J., et al. 2023, ApJ, 959, 49
Dorigo Jones, J., & Reyes, B. 2025, 21cmKAN, v1.0.0, Zenodo, doi:10.5281/zenodo.16822024
Fan, X., Strauss, M. A., Becker, R. H., et al. 2006, AJ, 132, 117
Feroz, F., & Hobson, M. P. 2008, MNRAS, 384, 449
Feroz, F., Hobson, M. P., & Bridges, M. 2009, MNRAS, 398, 1601
Feroz, F., Hobson, M. P., Cameron, E., & Pettitt, A. N. 2019, OJAp, 2, 10
Fialkov, A., Barkana, R., & Visbal, E. 2014, Natur, 506, 197
Fialkov, A., Barkana, R., Visbal, E., Tseliakhovich, D., & Hirata, C. M. 2013, MNRAS, 432, 2909
Francom, D., & Sansó, B. 2020, Journal of Statistical Software, 94, 1
Friedman, J. H. 1991, The Annals of Statistics, 19, 1
Furlanetto, S. R., Oh, S. P., & Briggs, F. H. 2006, PhR, 433, 181
Genet, R., & Inzirillo, H. 2024, TKAN: Temporal Kolmogorov-Arnold Networks Tech. Rep. 4825654, Université Paris-Dauphine
Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, Natur, 585, 357
Haslam, C. G. T., Salter, C. J., Stoffel, H., & Wilson, W. E. 1982, A&AS, 47, 1
Hibbard, J. J., Rapetti, D., Burns, J. O., Mahesh, N., & Bassett, N. 2023, ApJ, 959, 103
Hibbard, J. J., Tauscher, K., Rapetti, D., & Burns, J. O. 2020, ApJ, 905, 113
Hornik, K., Stinchcombe, M., & White, H. 1989, NN, 2, 359
Hunter, J. D. 2007, CSE, 9, 90
Jin, X., Yang, J., Fan, X., et al. 2023, ApJ, 942, 59
Kamai, I., & Perets, H. B. 2025, AJ, 169, 59
Kern, N. S., Parsons, A. R., Dillon, J. S., et al. 2020, ApJ, 888, 70
Kim, B., Wattenberg, M., Gilmer, J., et al. 2018, in Proc. Machine Learning Research, 80, Proc. 35th Int. Conf. on Machine Learning, ed. J. Dy & A. Krause (Maastricht: MLR Press), 2668, https://proceedings.mlr.press/v80/kim18d.html
Kingma, D., & Ba, J. 2015, in 3rd Int Conf. on Learning Representations (ICLR) (San Diego, CA: ICLR), 13
Kluyver, T., Ragan-Kelley, B., Pérez, B., et al. 2016, in Positioning and Power in Academic Publishing: Players, Agents and Agendas, ed. F. Loizides & B. Scmidt (Netherlands: IOS Press), 87, https://eprints.soton.ac.uk/403913/
Koh, P. W., & Liang, P. 2017, in Proceedings of the 34th International Conference on Machine Learning, 70, ed. D. Precup & Y. W. Teh (Maastricht: MLR Press), 1885, https://proceedings.mlr.press/v70/koh17a.html
Kolmogorov, A. N. 1957, Dokl. Akad. Nauk USSR, 114, 953, https://cir.nii.ac.jp/crid/1570009749757512064
Kraus, J. 1966, Radio Astronomy (New York: McGraw-Hill)
Lemos, P., Weaverdyck, N., Rollins, R. P., et al. 2023, MNRAS, 521, 1184
Liaw, R., Liang, E., Nishihara, R., et al. 2018, arXiv:1807.05118
Lipton, Z. C. 2018, Commun. ACM, 61, 36
Liu, H., Liu, C., Wang, J. T. L., & Wang, H. 2019, ApJ, 877, 121
Liu, Z., Ma, P., Wang, Y., Matusik, W., & Tegmark, M. 2024, arXiv:2408.10205
Liu, Z., Wang, Y., Vaidya, S., et al. 2025, arXiv:2404.19756
McGreer, I. D., Mesinger, A., & D'Odorico, V. 2015, MNRAS, 447, 499
Mesinger, A., Furlanetto, S., & Cen, R. 2011, MNRAS, 411, 955
Mirocha, J. 2014, MNRAS, 443, 1211
Mirocha, J., Furlanetto, S. R., & Sun, G. 2017, MNRAS, 464, 1365
Mirocha, J., Skory, S., Burns, J. O., & Wise, J. H. 2012, ApJ, 756, 94
Montavon, G., Samek, W., & Müller, K.-R. 2018, DSP, 73, 1
Murray, S. G., Bowman, J. D., Sims, P. H., et al. 2022, MNRAS, 517, 2264
Pagano, M., Sims, P., Liu, A., et al. 2024, MNRAS, 527, 5649
Paszke, A., Gross, S., Massa, F., et al. 2019, in Advances in Neural Information Processing Systems 32 (NeurIPS 2019), 32, ed. H. Wallach, H. Larochelle, A. Beygelzimer et al. (Red Hook, NY: Curran Associates), https://papers.nips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html
Piras, D., & Spurio Mancini, A. 2023, OJAp, 6, 20
Qin, Y., Mesinger, A., Park, J., Greig, B., & Muñoz, J. B. 2020, MNRAS, 495, 123
Rapetti, D., Tauscher, K., Mirocha, J., & Burns, J. O. 2020, ApJ, 897, 174
Rudin, C. 2019, NatMI, 1, 206
Rumelhart, D. E., Hinton, G. E., & Williams, R. J. 1986, Natur, 323, 533
Saxena, A., Meerburg, P. D., de Lera Acedo, E., Handley, W., & Koopmans, L. V. E. 2023, MNRAS, 522, 1022
Saxena, A., Meerburg, P. D., Weniger, C., Acedo, E. d. L., & Handley, W. 2024, RASTI, 3, 724
Shaver, P. A., Windhorst, R. A., Madau, P., & de Bruyn, A. G. 1999, A&A, 345, 380
Shen, E., Anstey, D., de Lera Acedo, E., & Fialkov, A. 2022, MNRAS, 515, 4565
Shi, H., Huang, Z., Yan, Q., et al. 2025, arXiv:2507.09632
Shukla, K., Toscano, J. D., Wang, Z., Zou, Z., & Karniadakis, G. E. 2024, CMAME, 431, 117290
Siemiginowska, A., Eadie, G., Czekala, I., et al. 2019, BAAS, 51, 355
Simonyan, K., Vedaldi, A., & Zisserman, A. 2014, Workshop at Int. Conf. on Learning Representations 2014, 8, https://www.robots.ox.ac.uk/~vgg/publications/2014/Simonyan14a/
Sims, P. H., Bowman, J. D., Mahesh, N., et al. 2023, MNRAS, 521, 3273
Singh, S., Subrahmanyan, R., Udaya Shankar, N., et al. 2018, ApJ, 858, 54
Skilling, J. 2004, in AIP Conf. Ser. 735, Bayesian Inference and Maximum Entropy Methods in Science and Engineering: 24th Int. Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, ed. R. Fischer, R. Preuss, & U. V. Toussaint (Melville, NY: AIP), 395
University of Colorado Boulder Research Computing 2023, Alpine (Boulder: Univ. Colorado)
Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, NatMe, 17, 261
Visbal, E., Barkana, R., Fialkov, A., Tseliakhovich, D., & Hirata, C. M. 2012, Natur, 487, 70
Wang, H., Fu, T., Du, Y., et al. 2023, Natur, 620, 47
Wehenkel, A., Gamella, J. L., Sener, O., et al. 2025, arXiv:2405.08719