Modeling early-universe energy injection with dense neural networks

Yitian Sun[*] and Tracy R. Slatyer[†]

Center for Theoretical Physics, Massachusetts Institute of Technology,
Cambridge, Massachusetts 02139, USA
and The NSF AI Institute for Artificial Intelligence and Fundamental Interactions,
Cambridge, Massachusetts 02139, USA

(Received 2 September 2022; accepted 3 March 2023; published 27 March 2023)

We show that dense neural networks can be used to accurately model the cooling of high-energy particles in the early universe, in the context of the public code package DarkHistory. DarkHistory self-consistently computes the temperature and ionization history of the early universe in the presence of exotic energy injections, such as might arise from the annihilation or decay of dark matter. The original version of DarkHistory uses large precomputed transfer function tables to evolve photon and electron spectra in redshift steps, which require a significant amount of memory and storage space. We present a light version of DarkHistory that makes use of simple dense neural networks to store and interpolate the transfer functions, reducing their total size by a factor of ~ 400, from ~ 17 Gb to ~ 45 Mb, which performs well on small computers with limited memory and storage. This method anticipates future expansion with additional parametric dependence in the transfer functions without requiring exponentially larger data tables.

DOI: 10.1103/PhysRevD.107.063541

I. INTRODUCTION

Dark matter (DM) constitutes 84% of the matter content in the universe [1] and plays an important role in the evolution of the early universe. It has so far eluded detection in all channels other than gravitational interactions. DM annihilation or decay could inject energy in the form of Standard Model particles, modifying the temperature and ionization of the intergalactic medium (IGM) and the anisotropies of the cosmic microwave background (CMB); studies of these observables have placed strong constraints on such energy injections (e.g., [2–15]).

DarkHistory [16] is a Python package developed to calculate the evolution of the IGM temperature and ionization in the early universe in the presence of such exotic energy injections. For an injected spectrum of Standard Model (SM) particles, it calculates the particle cascade by computing (1) the production of photons and electrons/positrons by the decay of the originally injected SM particles; (2) the subsequent secondary particle cascade and energy deposition arising from this exotic injection of photons/electrons/positrons, due to interaction with the IGM and the photon bath; and (3) modifications to the IGM's temperature and ionization from the secondary particles and their energy deposition, using a simple three-level atom (TLA) model.

These calculations are carried out in redshift steps starting prior to recombination (at redshift $1+z=3000$ by default) and ending well after reionization near the present day (redshift $1+z=4$). In particular, the particle cascade in step (2) is evaluated using precomputed transfer functions, which are matrices that take an input spectrum and output the spectrum of secondary particles (for a given redshift step). DarkHistory includes the backreaction effects of changes to the ionization level of matter, which means the transfer functions themselves are functions of the gas ionization levels, as well as redshift. In the previous version of DarkHistory, this dependence is realized by interpolating tables of transfer function matrices on a grid of values for the hydrogen and helium ionization fractions, as well as a grid of redshift values.

At around 1.5 Gb per table with 12 tables around this size, the transfer functions take up significant storage space as well as memory during runtime, since they are all loaded in a standard run. They will also be difficult to scale up to include additional parametric dependence, as the expected size scales exponentially with the number of added parameters. Intuitively, storing the transfer functions as tables is an overrepresentation of the information content within, since the transfer functions, despite not being smooth globally, can be divided into multiple regions that are relatively smooth, each of which could plausibly be fitted

[*] yitians@mit.edu
[†] tslatyer@mit.edu

with analytical functions. However, in practice, finding such a solution is quite nontrivial, and even if a solution was found by *ad hoc* methods, it would be quite specific to individual transfer functions and potentially difficult to maintain under changes to the code. In this work, we present a general solution to these issues by replacing the transfer function tables with trained dense neural networks.

Recently, machine learning and especially neural networks (NNs) has seen many applications in high energy physics and astrophysics [17–20]. NNs are often used to capture highly nonlinear and complex relations between inputs and outputs, and in particular, dense neural networks (DNNs), also known as fully connected neural networks, are a class of general function approximators given sufficient neuron numbers [21].

In the updated version of DarkHistory we present in this work, we use lightweight DNNs to store and automatically interpolate DarkHistory's transfer functions. A transfer function is essentially a multidimensional table, with 2 dimensions corresponding to the input and output particle energy, and the rest corresponding to physical parameters which in this case will be a subset of redshift z, ionized hydrogen fraction $x_{\mathrm{HII}} \equiv n_{\mathrm{HII}}/n_{\mathrm{H}}$, and singly ionized helium fraction $x_{\mathrm{HeII}} \equiv n_{\mathrm{HeII}}/n_{\mathrm{H}}$. (As a matter of convenience, we define the singly ionized helium fraction with the *hydrogen* number density as the denominator, so that x_{HeII} and x_{HII} can be easily summed.) The DNN-based transfer functions are shown in schematic form in Fig. 1. We let a DNN take in all of the relevant parameters on equal footing and predict (the natural logarithm of) the transfer function value P.

Using DNNs as transfer functions has several benefits:

(i) The DNNs we use are smaller in size compared to the stored transfer function tables, by a factor of ∼400.

(ii) The computed matter temperature history and ionization history match those calculated using the transfer function tables to within a few percent relative difference (with subpercent relative differences in regions when the species in question are more than 10% ionized), while the spectral distortion due to upscattered CMB photons (see Sec. II for precise definition) matches to below the 10 percent level. These errors are small compared to current experimental uncertainties.

(iii) We expect the DNNs to have improved scaling in size when additional parameters are added compared to the original tables. (Including a smooth dependence on an additional parameter might result in a $\mathcal{O}(1)$ increase in the DNN neuron number to reach similar accuracy due to increased information content. On the other hand, adding an additional dimension to a data table would increase its size by a multiplicative factor of the number of bins in the new parameter.)

(iv) The DNNs automatically interpolate to any the physical parameter values and input/output particle energies within the trained range. This allows the use of flexible binning in DarkHistory, and will also allow us to perform interpolation on sparse training data [22,23]. The latter may become necessary in future extensions of DarkHistory, when probing dependence on an increasing number of physical parameters and generating dense grids of training data becomes computationally expensive.

(v) The DNNs predict transfer functions quickly, taking up a similar amount of time to the rest of the evolution routine for injected particles. Thus compared to retrieving tabular data from memory on a personal computer, the use of DNNs results in only a $\mathcal{O}(1)$ increase in total runtime.

(vi) Open source building and training tools for NNs and especially simple architectures like DNNs are readily available. (In this work we use TensorFlow 2.0 [24] with Keras [25].)

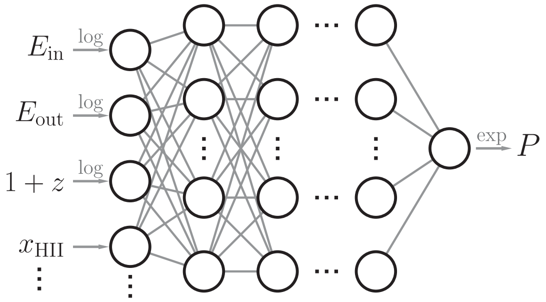


FIG. 1. Model schematics of a dense neural network transfer function. A DNN takes in (the logarithm of) the input and output photon/electron energy and necessary physical parameters including the redshift z, ionized hydrogen fraction x_{HII}, and singly ionized helium fraction x_{HeII}, and outputs (the logarithm of) the transfer function value P. The transfer function matrix acting on a given discretized spectrum is then obtained by evaluating the DNN on the given energy abscissa.

In Sec. II, we introduce DarkHistory and the roles of transfer functions. In Sec. III, we detail the training and implementation of the DNN transfer functions. In Sec. IV we present test runs and discuss the performance of DNN transfer functions compared to baseline DarkHistory. Finally, in Sec. V we summarize our results, briefly discuss other possible approaches, and outline some ideas for future expansion with DNN transfer functions in DarkHistory.

II. TRANSFER FUNCTIONS IN DarkHistory

To better illustrate the role of transfer functions, we briefly introduce the procedure followed in DarkHistory and sketched in Fig. 2, which is modified from a flow chart in Ref. [16]. In Fig. 2, boxed quantities represent particle spectra, and arrows represent transfer functions, which are functions acting on spectra. DarkHistory stores the free

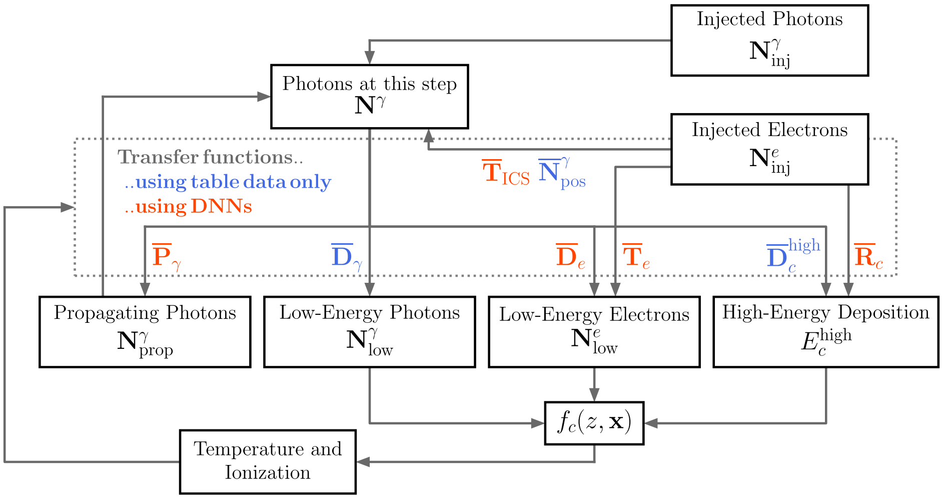

Injected Photons
$\mathbf{N}^{\gamma}_{\rm inj}$
Photons at this step
$\mathbf{N}^{\gamma}$
Injected Electrons
$\mathbf{N}^{e}_{\rm inj}$
Transfer functions..
..using table data only
..using DNNs
$\overline{\mathbf{T}}_{\rm ICS}$ $\overline{\mathbf{N}}^{\gamma}_{\rm pos}$
$\overline{\mathbf{P}}_{\gamma}$
$\overline{\mathbf{D}}_{\gamma}$
$\overline{\mathbf{D}}_{e}$
$\overline{\mathbf{T}}_{e}$
$\overline{\mathbf{D}}^{\rm high}_{c}$
$\overline{\mathbf{R}}_{c}$
Propagating Photons
$\mathbf{N}^{\gamma}_{\rm prop}$
Low-Energy Photons
$\mathbf{N}^{\gamma}_{\rm low}$
Low-Energy Electrons
$\mathbf{N}^{e}_{\rm low}$
High-Energy Deposition
$E^{\rm high}_{c}$
$f_c(z, \mathbf{x})$
Temperature and
Ionization

FIG. 2. DarkHistory flow chart and transfer functions. Starting with injected photons and electrons, this flow chart illustrates the work flow of DarkHistory in the evolution of each redshift step. The boxed quantities are particle spectra and the arrows indicate transfer functions, which act on the particle spectra. The orange transfer functions are generated or reconstructed using DNNs, while the blue ones only use tabulated data. (Modified from Fig. 1 in Ref. [16].)

streaming photon spectrum, the IGM temperature, and the IGM's ionized hydrogen fraction and singly ionized helium fraction at each redshift step (these quantities are assumed to be homogeneous). For each redshift step, DarkHistory:

(1) Converts injected SM particles at that redshift to injected photons $\mathbf{N}^{\gamma}_{\rm inj}$ and electron/positrons $\mathbf{N}^{e}_{\rm inj}$ (hereafter referred collectively as electrons). In DarkHistory, high energy (>3 keV) positrons are treated as electrons since their dominant energy-loss process, inverse Compton scattering (ICS) on the CMB, does not depend on the particle charge. Lower-energy positrons are assumed to annihilate and the resulting photon spectrum is tracked; their kinetic energy is approximated as following the same pattern of energy deposition as that of the electrons (see Ref. [16] for a more in-depth discussion).

(2) Computes any injected electron spectrum's energy deposition into ionization, excitation, or heating by applying the transfer function $\bar{\mathbf{R}}_c$. Computes secondary photon and electron spectra produced from injected electrons due to ICS, positronium formation and decay, and atomic processes by applying the ICS transfer function $\bar{\mathbf{T}}_{\rm ICS}$ and secondary electron transfer function $\bar{\mathbf{T}}_e$, and evaluating the spectrum of gamma rays produced from positron annihilation $\bar{\mathbf{N}}^{\gamma}_{\rm pos}$. These secondary photon spectra are added to the spectrum of photons $\mathbf{N}^{\gamma}$ propagated from the previous step, plus any injected photon spectrum.

(3) Computes the secondary particles and energy deposition produced by a propagating photon spectrum $\mathbf{N}^{\gamma}$, due to a variety of processes, including photon-photon scattering, Compton scattering, pair production, photoionization, and redshifting. The production of secondary electrons/positrons in the same redshift step, and their subsequent production of photons via ICS on the CMB or positron annihilation, are also included. The propagating photon spectrum for the next redshift step is obtained by applying the high energy photon transfer function $\bar{\mathbf{P}}_{\gamma}$ to $\mathbf{N}^{\gamma}$. The low energy photon spectrum, which stores photons below 3 keV that either photoionize within the redshift step or lie below 13.6 eV, is obtained by applying the low energy photon transfer function $\bar{\mathbf{D}}_{\gamma}$. The low energy electron spectrum, which stores electrons with kinetic energy below 3 keV where atomic cooling dominates ICS and is treated separately in the electron cooling module, is obtained by applying the low energy electron transfer function $\bar{\mathbf{D}}_{e}$. Finally, the photon's energy deposition into ionization, excitation, and heating is obtained by applying the high energy deposition transfer function $\bar{\mathbf{D}}^{\rm high}_{c}$.

(4) Computes the change to IGM temperature and ionization by first calculating the energy deposition fraction f_c's, from the low-energy electron/photon spectra and direct energy deposition by higher-energy particles, and then performing the TLA integration (see Ref. [16] for details).

In general, transfer functions from an input spectrum to another spectrum (such as $\bar{\mathbf{T}}_{\rm ICS}$, $\bar{\mathbf{P}}_{\gamma}$) take up much more space than those outputting an energy value (such as $\bar{\mathbf{D}}^{\rm high}_{c}$) or those that are diagonal (such as $\bar{\mathbf{D}}_{\gamma}$). For this version of DarkHistory, we focus on replacing the largest spectral transfer functions with DNNs, but similar procedures can be applied to the lower-dimension transfer functions in the future. In the following we introduce the two major types of transfer functions we will replace in more detail.

A. ICS transfer functions

As discussed above, the ICS transfer functions describe the spectrum of scattered photons from the complete cooling of injected electrons. Unlike the transfer functions applied to the photon spectrum, the ICS transfer functions $\bar{\mathbf{T}}_{\mathrm{ICS}}$, $\bar{\mathbf{T}}_e$, and $\bar{\mathbf{R}}_c$ are not directly interpolated from tables, but reconstructed from reference ICS transfer function tables. Section III. D and Appendix A of Ref. [16] describe in detail how this is achieved, and we only provide a simple summary here: since an electron quickly deposits all of its energy in one of our redshift steps, in order to obtain the total secondary spectrum or energy deposition by an electron, we need to consider multiple interaction events (via ICS and atomic processes). This can be done recursively: one can reconstruct the full ICS secondary spectrum and energy deposition for an electron of energy E knowing the same information for all electrons with $E' < E$. With discretized energy abscissa, the full ICS-induced photon spectrum and energy output can be solved recursively from the lowest energy bin. As a result, at each redshift, one can solve for the full electron ICS transfer functions using transfer functions describing a single ICS scattering event (as well as functions describing the interaction rates due to atomic processes).

The transfer functions for a single ICS scattering event on the CMB have simple (approximate) scaling relations with respect to the temperature of the CMB T, as described in detail in Appendix A of Ref. [16]. As such one can derive ICS transfer functions at different redshifts from a single transfer function at a fixed redshift, assuming the CMB is the dominant radiation background (in DarkHistory, $1 + z = 400$ is used). These reference transfer functions are interpolated from tables `ics_thomson`, `ics_rel`, and `ics_engloss`, corresponding respectively to transfer functions for the secondary photon spectra of nonrelativistic electrons and relativistic electrons, and the relativistic electron energy loss in a single ICS scattering event on the CMB. It is these tables that we will fit with DNNs.

B. Photon transfer function

The high energy photon transfer function $\bar{\mathbf{P}}_\gamma$, low energy photon transfer function $\bar{\mathbf{D}}_\gamma$, and low energy electron transfer function $\bar{\mathbf{D}}_e$ are interpolated from corresponding tables `highengphot`, `lowengphot`, and `lowengelec`. They in general depend on the CMB temperature through redshift and matter ionization levels (ionized hydrogen fraction and singly ionized helium fraction). For each combination of these physical parameters, `lowengphot` can be represented as 1-D arrays (with the one dimension being input/output energy) and is a factor of 500 smaller than the other transfer functions, so it is at present not replaced with a DNN. We also found that the numerical calculation of Compton scattering used in the previous version of DarkHistory was inaccurate in some parts of parameter space (in particular populating kinematically forbidden regions), and so we have updated the relevant tabular transfer functions to ensure sufficient accuracy.[1] We now describe some special features of the photon transfer functions:

(a) *Redshift regimes and matter ionization dependence.* All photon transfer functions depend on the redshift, as the photon cooling processes involve interactions with the redshift-dependent photon background and/or interstellar medium. For late redshifts ($z < 40$) encompassing the epoch of reionization, the transfer functions are allowed to vary with both the ionized hydrogen fraction x_{HII} and the singly ionized helium fraction x_{HeII}, which can be altered by exotic energy injections. For redshifts between helium recombination and reionization ($40 < z < 1600$), the ionized helium fraction can be safely approximated as zero [16]; the transfer functions are precomputed assuming no helium ionization, but can depend on the hydrogen ionized fraction. Before helium recombination ($z > 1600$), exotic energy injections consistent with current experimental bounds have little impact on the thermal equilibrium determining hydrogen and helium ionization levels [16], so DarkHistory uses Recfast [27] ionization fractions as a baseline to precompute the transfer functions, which only depends on redshift.

For the DNN implementation, the flexibility of the network allows one DNN to be trained on the entire redshift range $4 < z < 3000$ with a learned dependence on the ionization levels, for each transfer function. However, using different networks for different redshift regimes gives slightly better accuracy, and the latter approach is chosen in this version of DarkHistory.

(b) *Redshift step coarsening and energy conservation.* In the previous version of DarkHistory, photon transfer functions are computed with redshift step $\Delta \log(1 + z) = 0.001$. One can choose to increase the (log) step size to multiples of 0.001 to speed up computation. To combine multiple redshift steps, DarkHistory precomposes multiple propagation transfer functions $\bar{\mathbf{P}}_\gamma$ and applies them appropriately to the deposition transfer functions $\bar{\mathbf{D}}_\gamma$, $\bar{\mathbf{D}}_e$, and $\bar{\mathbf{D}}_c^{\mathrm{high}}$. Since the DNN implementation of transfer functions introduces a small amount of error that can accumulate over many redshift steps, in order to increase numerical stability, we train the DNNs to reproduce the precomposed transfer functions, and require the use of a

[1] This issue primarily affected secondary electrons in the 10 eV to 3 keV range produced by Compton scattering of photons with energies between 100 eV and 10 s of keV; the effects on observable quantities were very small for all cases we checked. The updated tables on Zenodo [26] incorporate this correction.

fixed log redshift step of $\Delta\log(1+z) = 0.012$.[2] To further decrease numerical error, we store the total fraction of the injected energy entering each type of secondary particle spectrum, for an injected photon of any given energy, and use these data (which are of $\sim 1\%$ the size of the transfer functions) to ensure energy conservation while accounting for energy loss to redshifting. We discuss the precise procedure of transfer function precomposing and imposing energy conservation in Appendix A 1.

III. TRANSFER FUNCTIONS FROM NEURAL NETWORKS

As described earlier and as indicated in Fig. 1, we update the largest transfer function tables (ics_thomson, ics_rel, ics_engloss, highengphot, and lowengelec) to DNN networks that take in input and output particle energies, redshift, and possibly (depending on redshift) the ionized hydrogen fraction and singly ionized helium fraction, to produce the transfer function value P. In general, P can vary greatly across many orders of magnitude. (For example, after recombination the probability for a 10 keV photon to free stream, losing energy only through redshifting, is substantial, while the chance of it directly producing secondary photons of 1 keV is very close to 0, since this outcome is not kinematically allowed in a single Compton scattering event, nor is the scattered electron produced by Compton scattering or photoionization able to up-scatter other photons to 1 keV.) As such, we train the networks to output the natural logarithm of the transfer function value $\log P$. Similarly, since the input/output energy abscissa and our redshift steps are also binned in log space by default, the networks also take the log value of these as inputs. The ionized hydrogen fraction and singly ionized helium fraction are linearly scaled to match the spread of other parameters before being fed into the DNN.

For high energy photons, the redshift-coarsened transfer functions (see Sec. II B) are fitted. Note also that the transfer function value P can be negative since by convention the CMB spectrum is subtracted from the transfer function and the negative values (and the positive values at higher energies) represent CMB photons being upscattered. In this case $\log|P|$ is predicted by the DNNs, and the negative value region is recovered in a postprocessing step that identifies local minima of $\log|P|$, which takes up an negligible time compared to the rest of the DarkHistory routine. (The sign of the value at the minimum is ambiguous, so we linearly interpolated the transfer function values of the two closest bins in the output energy. We expect the errors coming from this step to be small.) Additionally, the transfer functions values near the diagonal in the input/output energy dimensions (corresponding to the free-streaming photons) are adjusted to enforce energy conservation up to redshifting. For details please refer to Appendix A 1.

After optimizing hyperparameters describing the model and training, via grid searching, we find that for all transfer functions in question, the best performance was obtained by using DNNs with 7 hidden layers with 400 neurons each, making the number of parameters per DNN $\sim 400^2 \times (7\text{–}1) \sim 9.7 \times 10^5$, and 2.9×10^6 for each transfer function built from 3 such networks.

Training is done with TensorFlow 2.0 [24] and Keras [25]; Adagrad [28] is used as the optimizer, with the mean squared error of $\log P$ as the loss function. Each DNN is trained on 2 Nvidia V100 GPUs for $\mathcal{O}(10)$ hours or equivalent. For each epoch, training data and evaluation data are separately generated by interpolating the multidimensional transfer function table on uniformly random sampled inputs. Since training data is not reused across epoch, there is no concern of overfitting to a fixed subset of full dataset. We estimate the number of epochs at which the loss function evaluation stops improving significantly, and terminate trainings at twice that number of epochs. At this point the best model is selected. To check for systematic offsets of the table transfer functions and the NNs, we trained multiple NNs with random initial values and randomly sampled training data. We found no obvious systematic offsets between the ensemble of NNs and the tables; e.g., at any given point the different NNs both underpredicted and overpredicted the table data. During initial trainings, we identified locations where the NN accuracy was limited by interpolation errors when the training data was generated from linear-interpolating grid data. (This only occurred at high redshifts of $1+z > 1600$, before recombination, thus this error had minimal impact on viable DM energy injection scenarios.) We then updated the interpolation table with denser grid points such that the interpolation error is always smaller than the errors reached by the NNs.

The codes related to the DNN transfer functions are stored in the nntf module under DarkHistory. A new example file “Example 12: Using Neural Network transfer functions.ipynb” [29] is provided to demonstrate using the DNN transfer functions and comparison with the baseline DarkHistory (which will be available if the appropriate data tables are present).

IV. PERFORMANCE

In this section, we describe the accuracy and speed of generation of the DNN transfer functions, as well as the accuracy of full runs over a range of simple DM injection scenarios using the DNN transfer functions.

[2] It was shown in [16] that an increased redshift step size at this level will lead to errors smaller than 10^{-3}, and so will not be the primary contributor to numerical error in our context. Note also that using a n-fold coarsened log redshift step is not equivalent to using every nth entry in the transfer function table. The redshift abscissa used for the table are not directly related to (and in general are coarser than) the redshift steps used during a run; the table is interpolated to obtain the transfer functions at each redshift step.

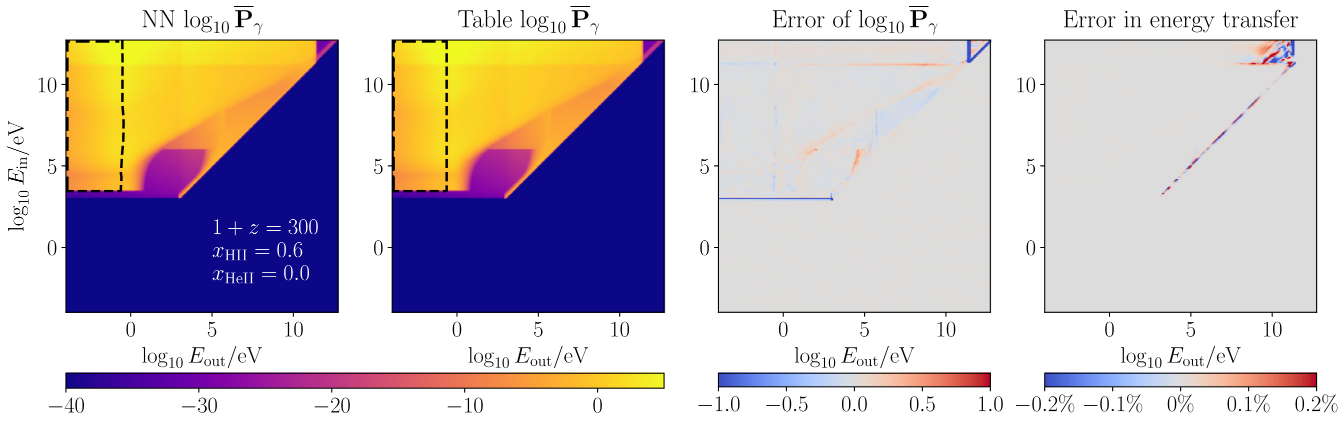


FIG. 3. Comparison of DNN and tabulated transfer functions. The first two panels show the high energy photon transfer functions generated at redshift $1+z=300$, ionized hydrogen fraction $x_{\rm HII}=0.6$ (and singly ionized helium fraction $x_{\rm HeII}=0.0$). The transfer function value in the region encircled by the black dashed line is negative and the $\log_{10}$ of the absolute value is shown. The third panel shows the error in $\log_{10}\bar{\mathbf{P}}_\gamma$ (defined as $\Delta\log_{10}\bar{\mathbf{P}}_\gamma \equiv \log_{10}|\bar{\mathbf{P}}_{\gamma\rm DNN}| - \log_{10}|\bar{\mathbf{P}}_{\gamma\rm table}|$), which is concentrated on the physical features of the transfer functions. The fourth panel shows the relative error in the energy transfer function (see more details in Sec. IV A). The errors are concentrated on the diagonal that corresponds to photon propagation and redshifting.

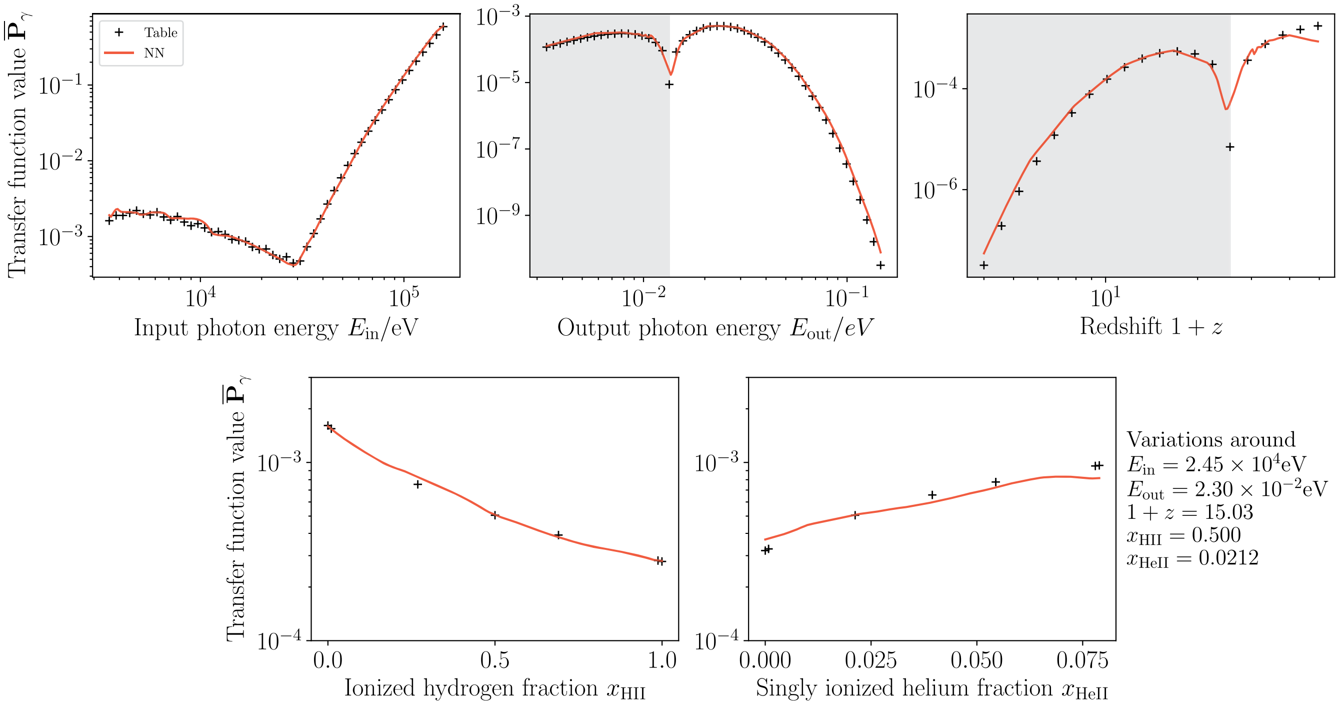
A. Transfer function value prediction

In Fig. 3, a high energy photon transfer function generated by a DNN is compared against one interpolated from tables. As one can see, the errors in the raw output i.e. logarithm value of the transfer functions are concentrated near the distinct physical features in the transfer function, such as at the output photon energy of ~ 0.5 MeV corresponding to positronium decay. The absolute values in logarithm errors $\Delta\log_{10}|P|$ can be interpreted as relative errors in $|P|$ (up to a ln 10 factor). In this particular slice through the high energy photon transfer function, the average $\Delta\log_{10}|P|$ when $|P|>10^{-20}$ is 0.017, corresponding to a relative error of $\sim 4\%$. (The range of $\log_{10}|P|$ is about ~ -45 to 6). Overall $\Delta\log_{10}|P|$ is comparable to this value, for all DNN transfer functions. A summary of the errors can be found in Table I.

To see the impact of these errors in a DarkHistory evolution run, it is also useful to look at errors in energy (per bin) transition rates, besides the particle number transition rates.

TABLE I. Error, prediction time, and size of DNN transfer functions used in DarkHistory. The error of the DNN transfer functions in comparison to the tabular transfer functions is defined as $|\Delta\log_{10}|P|| \equiv |\log_{10}|P_{\rm DNN}| - \log_{10}|P_{\rm table}||$ when $|P|>10^{-20}$. Both the errors and prediction times are evaluated from random draws in the relevant domains for each DNN until the values stabilize. The prediction times are evaluated on a 8-CPU personal computer and on a cluster with 2 Tesla V100 GPUs. For comparison, in each redshift step (at certain redshift regime), the high energy photon (compounded), high energy photon (propagator), and low energy electron transfer functions are generated, while the remainder of the run time in the step takes around 0.2–0.3 seconds. All DNN transfer functions use 6 hidden layers of 400 neurons each, but the evaluation time differs due to the length of the input array corresponding to the size of the evaluated transfer function matrix ($\sim 10^5$ for high energy photon, $\sim 4\times10^4$ for low energy electron, $\sim 2.5\times10^5$ for ICS).

		Prediction time			
Transfer function	DNN $\|\Delta\log_{10}\|P\|\|$	CPU (s)	GPU (s)	Table size (Gb)	NN size (Mb)
High energy photon (compounded) regime 0	0.029	1.01	0.126	4.7	11.4
High energy photon (compounded) regime 1	0.043	1.04	0.127		
High energy photon (compounded) regime 2	0.016	1.02	0.127		
High energy photon (propagator) regime 0	0.029	1.02	0.128	4.7	11.4
High energy photon (propagator) regime 1	0.043	1.05	0.129		
High energy photon (propagator) regime 2	0.062	1.07	0.128		
Low energy electron regime 0	0.047	0.384	0.079	4.7	11.4
Low energy electron regime 1	0.049	0.380	0.080		
Low energy electron regime 2	0.040	0.380	0.081		
ICS Thomson	0.00199	0.045	0.037	0.93	3.8
ICS relativistic	0.00410	0.045	0.037	0.93	3.8
ICS electron energy loss	0.00250	0.046	0.037	0.93	3.8



Variations around
$E_{\text{in}} = 2.45 \times 10^4\text{eV}$
$E_{\text{out}} = 2.30 \times 10^{-2}\text{eV}$
$1 + z = 15.03$
$x_{\text{HII}} = 0.500$
$x_{\text{HeII}} = 0.0212$

FIG. 4. The DNN high energy photon transfer function network's fitting and interpolation of the table transfer function values. The panels show the DNN transfer function's (redshift regime $4 < z < 40$) dependence on each of the five input variables, in a limited region around a central value indicated on the plot. The shaded region is where both the table value and the DNN prediction are negative (and the log of the absolute value is shown). A fairly consistent relative error below 10% is observed.

The energy transfer function deemphasize errors at low energies where many particles can be produced with only a small fraction of the original particle's energy, and thus such errors have a small effect on heating and ionization. Since the standard photon transfer functions are maps between particle number spectra N_i, the transfer function values have the physical meaning of number transition rates. The particle energy spectrum is related to the number spectrum by

$$E_i = \mathcal{E}_i N_i = \mathcal{E}_i^2 \left(\frac{\mathrm{d}N}{\mathrm{d}\mathcal{E}}\right)_i, \tag{1}$$

where N_i is the number of particles in the ith bin, and $\mathcal{E}_i$ its central energy. (Note that the energy bins are log-spaced.) For a particle number transfer function P with

$$N_i^{\text{out}} = \sum_j N_j^{\text{in}} P_{ji}, \tag{2}$$

the corresponding energy transfer function P^E is defined by

$$E_i^{\text{out}} = \sum_j E_j^{\text{in}} P_{ji}^E \Rightarrow P_{ji}^E = P_{ji}\mathcal{E}_i/\mathcal{E}_j. \tag{3}$$

In the last panel of Fig. 3, we show the relative error in the high energy photon transfer function. As expected, the errors are concentrated on the highest output energy for a given input energy. The relative errors are generally subpercent.

Figure 4 shows that the DNN networks interpolate sensibly between the fixed abscissa values in the transfer function tables, for the high energy photon transfer function in the lowest redshift regime ($4 < z < 40$), as an example. The errors in the transfer function interpolation are consistent with average values shown in Table I.

B. Performance over a range of scenarios

Figure 5 shows the evolution of integrated variables in one particular setting: 0.1 GeV DM particles decaying into electron-positron pairs (this is the same as the example used in Fig. 4 of Ref. [16]). In this example, the error introduced by the DNN in the matter temperature history and hydrogen ionization history is consistently subpercent, while the error in singly ionized helium fraction is subpercent when $n_{\text{HeII}}/n_{\text{He}} > 10^{-3}$. For a scenario where DM decays into electron-positron pairs, over a range of DM rest masses, the maximum relative error for temperature and ionized hydrogen fraction over the entire redshift range is consistently below 2%, as shown in Fig. 6. Taking into account injection scenarios with DM decaying to photons, and also undergoing s-wave annihilation into e^+e^- or photons, the relative error is always below 8%. As a comparison, various approximations entering the DarkHistory transfer functions calculation—such as in certain cross sections relevant to the secondary particle cascade, the treatment of helium,

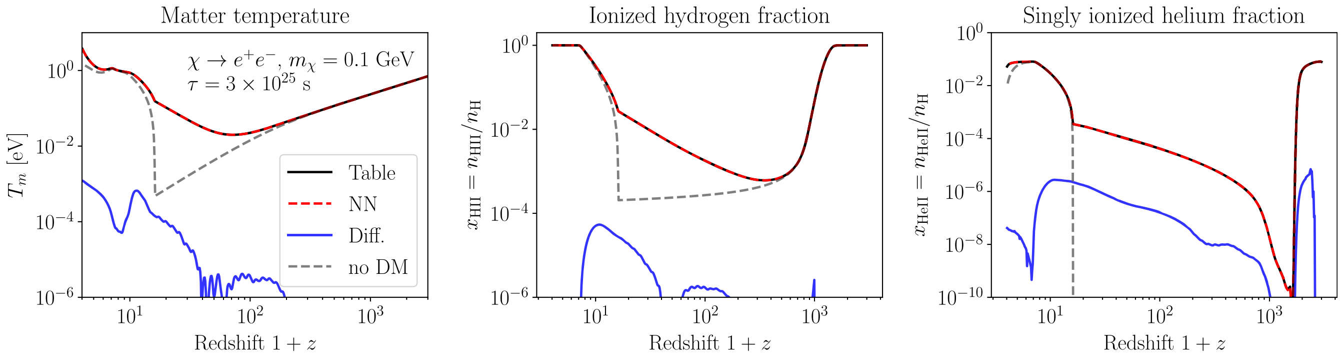


FIG. 5. Example evolution of matter temperature and ionization levels. The evolution of matter temperature and ionization levels, in an example scenario where 0.1 GeV DM is decaying into e^+e^- pairs, as predicted using the original tabulation (black lines) and using the DNN transfer functions (red dashed lines). The blue line shows the difference between the two, and the gray dashed line shows the standard evolution without any external injections.

and the modeling of low-energy particles—can affect the temperature and ionization history at the level of 10%, as studied in [9,16]. Thus few-percent-level errors are a subdominant contribution to the known error budget, and reducing them further will not meaningfully improve the accuracy of the forecast signals. DarkHistory also computes the partial photon spectral distortion from high energy photon and electron processes, mostly from ICS of electrons or positrons on the CMB. This distortion is stored in the low energy photon spectrum. Note that spectral distortions arising from atomic transitions, due to photons and electrons below 3 keV, are not included in this spectrum (which is why we label it as "partial" or "incomplete"). The DNN transfer functions introduce a small amount of error in this spectral distortion, as shown in Figs. 6 and 7. While the shape of the spectral distortion is generally correct, the error in the location of the distortion zero can cause errors with a magnitude up to 10% of the distortion magnitude due to the errors in the distortion zero location. In a future update of DarkHistory, we are anticipating an update to include the correct treatment of the complete photon spectral distortion including contributions from atomic transitions. A small photon spectral error would allow the DNN functions to be used simultaneously with these updates.

In Appendix A 2, we include the errors in some other exotic injection scenarios, including DM decaying to photon pairs, and DM annihilating to photon or e^+e^- pairs. Although these examples cover only a few simple injection scenarios, they can serve to test the whole range of

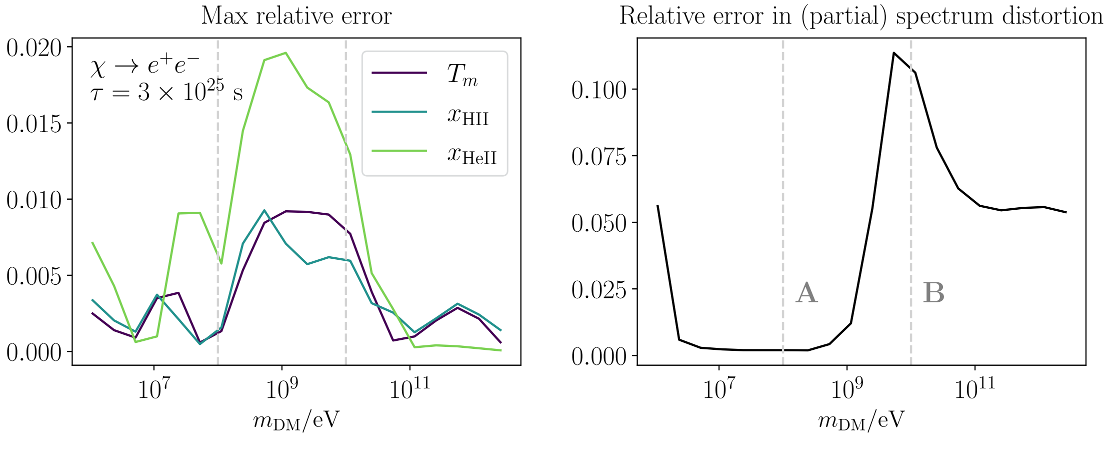


FIG. 6. Relative errors in temperature, ionization levels, and low energy photon distortion across a range of DM mass in a scenario with DM decaying to e^+e^-. The left panel shows the maximum relative error at any point in the evolution in $4 < z < 3000$ of matter temperature, ionized hydrogen fraction, and singly ionized helium fraction across a range of DM exotic electron injection energies*. (*Due to its very small absolute value, the relative error for singly ionized helium fraction $x_{\rm HeII}$ when $n_{\rm HeII}/n_{\rm He} < 10^{-3}$ is not included in this plot. Since $n_{\rm HeII}/n_{\rm He}$ changes rapidly between $\mathcal{O}(1)$ and $< 10^{-3}$, we are essentially only counting its relative error when its value is order unity.) The relative errors are generally below 5%. The right panel shows the maximum error over the maximum value in the low energy photon spectral distortion at $z = 0$. The spectral distortion errors at the two dashed-line values of $m_{\rm DM} = 0.1$ GeV and 10 GeV are shown in Fig. 7.

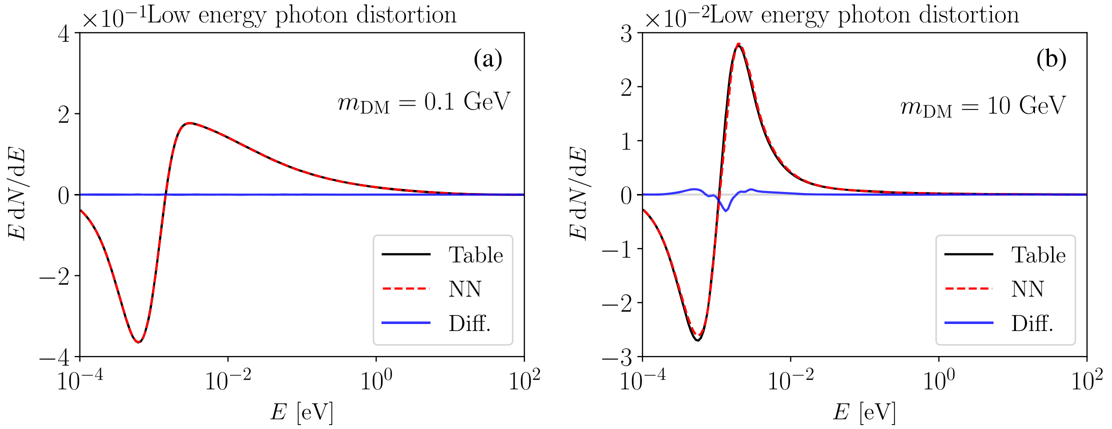


FIG. 7. Example partial low-energy photon spectral distortion in the present day. The two panels show two examples of low energy photon spectral distortion outputs (see definition in Sec. II) from two different runs: 0.1 GeV and 10 GeV DM decaying to e^-e^+ pairs. They represent the extremes of large and small relative errors for runs over the full range of DM masses we consider, as shown in Fig. 6. The photon number density (per bin) is normalized against the baryon number density. The black lines represent outputs generated using tabulated transfer functions while the red dashed lines represent that using the DNN transfer functions. The blue line shows the different between the two. Note that the relatively large errors shown in the right panel are in part due to the error in the location of the zero.

DarkHistory's dependence on transfer functions, since all exotic energy injections are converted to either e^+e^- or photon injections. We expect exotic energy injections with a more complicated injection spectrum (such as, e.g., annihilation to quarks) to have similar amounts of error associated with using the DNN transfer function.

C. Compute time, memory, and disk usage

The time it takes to generate a NN photon transfer function is $\lesssim 1$ second each on an 8-CPU personal computer. In each time step, 3 photon transfer functions are generated, thus taking more time than the remainder of the time step, which runs at about 0.26 seconds. On 2 Tesla V100 GPUs, the NN prediction times are reduced by about a factor of 10 compared to on CPUs; the time to generate all 3 photon transfer functions is about 0.34 seconds, comparable to the remainder of the time step. (The remainder of the processes are not GPU-accelerated.)

The on-disk sizes of photon transfer functions are reduced from their table counterparts of 4.7 Gb each by a factor of 420 to 11.4 Mb each, while the size of the smaller ICS transfer functions are reduced from 0.93 Gb each by a factor of 250 to 3.8 Mb each. The total reduction of on-disk size is about a factor of 380. The memory usage of the NN transfer functions will depend on the implementation of TensorFlow: on an 8-CPU computer, the memory usage is reduced by more than a factor of 10 compared to reading in table transfer functions, from using 15 Gb to less than 1 Gb. On 2 Tesla V100 GPUs, about 1 Gb of memory is used, while $\mathcal{O}(1)$ of the dedicated GPU memory of 64 Gb is used.

The accuracy, prediction time, and on-disk sizes for all other transfer functions are summarized in Table I.

V. CONCLUSION

In this work, we have made use of simple dense neural networks to approximate complex and multi-dimensional transfer functions in DarkHistory to reduce storage and memory usage, as well as to enable the possibility of adding more parameter dependence to the transfer function. The DNN transfer functions achieve good accuracy in computing the evolution history of matter temperature and ionization, as well as the partial CMB spectral distortion evaluated by the current version of DarkHistory; typical errors are at the few percent level, and are comparable to or smaller than estimates of systematic uncertainties in previous studies of constraints on energy injection [9,10,16].

The DNN-based functionality is available in the DarkHistory Github repository at https://github.com/hongwanliu/DarkHistory, and the necessary data files (one can choose to download the large tables, or DNN and auxiliary files, or both) are hosted on Zenodo [26] (see the Github repository for details).

While the use of DNNs offers one solution to this challenge, there may well be other viable solutions. The information in the DNNs still seems likely to be an overrepresentation of the piecewise-smooth transfer functions. We have briefly explored some alternative methods, including fitting to conventional functions directly, and with the assistance of symbolic regression techniques [30]; however, DNN networks stand out as the best solution (so far) in terms of fitting accuracy and ease of implementation. There is work ongoing to expand the capabilities of DarkHistory, and we look forward to exploring NN-based and alternative techniques in this context.

The code for DarkHistory with DNN transfer functions is available at [31]. The associated data files can be downloaded on Zenodo at [26].

ACKNOWLEDGMENTS

It is a pleasure to thank Hongwan Liu, Wenzer Qin, Gregory Ridgway, Jesse Thaler, Ken van Tilburg, Kathrin Nippel, Nils Schoeneberg, Anastasia Fialkov, Stefan Heimersheim, Julian Muñoz, Laura Lopez Honorez, Vivian Poulin, and Yacine Ali-Haimoud for useful conversations pertaining to this work. Y. S. and T. R. S. were supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics of U.S. Department of Energy under grant Contract No. DE-SC0012567 through the Center for Theoretical Physics at MIT, and the National Science Foundation under Cooperative Agreement No. PHY-2019786 (The NSF AI Institute for Artificial Intelligence and Fundamental Interactions, [32]). The computations in this paper were run on MIT LNS's Erebus machine and the FASRC Cannon cluster supported by the FAS Division of Science Research Computing Group at Harvard University. This research made use of the IPython [33], Jupyter [34], Matplotlib [35], NumPy [36], TensorFlow [24], Keras [25], and SciPy [37] software packages.

APPENDIX

1. Redshift step coarsening and energy conservation

In this appendix, we describe how the photon transfer function change with coarsened redshift step, (expanding on Sec. III E 3 of Ref. [16]), and how energy conservation while correctly accounting for photon energy loss due to redshift is implemented.

a. Transfer functions without coarsening

In DarkHistory's `main.evolve` function, evolution is discretized into log-normal redshift steps with fixed spacing d (`dlnz` in code) where the next redshift z' is expressed in terms of z such that:

$$\log(1+z') = \log(1+z) - d. \tag{A1}$$

Following DarkHistory's flow described in Fig. 2, to obtain the propagating photon spectrum $\mathbf{N}^{\gamma}_{\mathrm{prop}}$, low energy photon spectrum $\mathbf{N}^{\gamma}_{\mathrm{low}}$, low energy electron spectrum $\mathbf{N}^{e}_{\mathrm{low}}$ and energy deposition array E_c^{high}, we retrieve the corresponding transfer functions $\bar{\mathbf{P}}_{\gamma}$, $\bar{\mathbf{D}}_{\gamma}$, $\bar{\mathbf{D}}_{e}$, and $\bar{\mathbf{D}}_c^{\mathrm{high}}$ at a middle redshift z_{mid} given by

$$\log(1+z_{\mathrm{mid}}) = \log(1+z) - d/2. \tag{A2}$$

The corresponding transfer functions $\bar{\mathbf{P}}_{\gamma}$, $\bar{\mathbf{D}}_{\gamma}$, and $\bar{\mathbf{D}}_{e}$ all take in the propagating photon spectrum $\mathbf{N}^{\gamma}_{\mathrm{prop}}$ at redshift z and produce secondary spectra at z'. ($\bar{\mathbf{D}}_c^{\mathrm{high}}$ produces energy deposition in this redshift step.) They are implemented as:

$$\begin{aligned}
\mathbf{N}^{\gamma}_{\mathrm{prop}}(z') &= \mathbf{N}^{\gamma}_{\mathrm{prop}}(z)\cdot\bar{\mathbf{P}}_{\gamma}(z_{\mathrm{mid}}) = \sum_j \mathbf{N}^{\gamma}_{\mathrm{prop}\,j}\bar{\mathbf{P}}_{\gamma\,ji}\\
\mathbf{N}^{\gamma}_{\mathrm{low}}(z') &= \mathbf{N}^{\gamma}_{\mathrm{prop}}(z)\cdot\bar{\mathbf{D}}_{\gamma}(z_{\mathrm{mid}})\\
\mathbf{N}^{e}_{\mathrm{low}}(z') &= \mathbf{N}^{\gamma}_{\mathrm{prop}}(z)\cdot\bar{\mathbf{D}}_{e}(z_{\mathrm{mid}})\\
E_c^{\mathrm{high}}(z') &= \mathbf{N}^{\gamma}_{\mathrm{prop}}(z)\cdot\bar{\mathbf{D}}_c^{\mathrm{high}}(z_{\mathrm{mid}}) = \sum_j \mathbf{N}^{\gamma}_{\mathrm{prop}\,j}\bar{\mathbf{D}}_c^{\mathrm{high}}{}_j,
\end{aligned} \tag{A3}$$

where i and j are indices of discretized energy abscissa.

Energy conservation can be enforced straightforwardly: for an injection in any photon energy bin with central energy $\mathcal{E}_i$, the total output energy on the right hand side of the above equations should add up to $\mathcal{E}_i$ minus the loss to redshifting of the propagating photons. Photons below a certain energy E_{relevant} do not contribute to redshift energy loss because they either dump all of their energies efficiently within one redshift step or free stream and no longer interact, in which case they are stored as an array history of low energy photons $\mathbf{N}^{\gamma}_{\mathrm{low}}(z')$ and not immediately redshifted.

For the propagating photons that are redshifted by $\bar{\mathbf{P}}_{\gamma}$, the energy lost is approximately:

$$|E_{\mathrm{redshift},i}| = \left(1-\frac{1+z'}{1+z}\right)\mathcal{E}_i = \frac{z-z'}{1+z}\mathcal{E}_i \approx d\mathcal{E}_i \quad \text{for } \mathcal{E}_i > E_{\mathrm{relevant}}. \tag{A4}$$

Let the energy abscissa (log-central energies of each bin) for photons and electron be $\mathcal{E}^{\gamma}{}_i$ and $\mathcal{E}^{e}{}_i$ respectively. We can express the above equation as

$$|E_{\mathrm{redshift},i}| = d_i\mathcal{E}_i, \tag{A5}$$

where $d_i = d$ when the photon with energy $\mathcal{E}_i$ should be redshifted and $d_i = 0$ otherwise. (This renders d_i a function of redshift and hydrogen and helium ionization levels in general.)

Then the energy conservation constraint can be written as

$$\mathcal{E}_i = \sum_j \left(\bar{\mathbf{P}}_{\gamma ij}\mathcal{E}^{\gamma}{}_j + \bar{\mathbf{D}}_{\gamma ij}\mathcal{E}^{\gamma}{}_j + \bar{\mathbf{D}}_{eij}\mathcal{E}^{e}{}_j + \bar{\mathbf{D}}_c^{\mathrm{high}}{}_i + d_i\mathcal{E}_i\right) \tag{A6}$$

This relation is imposed by shifting the near diagonal (propagating) part of the high energy photon transfer function $\bar{\mathbf{P}}_{\gamma}$. Any energy nonconservation due to numerical errors or errors from approximating the transfer function as DNNs can be absorbed into this shift.

b. Coarsening

DarkHistory can enlarge the redshift step d to an multiple of a preset step d_0. Let the coarsening multiple be c, then the next redshift step z' to z is such that

$$\log(1+z') = \log(1+z) - c\cdot d_0. \tag{A7}$$

The photon and electron transfer functions are built with log redshift step d_0, so we have to reconstruct the transfer functions for d. We first obtain single-step transfer functions (and d_i) with current ionization levels and a redshift value z_{mid} in the middle of the large redshift step d:

$$\log(1+z_{\text{mid}}) = \log(1+z) - c\cdot d_0/2. \tag{A8}$$

Then, we approximate the large redshift step as consisting of c single redshift steps, each with the same transfer function applied. The compounded transfer functions can be expressed as

$$\begin{aligned}
\mathbf{N}^\gamma_{\text{prop}}(z') &= \mathbf{N}^\gamma_{\text{prop}}(z)\bar{\mathbf{P}}_\gamma{}^c\\
\mathbf{N}^\gamma_{\text{low}}(z') &= \mathbf{N}^\gamma_{\text{prop}}(z)(1+\bar{\mathbf{P}}_\gamma+\bar{\mathbf{P}}_\gamma{}^2+\cdots+\bar{\mathbf{P}}_\gamma{}^{c-1})\bar{\mathbf{D}}_\gamma\\
\mathbf{N}^e_{\text{low}}(z') &= \mathbf{N}^\gamma_{\text{prop}}(z)(1+\bar{\mathbf{P}}_\gamma+\bar{\mathbf{P}}_\gamma{}^2+\cdots+\bar{\mathbf{P}}_\gamma{}^{c-1})\bar{\mathbf{D}}_e\\
E^{\text{high}}_c(z') &= \mathbf{N}^\gamma_{\text{prop}}(z)(1+\bar{\mathbf{P}}_\gamma+\bar{\mathbf{P}}_\gamma{}^2+\cdots+\bar{\mathbf{P}}_\gamma{}^{c-1})\bar{\mathbf{D}}^{\text{high}}_c,
\end{aligned} \tag{A9}$$

with all transfer function evaluated at z_{mid}. In the DNN implementation, the compounded transfer functions $\bar{\mathbf{P}}_\gamma{}^c$, $(1+\bar{\mathbf{P}}_\gamma+\bar{\mathbf{P}}_\gamma{}^2+\cdots+\bar{\mathbf{P}}_\gamma{}^{c-1})$, and $\bar{\mathbf{D}}_e$ are learned as DNN networks, and this step can be carried out without using the value of $\bar{\mathbf{P}}_\gamma$ itself. (The low energy photon transfer function $\bar{\mathbf{D}}_\gamma$ can be quickly reconstructed using the CMB energy loss information.)

Imposing energy conservation is similar: At each d_0 step, the redshift energy loss is $\mathbf{N}^\gamma_{\text{prop}}(z+nd_0)\cdot d_{0i}\mathcal{E}_i$, where $\mathbf{N}^\gamma_{\text{prop}}(z+nd_0) = \mathbf{N}^\gamma_{\text{prop}}(z)\cdot\bar{\mathbf{P}}_\gamma{}^n$. So the total redshift energy loss for a $\mathcal{E}_i$ photon is

$$|E_{\text{redshift},i}| = [(1+\bar{\mathbf{P}}_\gamma+\bar{\mathbf{P}}_\gamma{}^2+\cdots+\bar{\mathbf{P}}_\gamma{}^{c-1})d_0\mathcal{E}]_i. \tag{A10}$$

Let $\bar{\mathbf{S}}_{\gamma,c} = 1+\bar{\mathbf{P}}_\gamma+\bar{\mathbf{P}}_\gamma{}^2+\cdots+\bar{\mathbf{P}}_\gamma{}^{c-1}$. Then the energy conservation condition is

$$\mathcal{E}_i = \bar{\mathbf{P}}_\gamma{}^c_{ij}\mathcal{E}^\gamma{}_j + \bar{\mathbf{S}}_{\gamma,c}(\bar{\mathbf{D}}_{\gamma ij}\mathcal{E}^\gamma{}_j + \bar{\mathbf{D}}_{eij}\mathcal{E}^e{}_j + \bar{\mathbf{D}}^{\text{high}}_c{}_i + d_i\mathcal{E}_i) \quad \forall\, i,\ \text{sum on } j. \tag{A11}$$

Again, the propagating photon spectrum can be adjusted to account for energy nonconservation from numerical and DNN prediction errors.

2. Performances in other DM injection scenarios

Performance metrics of DarkHistory using DNN transfer functions are further demonstrated in Fig. 8 (DM decaying to photon pairs), Fig. 9, (DM annihilating to electron/positron pairs with s-wave cross section), and Fig. 10 (DM annihilating to photon pairs with s-wave cross section).

3. Performances and prediction time of individual DNN transfer functions

The prediction accuracy and prediction time of individual DNNs are summarized in Table I.

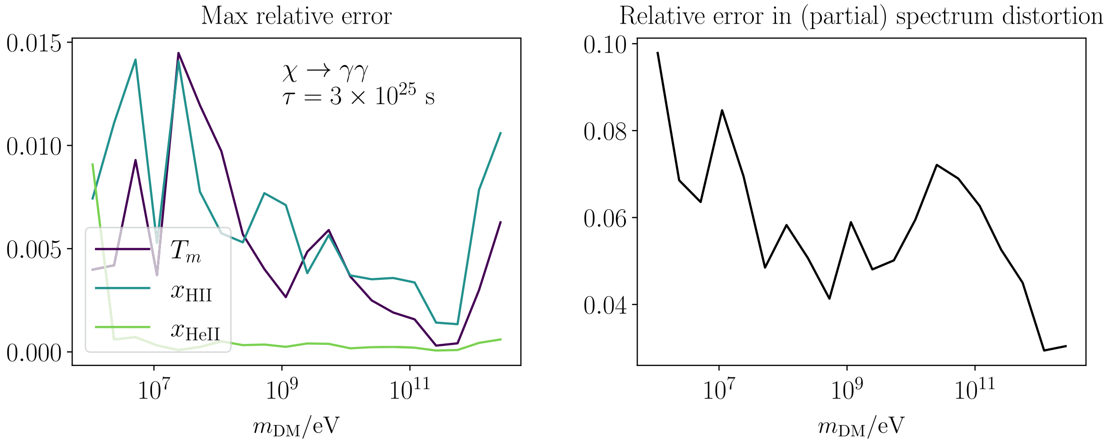


FIG. 8. Relative errors in physical quantities across a range of DM mass, in a scenario with DM decaying to a pair of photons. The panels have similar construction to Fig. 6. Note that the matter temperature and ionization fractions have max relative errors below 2%, and the errors in the low-energy photon spectral distortion are also below 10%.

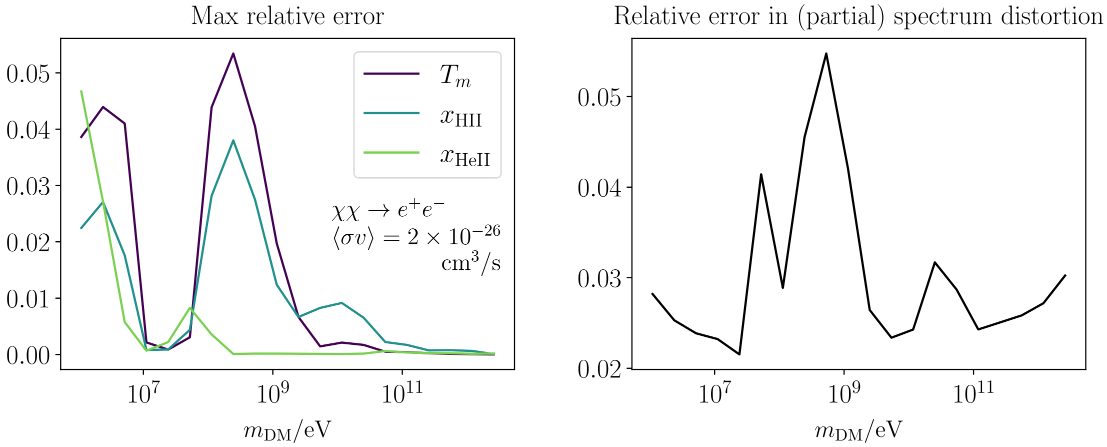


FIG. 9. Relative errors in physical quantities across a range of DM mass in a scenario where DM undergoes s-wave annihilation to a electron/positron pair. The panels have similar construction to Fig. 6.

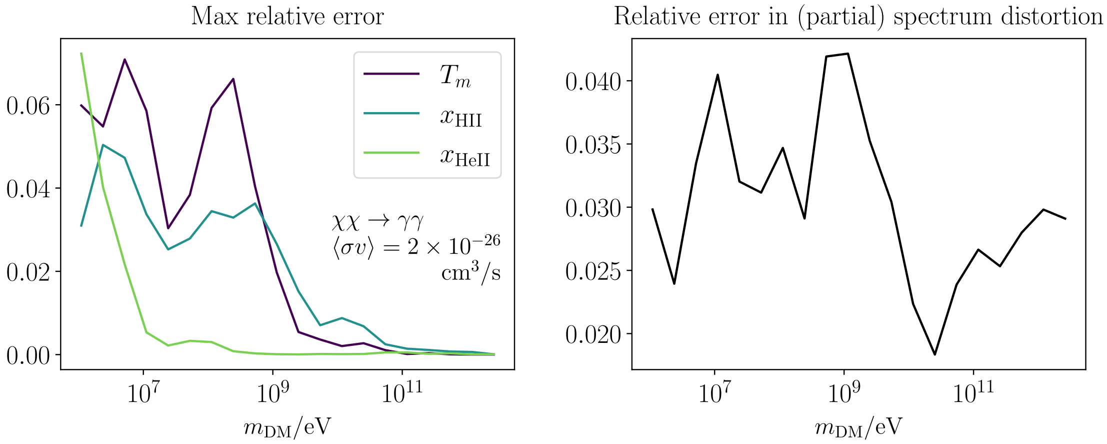


FIG. 10. Relative errors in physical quantities across a range of DM mass in a scenario where DM undergoes s-wave annihilation to a photon pair. The panels have similar construction to Fig. 6.

[1] N. Aghanim, Y. Akrami, M. Ashdown, J. Aumont, C. Baccigalupi, M. Ballardini, A. Banday, R. Barreiro, N. Bartolo, S. Basak *et al.*, Astron. Astrophys. **641**, A6 (2020).

[2] J. A. Adams, S. Sarkar, and D. Sciama, Mon. Not. R. Astron. Soc. **301**, 210 (1998).

[3] X. Chen and M. Kamionkowski, Phys. Rev. D **70**, 043502 (2004).

[4] N. Padmanabhan and D. P. Finkbeiner, Phys. Rev. D **72**, 023508 (2005).

[5] L. Zhang, X. Chen, Y.-A. Lei, and Z.-g. Si, Phys. Rev. D **74**, 103519 (2006).

[6] T. R. Slatyer, N. Padmanabhan, and D. P. Finkbeiner, Phys. Rev. D **80**, 043526 (2009).

[7] T. Kanzaki, M. Kawasaki, and K. Nakayama, Prog. Theor. Phys. **123**, 853 (2010).

[8] T. R. Slatyer, Phys. Rev. D **87**, 123513 (2013).

[9] S. Galli, T. R. Slatyer, M. Valdes, and F. Iocco, Phys. Rev. D **88**, 063502 (2013).

[10] T. R. Slatyer, Phys. Rev. D **93**, 023527 (2016).

[11] T. R. Slatyer, Phys. Rev. D **93**, 023521 (2016).

[12] H. Liu, T. R. Slatyer, and J. Zavala, Phys. Rev. D **94**, 063507 (2016).

[13] T. R. Slatyer and C.-L. Wu, Phys. Rev. D **95**, 023010 (2017).

[14] V. Poulin, J. Lesgourgues, and P. D. Serpico, J. Cosmol. Astropart. Phys. 03 (2017) 043.

[15] H. Liu, W. Qin, G. W. Ridgway, and T. R. Slatyer, Phys. Rev. D **104**, 043514 (2021).

[16] H. Liu, G. W. Ridgway, and T. R. Slatyer, Phys. Rev. D **101**, 023530 (2020).

[17] Y. LeCun, Y. Bengio, and G. Hinton, Nature (London) **521**, 436 (2015).

[18] D. Bourilkov, Int. J. Mod. Phys. A **34**, 1930019 (2019).

[19] A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel, A. Aurisano, K. Terao, and T. Wongjirad, Nature (London) **560**, 41 (2018).

[20] K. Albertsson, P. Altoe, D. Anderson, M. Andrews, J. P. A. Espinosa, A. Aurisano, L. Basara, A. Bevan, W. Bhimji, D. Bonacorsi *et al.*, in *Journal of Physics: Conference Series* (IOP Publishing, Bristol, United Kingdom, 2018), Vol. 1085, p. 022008.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, 2016), http://www.deeplearningbook.org.

[22] J. Duchi, E. Hazan, and Y. Singer, J. Mach. Learn. Res. **12**, 2121 (2011).

[23] J. C. Duchi, M. I. Jordan, and H. B. McMahan, *Advances in Neural Information Processing Systems* (Curran Associates, Inc, 2013).

[24] M. Abadi *et al.*, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), software available from https://www.tensorflow.org/.

[25] F. Chollet *et al.*, Keras (2015), https://github.com/fchollet/keras.

[26] Y. Sun and T. R. Slatyer, Compact data for DarkHistory with neural network transfer functions (v1.1), 10.5281/zenodo.6819281.

[27] W. Y. Wong, A. Moss, and D. Scott, Mon. Not. R. Astron. Soc. **386**, 1023 (2008).

[28] A. Lydia and S. Francis, Int. J. Inf. Comput. Sci **6**, 566 (2019).

[29] https://github.com/hongwanliu/DarkHistory/blob/nntf_alt_hep_clean/examples/Example_12_Using_Neural_Network_Transfer_Functions.ipynb.

[30] M. Schmidt and H. Lipson, Science **324**, 81 (2009).

[31] https://github.com/hongwanliu/DarkHistory/releases/tag/v1.1.0.

[32] http://iaifi.org/.

[33] F. Perez and B. E. Granger, Comput. Sci. Eng. **9**, 21 (2007).

[34] T. Kluyver *et al.*, in *Proceedings of the 20th International Conference on Electronic Publishing* (2016), pp. 87–90, 10.3233/978-1-61499-649-1-87.

[35] J. D. Hunter, Comput. Sci. Eng. **9**, 90 (2007).

[36] C. R. Harris *et al.*, Nature (London) **585**, 357 (2020).

[37] P. Virtanen *et al.*, Nat. Methods **17**, 261 (2020).