NEURAL BOLTZMANN EQUATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

The dynamics of particles in the early universe are described by Boltzmann equations, which involve high-dimensional phase-space integrals. Classical approaches use quadrature integration and evolve the system on a fixed momentum grid, which scales poorly to complicated systems and parameter scans, severely limiting the complexity of processes that can be studied. We introduce Neural Boltzmann Equations (NBEs), which combine three coupled concepts to overcome these limitations. First, particle properties are encoded in physics-inspired neural distribution functions, with parameters that can be predicted using neural networks, enabling efficient parameter scans. Second, phase-space integrals are evaluated with Monte Carlo, using importance sampling tools from collider physics. Third, we use the natural gradient method to evolve the system. After demonstrating the individual benefits of NBEs, we use the framework to perform the most numerically precise calculation to date of the effective number of relativistic neutrino degrees of freedom, $N_{\rm eff}$, in the Standard Model.

1 INTRODUCTION

Cosmology is entering a precision era, with ongoing and upcoming experiments pushing the limits on a range of precision observables, such as the Hubble constant, the dark energy equation of state, the tensor-to-scalar ratio imprinted on the cosmic microwave background, and the effective number of relativistic neutrino degrees of freedom. On the theory side, these processes are all governed by Boltzmann equations, which describe the evolution of interacting particle distributions in an expanding universe. Existing Boltzmann solvers make the underlying equations tractable through combinations of momentum discretisation, assumptions about the form of phase-space distributions, and problem-specific reductions of microscopic interaction rates. Parameter scans with conventional solvers are also expensive, often prohibitively so, as the full system must be re-solved for each parameter combination. The upcoming precision targets, together with the growing interest in new physics scenarios, call for Boltzmann solvers that overcome these computational limitations.

We introduce Neural Boltzmann Equations (NBEs), a framework for solving early-universe Boltzmann equations. Our approach focuses on flexibility without extensive process-specific assumptions, and scalability to complex and compute-intensive problems. NBEs combine three concepts:

- Neural distribution functions – a physics-inspired parameterisation that includes equilibrium solutions, works without discretisation, and can be sampled efficiently. The parameters can be predicted with a neural network based on arbitrary conditioning variables, enabling parameter scans with a single distribution function solve.
- Monte Carlo collision integrals – high-dimensional phase-space integrals, computed with importance sampling techniques developed for event generators at colliders.
- Natural gradient training – an efficient method of training neural distribution functions on the Boltzmann equation to high precision.

We discuss the three aspects in detail in Section 4. We then use toy problems to demonstrate the individual benefits of NBEs in Section 5. Finally, we use the framework to perform a precision calculation of the effective number of relativistic neutrino degrees of freedom in the Standard Model, and show that an NBE trained once can perform a full parameter scan for a new physics model.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2 BACKGROUND

Early-universe dynamics. The Boltzmann equation governs the evolution of particle populations in spacetime, coupling microscopic interactions to propagation on a non-trivial background geometry,

$$\mathcal{L}[f_a] = \mathcal{C}_a[f] \,, \tag{1}$$

where $f_a \geq 0$ is the dimensionless distribution function of species a, representing the occupation of its phase space. The exchange symmetry of identical particles distinguishes two particle types: bosons where $f_a > 1$ is possible, and fermions which satisfy $f_a \leq 1$. The Liouville operator of species a, $\mathcal{L}[f_a]$, describes the propagation of particles in curved spacetime, whilst the collision operator, $\mathcal{C}_a[f]$, encodes particle interactions involving all contributing species $f = (f_1, f_2, \dots)$.

We specialise to FLRW spacetime (Friedman, 1922; Lemaitre, 1927; Robertson, 1935; Walker, 1937), which captures the observed expansion, large-scale homogeneity, and isotropy of the universe. Homogeneity and isotropy reduce the dependence of the distribution functions f_a on 3 spatial and 3 momentum variables to a single momentum variable. Expansion is characterised by the scale factor, $a(t)$, and naturally absorbed by working in comoving variables. In terms of the comoving evolution variable $x = am_0$, with m_0 the characteristic energy scale, the Liouville operator is

$$\mathcal{L}[f_a] = Hx\frac{\partial f_a}{\partial x} \,, \qquad H = \frac{x}{m_0}\frac{\dot{a}}{a} \,, \tag{2}$$

with H the comoving Hubble parameter (Hubble, 1929). More generally, any physical quantity $Q^{\rm phys}$ with mass dimension d has a comoving counterpart $Q = (x/m_0)^d Q^{\rm phys}$, and we fully work in comoving quantities from now on. We reserve the symbol y for comoving momenta. This places all relevant evolution near $x = 1$, irrespective of the physical problem.

The collision operator details all processes that populate or deplete the phase space of species a. For each process k, the statistical factor $\Lambda_k[f]$ combines incoming and outgoing distribution functions with Pauli blocking, $(1 - f_b)$, and Bose enhancement, $(1 + f_b)$, factors for outgoing fermions or bosons. The process contribution is weighted by a squared matrix element $|\mathcal{M}_k|^2$ quantifying the interaction probability, and integrated over the phase space of all particles in the process k except particle a, denoted as $k \setminus a$ in $\mathrm{d}\Phi_{k\setminus a}$, using the phase space measure $\mathrm{d}\Phi_k$ of Eq. (4),

$$\mathcal{C}_a[f] = \frac{1}{2E_a}\sum_k \int \mathrm{d}\Phi_{k\setminus a}\, |\mathcal{M}_k|^2 \Lambda_k[f] \,, \quad \Lambda_k[f] = \prod_{b\in k_{\rm out}} f_b \prod_{b\in k_{\rm in}} (1 \mp f_b) - (\text{in} \leftrightarrow \text{out}) \,. \tag{3}$$

Here $E_a = (m_a^2 + y^2)^{1/2}$ and m_a are the particle energy and mass, respectively. Thermal equilibrium is characterised by detailed balance, $\Lambda_k[f] = 0$ for each process and phase-space point, implying $\mathcal{C}_a[f] = \mathcal{L}[f_a] = 0$. Solving $\Lambda_k[f] = 0$ yields the equilibrium solutions $f_{\rm eq}(E) = [\exp((E-\mu)/T) \pm 1]^{-1}$, the Fermi-Dirac (+) and Bose-Einstein (−) distributions, parameterised by the temperature T and chemical potential μ. Macroscopic observables such as number density n_a, energy density ρ_a, and pressure P_a are moments of the distribution function: $M_a[\phi] = \int \mathrm{d}^3y/(2\pi)^3 \phi(y) f_a(y)$. Specifically, $n_a = M_a[1]$, $\rho_a = M_a[E]$, and $P_a = M_a[y^2/(3E)]$.

Phase-space integrals. Predictions in high-energy physics, such as scattering rates at colliders, decay widths of unstable particles, or reaction rates in the early universe, require integration over the kinematics of all particles. The central object is the Lorentz-invariant phase-space measure

$$\mathrm{d}\Phi_k = \left(\prod_{a\in k} \mathrm{d}\Pi_a\right)(2\pi)^4\delta^4\left(\sum_{a\in k_{\rm in}} y_a - \sum_{a\in k_{\rm out}} y_a\right) \,, \qquad \mathrm{d}\Pi_a = \frac{\mathrm{d}^3 y_a}{(2\pi)^3 2E_a} \,. \tag{4}$$

The one-particle Lorentz-invariant phase-space measure, $\mathrm{d}\Pi_a$, is included for every particle that is integrated over, and the delta distribution, $\delta^4(\,\cdot\,)$ enforces conservation of both energy, E_a, and spatial momentum, $\vec{y}_a$, via the four-momentum $y_a = (E_a, \vec{y}_a)$. This leaves a $(3n-4)$-dimensional integral for n integrated particles. For $n > 2$, Monte Carlo integration is the only efficient integration method. After mapping integration variables to the unit hypercube $u \in [0,1]^{3n-4}$, the integrals read

$$I = \int \mathrm{d}u\, h(u) = \mathbb{E}_{u\sim p(u)}\left[\frac{h(u)}{p(u)}\right] \,, \qquad \sigma_I^2 = \frac{1}{N}\mathrm{Var}_{u\sim p(u)}\left[\frac{h(u)}{p(u)}\right] \,. \tag{5}$$

The proposal distribution, $p(u)$, should be chosen to approximate the normalised integrand. The variance, σ_I^2, measures the approximation quality, and vanishes for $p(u) = h(u)/I$, $h(u) \geq 0$.

High-energy physics event generators use several techniques to improve the quality of the proposal distribution, $p(u)$ (Alwall et al., 2011; Bothmann et al., 2019). Process-specific phase-space mappings map the unit hypercube to the space of valid momentum configurations, accounting for momentum conservation and the integrand shape. More recently, normalising flows and other generative networks have been used to further refine the proposal distribution (Heimel et al., 2023).

Natural gradient optimisation. The natural gradient method makes use of the function space geometry to enable optimal parameter updates (Amari & Nagaoka, 2000). This method is becoming increasingly popular in Variational Monte Carlo (Stokes et al., 2020) and physics-informed neural networks (Müller & Zeinhofer, 2023). Consider the problem of fitting the function $g(y;\theta)$, with $\theta \in \mathbb{R}^N$, to the target function $g_{\text{target}}(y)$, with residuals weighted by the factor $W = 1/g(\theta)$,

$$\theta_* = \text{argmin}_\theta L(\theta) \,, \qquad L(\theta) = r^\top W r \,, \qquad r = g(\theta) - g_{\text{target}} \,. \tag{6}$$

We sample B values of y from a density $p(y)$, where we evaluate g, g_{target} and p, and build the vectors $g(\theta), g_{\text{target}}, p \in \mathbb{R}^B$. The density factor p enters the weighting matrix $W = \text{diag}(1/(p\, g(\theta)))$. To solve the problem, we linearise the system as $g(\theta) = g(\theta_0) + J(\theta - \theta_0)$ with Jacobian $J = \partial g / \partial \theta$. The optimal parameter update is then given by the inverse of the Fisher matrix $F \in \mathbb{R}^{N\times N}$

$$\theta_* = \theta_0 - F^{-1} J^\top W r_0 \,, \qquad F = J^\top W J \,, \qquad r_0 = g(\theta_0) - g_{\text{target}} \,. \tag{7}$$

Standard gradient descent neglects the function space geometry and sets $F \propto 1$. For small networks, $N \leq B$, we invert F directly, whereas for large networks, $N > B$, we follow Chen & Heyl (2024) and rewrite $(J^\top W J)^{-1} J^\top W = J^\top (J J^\top)^{-1}$, then invert the Gram matrix $J J^\top \in \mathbb{R}^{B\times B}$ instead.

3 Related work

Boltzmann equation solvers. Existing solvers are typically tailored to a specific problem, each with its own analytic reductions and numerical approximations to make its system tractable. For neutrino decoupling, FortEPiaNO evolves neutrino density matrices on a fixed Gauss-Laguerre momentum grid with analytically reduced collision integrals (Gariazzo et al., 2019). Neutrino direct simulation Monte Carlo (Ovchynnikov & Syvolap, 2025a;b) instead tracks individual neutrinos and simulates interactions with Monte Carlo. For the cosmic microwave background (CMB), CAMB and CLASS solve the Einstein-Boltzmann hierarchy on a fixed background, predicting CMB anisotropies and the matter power spectrum (Lewis et al., 2000; Blas et al., 2011). For dark matter, micrOMEGAs computes relic abundances for generic models using momentum-integrated Boltzmann equations, where thermally averaged rates replace collision operators (Bélanger et al., 2018; Alguero et al., 2024). DRAKE goes beyond kinetic equilibrium and solves for the distribution function of non-relativistic, dilute dark matter in an equilibrium background (Binder et al., 2021). ULYSSES specialises to leptogenesis and primarily solves for moments of the distribution function in an equilibrium Standard Model bath with analytically reduced collision integrals (Granelli et al., 2021; 2026).

NBEs address two bottlenecks of existing Boltzmann solvers. First, moment-based solvers discard phase-space information, while momentum-grid solvers scale steeply with grid size, *e.g.* cubically for $2 \leftrightarrow 2$ processes. Parameter scans multiply this cost, since each mass or coupling requires a full solve. NBEs instead parameterise distribution functions with the physics-inspired neural distribution function ansatz, and conditioning on masses or couplings allows one model to learn a family of solutions across parameter space. Second, beyond $2 \leftrightarrow 2$ processes, collision integrals become prohibitively expensive with standard quadrature methods: every additional particle adds a three-dimensional momentum integral, so that analytic reductions and grid choices must be re-derived for each new process. NBEs instead use Monte Carlo methods, which scale well to complex processes and require only the matrix element for a new process, with no fixed grids or analytic reductions.

Neural quantum states. A neural network serves as a variational ansatz for the state of a quantum system, trained directly on the equations governing the system rather than data. In the original formulation for quantum lattice models, a network parameterises the many-body wave function $\psi(s;\theta)$ for discrete lattice configurations s, and the parameters θ are optimised using the variational principle (Carleo & Troyer, 2017). Neural wave functions (Pfau et al., 2020; Hermann et al., 2020) extend this idea to electrons in continuous space, solving the electronic Schrödinger equation of atoms and molecules from first principles. Neural distribution functions do the same for kinetic theory: a network parameterises phase-space distribution functions and is trained on the residual of the Boltzmann equation, which plays the role of the Hamiltonian in the examples above.

4 METHODS

Neural Boltzmann Equations combine three ingredients. First, we construct a physics-inspired ansatz for bosonic and fermionic distribution functions that generalises the known equilibrium solutions. Second, we efficiently evaluate phase-space integrals using Monte Carlo integration with importance sampling. Third, we solve the Boltzmann equation sequentially, using natural gradient updates at each time step. Additional details are given in Appendix A.

4.1 NEURAL DISTRIBUTION FUNCTIONS

Thermodynamic distribution functions detail the mean occupation at each phase-space point. Motivated by this property, we factorise distribution functions into the invariant density η, which sets the overall scale, and the normalised density, $p(y)$, encoding their momentum distribution:

$$f(y) = \frac{4\pi^2 E}{y^2}\eta p(y) \;, \qquad \eta = \int \mathrm{d}y \, \frac{y^2}{4\pi^2 E} f(y) \;, \qquad \int \mathrm{d}y \, p(y) = 1 \;. \tag{8}$$

The parameterisation uses the invariant density η instead of the number density, n, because it naturally appears in phase-space integrals, making $p(y)$ the optimal importance sampling distribution of the Lorentz-invariant phase space, see Appendix A.2. The deterministic prefactor $4\pi^2 E/y^2$ ensures that the second equation above holds. We parameterise the normalised density with a mixture of gamma distributions, $\gamma(\varepsilon|\alpha,\lambda)$, in terms of the rescaled kinetic energy $\varepsilon = (\sqrt{y^2+m^2}-m)/T$ as

$$p(y) = \frac{y}{ET}\sum_{m=1}^{M} w_m \gamma\left(\frac{\sqrt{y^2+m^2}-m}{T}\Big|\alpha_m,\lambda_m\right) \;, \qquad \gamma(\varepsilon|\alpha,\lambda) = \frac{\lambda}{\Gamma(\alpha)}(\lambda\varepsilon)^{\alpha-1}e^{-\lambda\varepsilon} \;. \tag{9}$$

With sufficiently many mixture components M, this parameterisation can represent the massless bosonic equilibrium distribution to arbitrary precision, see Appendix A.1. Fermionic distribution functions, which must satisfy the Pauli exclusion principle, $f(y) \leq 1$, are constructed from an unconstrained bosonic $f_0(y)$,

$$f(y) = \frac{f_0(y)}{1+f_0(y)} \in [0,1] \;, \qquad p(y) = \frac{1}{N}\frac{1}{1+f_0(y)}p_0(y) \;, \qquad N = \int \mathrm{d}y \, \frac{1}{1+f_0(y)}p_0(y) \;. \tag{10}$$

The Fermi-Dirac distribution corresponds to $f_0(y) = e^{-(E-\mu)/T}$, a single mixture component. We can sample from the normalised densities, $p(y)$, for both bosons and fermions, although the fermionic case requires an additional rejection sampling step with typical acceptance rate 0.8, see Appendix A.1. Moments of $f(y)$, such as number and energy density, can be evaluated efficiently using Gauss-Laguerre or Gauss-Hermite quadrature. See Appendix A.1 for additional information.

Together, the invariant density, η, and gamma mixture parameters, $(w_m,\alpha_m,\lambda_m)_{m=1}^M$, fully specify the parameterisation ω, which has $3M$ independent parameters due to the redundancy in the normalisation of the mixture weights. Depending on the application, ω is either optimised directly or predicted from theory conditions, ξ, by a neural network with weights θ,

$$\omega = \left\{\eta, (w_m,\alpha_m,\lambda_m)_{m=1}^M\right\} = \mathrm{NN}(\xi;\theta) \;. \tag{11}$$

We write $f(y;\omega)$ and $p(y|\omega)$ to highlight the dependence on the parameters ω, and use $M = 50$ throughout. If predicted with a neural network, the parameterisation is cached and reused after the initial evaluation. The same parameterisation describes both unconstrained and equilibrium distribution functions, where only temperature and chemical potential are kept as free parameters.

4.2 COLLISION OPERATOR

The collision operator, $\mathcal{C}_a[f]$, quantifies the net change in particle occupations due to microscopic interactions. We give the explicit expression for $\mathcal{C}_a[f]$ in Eq. (3), and the phase-space measure $\mathrm{d}\Phi_k$ in Eq. (4). Evaluating this integral is typically the compute bottleneck in the NBE framework.

To evaluate the collision operator with Monte Carlo integration, we first generate phase-space points following Algorithm 1. The initial-state angles, $\hat{\Omega}_k$, are sampled uniformly on the sphere and the

Algorithm 1 Phase-space generation for a process with m incoming and $n-m$ outgoing particles.

Require: observed momentum y_1, condition ξ, masses $m_1, \dots, m_n$, proposal densities $p_2, \dots, p_m$

$y_k \sim p_k(\cdot \mid \xi)$, $p_k \leftarrow p_k(y_k|\xi)$ for $k = 2, \dots, m$ ▷ sample initial momenta

$\hat{\Omega}_k \sim \mathcal{U}(S^2)$ for $k = 1, \dots, m$ ▷ sample initial angles

$u_{\text{out}} \sim \mathcal{U}\big([0,1]^{3(n-m)-4}\big)$ ▷ sample outgoing configuration

if neural importance sampling **then**

  $u,\ J_{\text{enc}} \leftarrow \text{ToUnitCube}(y_2, \dots, y_m, \hat{\Omega}_2, \dots, \hat{\Omega}_m, u_{\text{out}})$

  $u,\ J_{\text{flow}} \leftarrow \text{Flow}(u \mid y_1, \xi)$ ▷ MadNIS refinement

  $(y_2, \dots, y_m, \hat{\Omega}_2, \dots, \hat{\Omega}_m,\ u_{\text{out}}),\ J_{\text{dec}} \leftarrow \text{ToUnitCube}^{-1}(u)$

else

  $J_{\text{enc}},\ J_{\text{flow}},\ J_{\text{dec}} \leftarrow 1$

$\vec{y}_k \leftarrow y_k \hat{\Omega}_k$ for $k = 1, \dots, m$ ▷ build initial state

$\sqrt{s} \leftarrow \left[(\sum_{k \le m} E_k)^2 - |\sum_{k \le m} \vec{y}_k|^2\right]^{1/2}$ with $E_k = (y_k^2 + m_k^2)^{1/2}$

$\vec{y}_{m+1}, \dots, \vec{y}_n,\ J_{\text{ps}} \leftarrow \text{MadSpace}(u_{\text{out}};\ \sqrt{s},\ m_{m+1}, \dots, m_n)$ ▷ build final state

$w_{\text{ps}} \leftarrow J_{\text{enc}}\, J_{\text{flow}}\, J_{\text{dec}}\, J_{\text{ps}} \prod_{k=2}^{m} y_k^2/(4\pi^2 E_k p_k)$ ▷ importance weight

return $(\vec{y}_1, \dots, \vec{y}_n;\ w_{\text{ps}})$

momenta, y_k, from the neural distribution function densities, p_k. For the final state, we sample u_{out} uniformly from the unit hypercube and use MadSpace (Heimel et al., 2026) to map it onto valid phase-space configurations, with the initial-state information entering through the centre-of-mass energy, $\sqrt{s}$. We use neural importance sampling to further refine the phase-space generation with a normalising flow. The deterministic ToUnitCube transform maps all generated variables to the unit hypercube, where we apply the learned flow transform. We use the MadNIS implementation for the normalising flow (Heimel et al., 2024; 2026), conditioned on the fixed momentum, y_1, parameters, ξ, and trained to minimise the variance of the integral, see Eq. (15). The framework also accommodates advanced event-generation techniques such as topology-aware phase-space mappings and multi-channel Monte Carlo, though we do not need them for the examples in this work.

Phase-space generation returns the importance weight, w_{ps}, and the momentum configurations, $\vec{y}_1, \dots, \vec{y}_n$, which determine the matrix element, $|\mathcal{M}_k|^2$, and the gain-loss combination, $\Lambda_k[f]$. Summing over processes yields the collision operator

$$\mathcal{C}_a[f](y_1) = \frac{1}{2E_a} \sum_k \int \mathrm{d}\Phi_{k\setminus a} |\mathcal{M}_k|^2 \Lambda_k[f] = \frac{1}{2E_a} \sum_k \mathbb{E}_{\hat{\Omega}_1, \vec{y}_2, \dots, \vec{y}_n} \left[w_{\text{ps}} |\mathcal{M}_k|^2 \Lambda_k[f] \right] . \quad (12)$$

Precise evaluations of the collision integral demand large sample counts, and so dominate the NBE compute cost. We mitigate this by parallelising across particles and processes, and using the optimised MadSpace library. As the training procedure in Section 4.3 passes no gradients through the collision integral, we are free to include other event-generation tools if required.

4.3 Boltzmann equation solver

We now discuss the training of neural distribution functions to satisfy the Boltzmann equation. Multiplying Eq. (1) by y^2/H, writing $g(y;\theta) = y^2 f(y;\theta)$ for the fit function, $s = y^2\mathcal{C}/H$ for the source term, and dropping the particle index a, the Boltzmann equation reads

$$s(g) = \frac{\partial g}{\partial \log x} \simeq \frac{g_{n+1} - A}{\gamma h} , \quad (13)$$

where in the rightmost expression we discretised the derivative with step size $h = \Delta \log x$, and defined $g_m(y) \equiv g(y;\theta_m)$ for the fit function at step m. The discretisation scheme (A, γ) combines $A = A(g_n, g_{n-1}, \dots)$, which depends on the previous solutions, with a step-size weight γ. The standard difference quotient is $(A, \gamma) = (g_n, 1)$. We use a multi-step scheme, see Appendix A.3.

To solve Eq. (13), we consider the implicit equation $0 = F(g_{n+1}) = g_{n+1} - A - \gamma h s(g_{n+1})$, solved by iterating $g'_{n+1} = T(g_{n+1}) = g_{n+1} - M^{-1} F(g_{n+1})$ with an arbitrary function M. The choice $M = \partial F/\partial g = 1 - \gamma h\, \partial s/\partial g$ converges fastest, but requires a costly collision operator derivative. We instead use $M = 1 + \gamma h \kappa$, with $\kappa \ge 0$ a cheap approximation of $-\partial s/\partial g$ that exploits the

Algorithm 2 Boltzmann solver step $\log x_n \to \log x_{n+1}$ fixed-point iteration

Require: condition prior $p_{\text{prior}}(\xi)$, frozen previous solutions $\theta_n, \theta_{n-1}, \ldots$

$\theta_{n+1} \leftarrow \theta_n$ ▷ warm-start from previous step
repeat
  $\xi_i \sim p_{\text{prior}}(\xi)$ for $i = 1, \ldots, B$ ▷ sample conditions
  $\xi \leftarrow (\xi_1, \ldots, \xi_B)$ ▷ batched evaluation
  $\omega_l \leftarrow \text{NN}(\xi; \theta_l)$ for $l = n+1, n, \ldots$ ▷ cache distribution function parameters
  for species $a = 1, \ldots, N$ **do** ▷ update each species independently
    $y \sim p(\,\cdot\,|\omega_{n+1,a})$ ▷ sample test momenta
    $s(y)$, $\kappa(y) \leftarrow$ evaluate collision integral following Section 4.2
    $A(y)$, $\gamma \leftarrow$ evaluate scheme coefficients using Appendix A.3
    $g_{\text{target}}(y) \leftarrow T(g(y; \omega_{n+1,a}); s(y), \kappa(y), A(y), \gamma)$ ▷ fixed-point target Eq. (14)
    $r \leftarrow g(y; \omega_{n+1,a}) - g_{\text{target}}(y)$
    $w \leftarrow \left[p(y|\omega_{n+1,a})\, g(y; \omega_{n+1,a})\right]^{-1}$
    $J \leftarrow \frac{\partial g(y;\omega(\theta))}{\partial \theta}\Big|_{\theta=\theta_{n+1,a}}$ using autodiff
    $W \leftarrow \text{diag}(w)$, $F \leftarrow J^\top W J$
    $\theta_{n+1,a} \leftarrow \theta_{n+1,a} - F^{-1} J^\top W r$ ▷ natural-gradient step Eq. (7)
until objective $r^\top W r$ plateaus or Monte Carlo noise dominates

specific form of the collision operator, see Appendix A.3. In contrast to the naive choice, $M = 1$, this yields an update rule that is invariant under small perturbations of g_{n+1}, making the formalism applicable to stiff regimes that commonly arise whenever species are deep in thermal equilibrium,

$$g'_{n+1} = T(g_{n+1}) = \frac{A + \gamma h(s(g_{n+1}) + \kappa g_{n+1})}{1 + \gamma h \kappa} \,. \tag{14}$$

Algorithm 2 summarises the training for a single time step. Using the previous step parameters as an initial guess, $\theta_{n+1} = \theta_n$, we iterate until the objective plateaus or the feedback is dominated by Monte Carlo noise, see Appendix A.3. Each iteration samples a batch of conditions, ξ, caches the neural distribution function parameters, ω_{n+1}, ω_n, etc., for the required time steps, and updates the parameters $\theta_{n+1,a}$ of each species a independently. After sampling test momenta from the distribution function density, we evaluate the collision integral objects: s, κ, the scheme coefficients, A, and γ. The iteration target follows from Eq. (14), and a natural gradient step moves the parameters $\theta_{n+1,a}$ towards it. We use $\mathcal{O}(10^3)$ test points per particle, and $\mathcal{O}(10^3)$ collision integral evaluations per test point. The procedure converges within 5-10 (10-30) iterations for unconditional (conditional) distribution functions, depending on the sensitivity to the condition. To enforce the initial condition $g(y; \theta_0) = g_{\text{init}}(y)$, we use the same iterative algorithm with deterministic target $g_{\text{target}}(y) = g_{\text{init}}(y)$, which typically converges within 50 iterations. See Appendix A.3 for details.

For neural importance sampling, we pretrain the normalising flow on the initial distribution function and evolve it jointly with the neural distribution functions. In each iteration, we use one collision integral sample per (y_1, ξ) pair to update the normalising flow density $q(u|y_1, \xi)$, with $q(u|y_1, \xi)|_{\text{no-grad}}$ denoting terms excluded from the gradient calculation. We train the flow on a variance loss using the Adam optimiser, since natural gradient training gave no significant benefit here:

$$\mathcal{L} = \mathbb{E}_{\xi \sim p_{\text{prior}}(\xi), y_1 \sim p(y_1|\xi), u \sim q(u|y_1,\xi)} \left[\left(\frac{s(u)}{p(y_1|\xi) q(u|y_1,\xi)} \right)^2 \Big|_{\text{no-grad}} \frac{q(u|y_1,\xi)|_{\text{no-grad}}}{q(u|y_1,\xi)} \right] . \tag{15}$$

Computational cost. The collision operator dominates the cost, with typically 10^3 function evaluations per test momentum, whereas the network evaluation, natural gradient update, and MadNIS update do not scale with the collision operator sample count, and only contribute at the 10% level. The training runs in Section 5 take up to 5 hours on an NVIDIA A30 GPU, see Appendix C.

Limitations. NBEs use a parametric ansatz for the distribution function, which may require many components to express exotic shapes. Adding different components to the gamma mixture model may improve efficiency, but requires knowledge of the specific effects. NBEs also assume homogeneous, isotropic FLRW spacetime, reducing distribution function dependence from $f(\vec{y}, \vec{x})$ to $f(y)$.

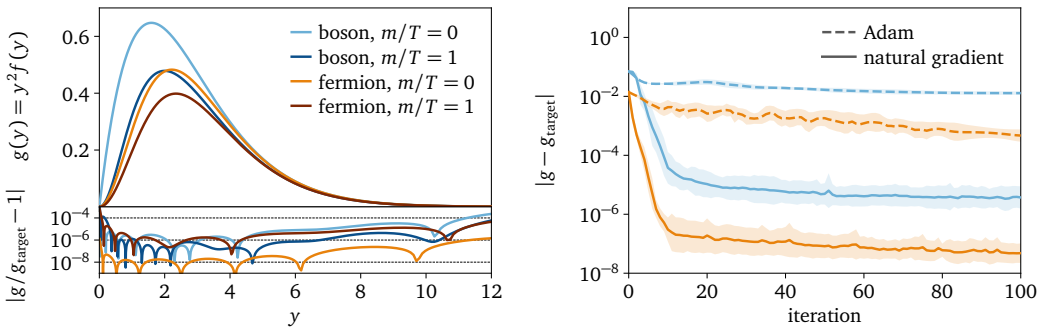

Figure 1: **Natural gradient training for neural distribution functions.** Left: Learned distribution functions for massless and massive bosons and fermions, compared to equilibrium targets. Right: Average residual over training time for massless bosons and fermions, comparing the natural gradient method with the Adam optimiser using the mean and standard deviation of ten independent trainings.

5 Results

Here we showcase the properties of NBEs on toy problems, and then perform a precision calculation of the effective number of relativistic neutrino degrees of freedom in the early universe. See Appendix C for details on the experiments, and Appendix D for additional results.

5.1 Equilibrium distributions

As a first test of the NBE framework, we train neural distribution functions with Algorithm 2 on a fixed target, $g_{\text{target}}(y) = g_{\text{init}}(y)$, i.e. the initialisation before time evolution. The left panel of Figure 1 shows the learned distribution functions for massive and massless bosons and fermions, which match equilibrium to 10^{-6} relative deviation in the bulk. The massless fermion distribution is exactly representable by a single mixture component in our parameterisation and therefore agrees to even higher precision. The precision is limited by the restriction to 50 mixture components, and by regularisation effects in the Fisher matrix inversion.

Second, we compare natural gradient training with standard gradient descent using Adam (Kingma & Ba, 2015) in the right panel of Figure 1. Natural gradient training converges to an average residual of 10^{-6} within 10 iterations, far outperforming Adam at 10^{-2} for the same iteration budget. The Adam residual continues to fall and reaches 10^{-4} after around 10,000 iterations. Although a natural gradient update takes about twice as long as an Adam update in our implementation, it is far more time-efficient overall and reaches a higher final precision thanks to the curvature information.

5.2 Thermalisation

We now consider a single massive scalar particle without cosmic expansion (Yoon, 2026), a toy, strongly interacting massive particle dark matter model (Hochberg et al., 2014). Starting out of equilibrium, the particle thermalises via the processes $\phi\phi \leftrightarrow \phi\phi$ and $\phi\phi \leftrightarrow \phi\phi\phi$ with common coupling strength λ. Whereas the $\phi\phi \leftrightarrow \phi\phi$ collision integral is 2-dimensional, the $\phi\phi \leftrightarrow \phi\phi\phi$ integral is 5-dimensional, motivating Monte Carlo integration.

The left panel of Figure 2 shows the collision operator as a function of the momenta, y, using NBEs, and the Boltzmann Equation Solver for Thermalisation (BEST) (Yoon, 2026), which uses a custom phase-space mapping refined with the VEGAS algorithm (Lepage, 2021). The momentum-averaged Monte Carlo uncertainty, $\bar{\sigma}$, is $20\times$ smaller for NBE than for BEST, corresponding to a $400\times$ more sample-efficient estimate, given that $\bar{\sigma} \propto 1/\sqrt{N_s}$. This gain comes from the optimised importance sampling, namely MadSpace phase-space mappings and distribution function sampling. MadNIS refinement increases the gain to $1000\times$ in total.

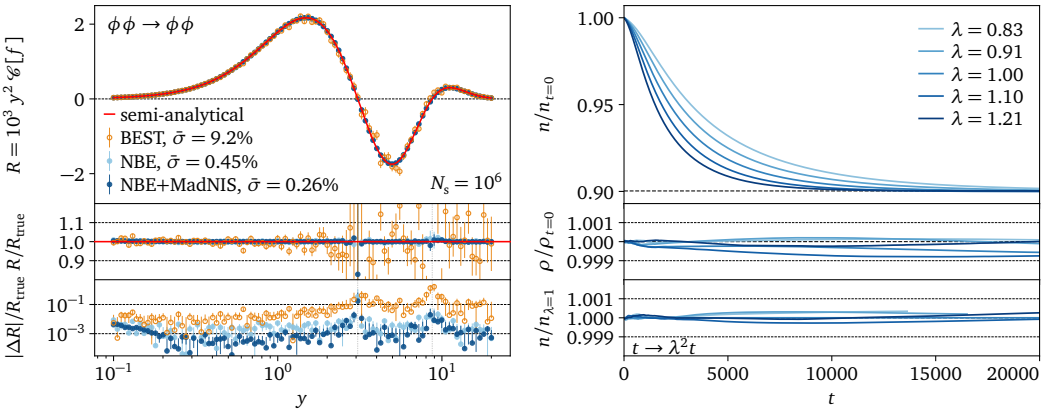

Figure 2: **Efficient collision operator evaluation and coupling-conditioned training.** Left: Collision operator for $\phi\phi \leftrightarrow \phi\phi$, calculated semi-analytically and with Monte Carlo using BEST and NBE, with the momentum-averaged relative Monte Carlo uncertainty, $\bar{\sigma}$, as precision metric. Right: Number density, n_t, and energy density, ρ_t, over time for different couplings, see Yoon (2026) for the BEST result. The lower panel rescales $t \to \lambda^2 t$ to collapse the curves for different couplings.

We next condition an NBE on the coupling $\xi = \lambda$ with a uniform prior and evolve the system in time. The energy density ρ_t is conserved, whilst number-changing processes drive the number density n_t to $n_{t=\infty}/n_{t=0} \approx 0.900$ in equilibrium. The right panel of Figure 2 demonstrates the number density converging to the asymptotic value at a coupling-dependent rate, whilst the energy density remains constant to within 0.1%. Rescaling $t \to \lambda^2 t$ should align all curves, see Appendix C, and NBEs realise this to $< 0.1\%$. Although it must cover the full prior range, this conditional solve is only about $5\times$ slower than a comparable fixed-coupling solve in our implementation.

5.3 Neutrino Decoupling

In the early universe, weak interactions keep neutrinos in equilibrium with the electromagnetic plasma of photons (γ), electrons (e^-), and positrons (e^+) through scattering ($\nu e^\pm \leftrightarrow \nu e^\pm$) and annihilation ($\nu\bar{\nu} \leftrightarrow e^+e^-$) processes. At temperatures of $T \approx 1\,\text{MeV}$, or around one second after the Big Bang, neutrinos lose thermal contact with the electromagnetic plasma due to cosmic expansion, and free stream as the cosmic neutrino background. Shortly afterwards, at $T \approx m_e \approx 0.5\,\text{MeV}$, electrons and positrons annihilate ($e^+e^- \to \gamma\gamma$), reheating photons. The overlap of these processes controls the relative heating of the neutrino and electromagnetic sectors, and therefore the precision observable N_{eff}, the effective number of relativistic neutrino degrees of freedom:

$$N_{\text{eff}} = \frac{8}{7}\left(\frac{11}{4}\right)^{4/3} \frac{\rho_\nu}{\rho_{\text{EM}}}\bigg|_{T/m_e \to 0} . \tag{16}$$

Entropy conservation yields $N_{\text{eff}} = 3$ by construction for instantaneous decoupling. Interactions increase N_{eff} to 3.0440, see Appendix B for analytic expressions and Appendix C for numerical results on the individual effects. We visualise the process in Figure 3: the electromagnetic energy density $\rho_{\text{EM}} = \rho_\gamma + \rho_e + \rho_{\text{QED}}$ increases significantly due to electron-positron annihilation, whereas the neutrino energy densities ρ_ν increase only slightly due to the overlap of neutrino decoupling and electron-positron annihilation. Electron neutrinos experience more heating due to their additional interactions with the electromagnetic bath. The neutrino distribution functions depart from equilibrium by $\sim 1\%$ due to the same interactions.

The most precise calculation to date finds $N_{\text{eff}} = 3.0440 \pm 10^{-4}\,\text{(solver)} \pm 10^{-4}\,\text{(experiment)}$ (Bennett et al., 2021; Froustey et al., 2020; Akita & Yamaguchi, 2020), limited by the solver precision and the measured solar neutrino mixing angle (experiment), both contributing at the 10^{-4} level. We repeat this calculation using the NBE framework and the recent NuFIT-6.1 updates on experimental inputs (Esteban et al., 2024; 2025). Our result agrees, with a $50\times$ smaller solver uncertainty:

$$N_{\text{eff}} = 3.04392 \;\pm 2 \cdot 10^{-6}\,\text{(solver)} \;\pm 2 \cdot 10^{-5}\,\text{(experiment)} . \tag{17}$$

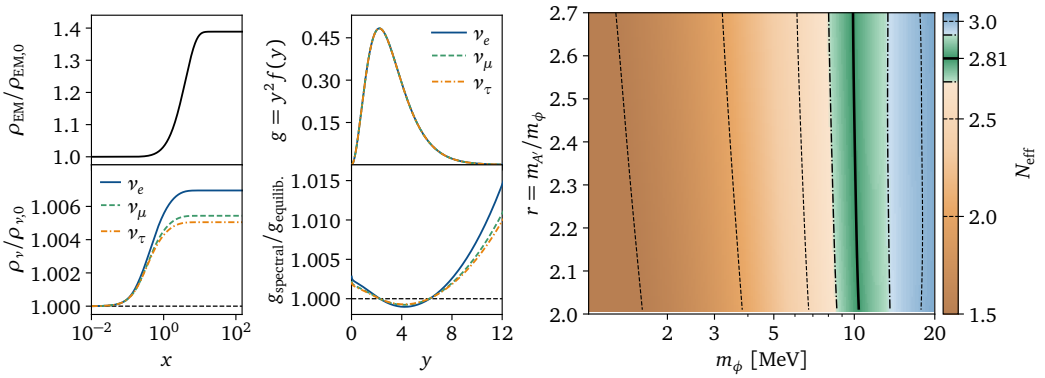

Figure 3: **Neutrino decoupling and new physics impact.** Left: Evolution of electromagnetic and neutrino energy densities during neutrino decoupling. Middle: Neutrino spectral distortions after decoupling, comparing the equilibrium solution to the full solution including spectral distortions. Right: $N_{\rm eff}$ for the new physics model of Escudero et al. (2026b), from a single NBE training.

We discuss the calculation in detail in Appendix C. Besides the now leading experimental uncertainty, there is a theory uncertainty from neglected higher-order effects in the neutrino collision integral. Previous partial calculations found corrections ranging from $7 \cdot 10^{-4}$ (Cielo et al., 2023) to $2 \cdot 10^{-5}$ (Drewes et al., 2024), turning these effects into the dominating uncertainty. This motivates a full calculation of the higher-order corrections to the neutrino collision integral, to keep the uncertainty on the theory prediction at a level that allows to neglect it in any experimental analysis.

On the experimental side, the most recent cosmic microwave background analysis reports $N_{\rm eff} = 2.81 \pm 0.12$ (Louis et al., 2025; Calabrese et al., 2025; Camphuis et al., 2026), with further improvements expected from ongoing experiments (Ade et al., 2019). This 2σ tension with the theory prediction may indicate either missing systematic uncertainties or new physics. For instance, Escudero et al. (2026b) extend the Standard Model by a dark photon, A', kinetically mixed with the photon and decaying invisibly $A' \to \phi\phi$ to a complex scalar ϕ that serves as a dark matter candidate. In a specific mass window, these decays feed the electromagnetic plasma and raise the electromagnetic energy density while evading other cosmological constraints, see Escudero et al. (2026b) for details.

To demonstrate the flexibility of NBEs, we implement the model of Escudero et al. (2026b) as an extension of our calculation, conditioning on the model parameters $\xi = (m_\phi, r)$ and $r = m_{A'}/m_\phi$, with uniform prior. As shown in the right panel of Figure 3, the prediction agrees with the measured $N_{\rm eff} = 2.81 \pm 0.12$ in a band $m_\phi \in [8\,\text{MeV}, 13\,\text{MeV}]$, nearly independent of the mass ratio r.

6 Conclusion

We have introduced Neural Boltzmann Equations (NBEs), a flexible framework for solving Boltzmann equations in the early universe. Collision integrals are evaluated with Monte Carlo integration, allowing the framework to scale comfortably to complex processes. At the heart of NBEs are neural distribution functions: a physics-inspired parameterisation that can be optimised with neural networks conditioned on theory parameters, enabling efficient parameter scans. The natural gradient method allows the training of conditional and unconditional distribution functions to high precision.

NBE importance sampling is $1000\times$ more sample-efficient than a recently proposed Monte Carlo Boltzmann solver, which allowed us to evolve a complex non-equilibrium system with number-changing $2 \leftrightarrow 3$ processes to per-mil precision. We then performed a precision calculation of the effective number of relativistic neutrino degrees of freedom with a $50\times$ smaller solver uncertainty, setting the stage for future calculations including additional higher-order effects. Finally, we used a single conditioned NBE for a two-dimensional parameter scan of an extended $N_{\rm eff}$ scenario motivated by the current tension between experiment and the Standard Model.

Together, these results establish NBEs as a precise, flexible, and scalable framework for early-universe calculations, bringing higher-multiplicity processes and high-dimensional parameter spaces within reach of precision early-universe calculations.

AI USE STATEMENT

In this work, we used generative AI tools to help develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or provide feedback on research methodology or experiments, to implement methods, and to interpret results. We have not used generative AI tools to formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, assist in translation. The use cases of generating synthetic datasets, cleaning and reformatting datasets, and supporting qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to create or modify scientific figures or images, to create or edit software code, to summarise or analyse existing literature, and for brainstorming. We have reviewed all AI-assisted work. We used AI extensively, both to get feedback on ideas and to turn those ideas into efficient implementations. Because AI is increasingly good at assembling brute-force solutions, i.e. combining many small ingredients until a task works, without making clear which of them actually matters, we focused on distilling the minimal set of ingredients the method needs. We verified and tested AI-generated code by working through it and rewriting it to improve readability. We take responsibility for the final content of this work, including text, claims or artefacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe the method in more detail in Appendix A, and give extended information about our experiments in Appendix C. We provide the analytic expressions for the neutrino decoupling calculation in Appendix B for completeness, even though they are also available in the literature. We attached our implementation in the supplementary materials and will publish it upon acceptance. It contains a file `REPRODUCE.md` with all commands required to obtain the main results, experiment configurations in the `config/` folder, and the folder `paper/` contains the code to reproduce all figures based on the runs. No external datasets are required.

REFERENCES

Peter Ade et al. The Simons Observatory: Science goals and forecasts. *JCAP*, 02:056, 2019. doi: 10.1088/1475-7516/2019/02/056.

Kensuke Akita and Masahide Yamaguchi. A precision calculation of relic neutrino decoupling. *JCAP*, 08:012, 2020. doi: 10.1088/1475-7516/2020/08/012.

G. Alguero, G. Belanger, F. Boudjema, S. Chakraborti, A. Goudelis, S. Kraml, A. Mjallal, and A. Pukhov. micrOMEGAs 6.0: N-component dark matter. *Comput. Phys. Commun.*, 299:109133, 2024. doi: 10.1016/j.cpc.2024.109133.

Johan Alwall, Michel Herquet, Fabio Maltoni, Olivier Mattelaer, and Tim Stelzer. MadGraph 5 : Going Beyond. *JHEP*, 06:128, 2011. doi: 10.1007/JHEP06(2011)128.

Shun-ichi Amari and Hiroshi Nagaoka. *Methods of Information Geometry*, volume 191 of *Translations of Mathematical Monographs*. American Mathematical Society, 2000.

Peter Brockway Arnold and Cheng-xing Zhai. The Three loop free energy for high temperature QED and QCD with fermions. *Phys. Rev. D*, 51:1906–1918, 1995. doi: 10.1103/PhysRevD.51.1906.

Geneviève Bélanger, Fawzi Boudjema, Andreas Goudelis, Alexander Pukhov, and Bryan Zaldivar. micrOMEGAs5.0 : Freeze-in. *Comput. Phys. Commun.*, 231:173–186, 2018. doi: 10.1016/j.cpc.2018.04.027.

Jack J. Bennett, Gilles Buldgen, Marco Drewes, and Yvonne Y. Y. Wong. Towards a precision calculation of the effective number of neutrinos $N_{\rm eff}$ in the Standard Model I: the QED equation of state. *JCAP*, 03:003, 2020. doi: 10.1088/1475-7516/2020/03/003. [Addendum: JCAP 03, A01 (2021)].

Jack J. Bennett, Gilles Buldgen, Pablo F. De Salas, Marco Drewes, Stefano Gariazzo, Sergio Pastor, and Yvonne Y. Y. Wong. Towards a precision calculation of $N_{\rm eff}$ in the Standard Model II: Neutrino decoupling in the presence of flavour oscillations and finite-temperature QED. *JCAP*, 04:073, 2021. doi: 10.1088/1475-7516/2021/04/073.

Tobias Binder, Torsten Bringmann, Michael Gustafsson, and Andrzej Hryczuk. Dark matter relic abundance beyond kinetic equilibrium. *Eur. Phys. J. C*, 81:577, 2021. doi: 10.1140/epjc/s10052-021-09357-5.

Tobias Binder, Marco Drewes, Yannis Georis, Michael Klasen, Giovanni Pierobon, and Yvonne Y. Y. Wong. Towards a precision calculation of N $_{eff}$ in the Standard Model IV: Estimating the impact of positronium formation. *JCAP*, 10:069, 2025. doi: 10.1088/1475-7516/2025/10/069.

Diego Blas, Julien Lesgourgues, and Thomas Tram. The Cosmic Linear Anisotropy Solving System (CLASS) II: Approximation schemes. *JCAP*, 07:034, 2011. doi: 10.1088/1475-7516/2011/07/034.

Enrico Bothmann et al. Event Generation with Sherpa 2.2. *SciPost Phys.*, 7(3):034, 2019. doi: 10.21468/SciPostPhys.7.3.034.

Erminia Calabrese et al. The Atacama Cosmology Telescope: DR6 constraints on extended cosmological models. *JCAP*, 11:063, 2025. doi: 10.1088/1475-7516/2025/11/063.

E. Camphuis et al. SPT-3G D1: CMB temperature and polarization power spectra and cosmology from 2019 and 2020 observations of the SPT-3G main field. *Phys. Rev. D*, 113(8):083504, 2026. doi: 10.1103/7wt3-9v2y.

Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017. doi: 10.1126/science.aag2302.

Ao Chen and Markus Heyl. Empowering deep neural quantum states through efficient optimization. *Nature Phys.*, 20(9):1476–1481, 2024. doi: 10.1038/s41567-024-02566-1.

Mattia Cielo, Miguel Escudero, Gianpiero Mangano, and Ofelia Pisanti. $N_{\rm eff}$ in the Standard Model at NLO is 3.043. *Phys. Rev. D*, 108(12):L121301, 2023. doi: 10.1103/PhysRevD.108.L121301.

Pablo F. de Salas and Sergio Pastor. Relic neutrino decoupling with flavour oscillations revisited. *JCAP*, 07:051, 2016. doi: 10.1088/1475-7516/2016/07/051.

Marco Drewes, Yannis Georis, Michael Klasen, Luca Paolo Wiggering, and Yvonne Y. Y. Wong. Towards a precision calculation of $N_{\rm eff}$ in the Standard Model. Part III. Improved estimate of NLO contributions to the collision integral. *JCAP*, 06:032, 2024. doi: 10.1088/1475-7516/2024/06/032.

Jens Erler and Shufang Su. The Weak Neutral Current. *Prog. Part. Nucl. Phys.*, 71:119–149, 2013. doi: 10.1016/j.ppnp.2013.03.004.

M. Escudero, G. Jackson, M. Laine, and S. Sandner. Fast and flexible neutrino decoupling. Part I. The Standard Model. *JCAP*, 02:046, 2026a. doi: 10.1088/1475-7516/2026/02/046.

Miguel Escudero, Maksym Ovchynnikov, and Neal Weiner. What does it take to have $N_{\rm eff} < 3$ at CMB times? 3 2026b.

Ivan Esteban, M. C. Gonzalez-Garcia, Michele Maltoni, Ivan Martinez-Soler, João Paulo Pinheiro, and Thomas Schwetz. NuFit-6.0: updated global analysis of three-flavor neutrino oscillations. *JHEP*, 12:216, 2024. doi: 10.1007/JHEP12(2024)216.

Ivan Esteban, M. C. Gonzalez-Garcia, Michele Maltoni, Ivan Martinez-Soler, João Paulo Pinheiro, and Thomas Schwetz. NuFIT 6.1 (2025). `http://www.nu-fit.org`, 2025.

A. Friedman. Über die krümmung des raumes. *Zeitschrift für Physik*, 10(1):377–386, Dec 1922. ISSN 0044-3328. doi: 10.1007/BF01332580. URL `https://doi.org/10.1007/BF01332580`.

Julien Froustey, Cyril Pitrou, and Maria Cristina Volpe. Neutrino decoupling including flavour oscillations and primordial nucleosynthesis. *JCAP*, 12:015, 2020. doi: 10.1088/1475-7516/2020/12/015.

S. Gariazzo, P. F. de Salas, and S. Pastor. Thermalisation of sterile neutrinos in the early Universe in the 3+1 scheme with full mixing matrix. *JCAP*, 07:014, 2019. doi: 10.1088/1475-7516/2019/07/014.

S. L. Glashow. Partial Symmetries of Weak Interactions. *Nucl. Phys.*, 22:579–588, 1961. doi: 10.1016/0029-5582(61)90469-2.

Alessandro Granelli, Kristian Moffat, Yuber F. Perez-Gonzalez, Holger Schulz, and Jessica Turner. ULYSSES: Universal LeptogeneSiS Equation Solver. *Comput. Phys. Commun.*, 262:107813, 2021. doi: 10.1016/j.cpc.2020.107813.

Alessandro Granelli, Juraj Klarić, Dhruv Pasari, Yuber F. Perez-Gonzalez, and Jessica Turner. ULYSSES the Third: An Odyssey Towards a Unified Python Toolkit for Leptogenesis. 5 2026.

Kjell Gustafsson. Control theoretic techniques for stepsize selection in explicit Runge-Kutta methods. *ACM Trans. Math. Software*, 17(4):533–554, 1991. doi: 10.1145/210232.210242.

Theo Heimel, Ramon Winterhalder, Anja Butter, Joshua Isaacson, Claudius Krause, Fabio Maltoni, Olivier Mattelaer, and Tilman Plehn. MadNIS - Neural multi-channel importance sampling. *SciPost Phys.*, 15(4):141, 2023. doi: 10.21468/SciPostPhys.15.4.141.

Theo Heimel, Nathan Huetsch, Fabio Maltoni, Olivier Mattelaer, Tilman Plehn, and Ramon Winterhalder. The MadNIS reloaded. *SciPost Phys.*, 17(1):023, 2024. doi: 10.21468/SciPostPhys.17.1.023.

Theo Heimel, Olivier Mattelaer, and Ramon Winterhalder. MadSpace – Event Generation for the Era of GPUs and ML. 2 2026.

Jan Hermann, Zeno Schätzle, and Frank Noé. Deep-neural-network solution of the electronic Schrödinger equation. *Nature Chem.*, 12(10):891–897, 2020. doi: 10.1038/s41557-020-0544-y.

Yonit Hochberg, Eric Kuflik, Tomer Volansky, and Jay G. Wacker. Mechanism for Thermal Relic Dark Matter of Strongly Interacting Massive Particles. *Phys. Rev. Lett.*, 113:171301, 2014. doi: 10.1103/PhysRevLett.113.171301.

Edwin Hubble. A relation between distance and radial velocity among extra-galactic nebulae. *Proc. Nat. Acad. Sci.*, 15:168–173, 1929. doi: 10.1073/pnas.15.3.168.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015.

R. Kleiss, W. James Stirling, and S. D. Ellis. A New Monte Carlo Treatment of Multiparticle Phase Space at High-energies. *Comput. Phys. Commun.*, 40:359, 1986. doi: 10.1016/0010-4655(86)90119-0.

K. S. Kumar, Sonny Mantry, W. J. Marciano, and P. A. Souder. Low Energy Measurements of the Weak Mixing Angle. *Ann. Rev. Nucl. Part. Sci.*, 63:237–267, 2013. doi: 10.1146/annurev-nucl-102212-170556.

Georges Lemaitre. A Homogeneous Universe of Constant Mass and Growing Radius Accounting for the Radial Velocity of Extragalactic Nebulae. *Annales Soc. Sci. Bruxelles A*, 47:49–59, 1927. doi: 10.1007/s10714-013-1548-3.

G. Peter Lepage. Adaptive multidimensional integration: VEGAS enhanced. *J. Comput. Phys.*, 439:110386, 2021. doi: 10.1016/j.jcp.2021.110386.

Antony Lewis, Anthony Challinor, and Anthony Lasenby. Efficient computation of CMB anisotropies in closed FRW models. *Astrophys. J.*, 538:473–476, 2000. doi: 10.1086/309179.

Thibaut Louis et al. The Atacama Cosmology Telescope: DR6 power spectra, likelihoods and ΛCDM parameters. *JCAP*, 11:062, 2025. doi: 10.1088/1475-7516/2025/11/062.

Ziro Maki, Masami Nakagawa, and Shoichi Sakata. Remarks on the unified model of elementary particles. *Prog. Theor. Phys.*, 28:870–880, 1962. doi: 10.1143/PTP.28.870.

Gianpiero Mangano, Gennaro Miele, Sergio Pastor, Teguayco Pinto, Ofelia Pisanti, and Pasquale D. Serpico. Relic neutrino decoupling including flavor oscillations. *Nucl. Phys. B*, 729:221–234, 2005. doi: 10.1016/j.nuclphysb.2005.09.041.

Johannes Müller and Marius Zeinhofer. Achieving High Accuracy with PINNs via Energy Natural Gradient Descent. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 25471–25485. PMLR, 2023.

Maksym Ovchynnikov and Vsevolod Syvolap. How new physics affects primordial neutrinos decoupling: Direct simulation Monte Carlo approach. *Phys. Rev. D*, 111(6):063527, 2025a. doi: 10.1103/PhysRevD.111.063527.

Maksym Ovchynnikov and Vsevolod Syvolap. Primordial Neutrinos and New Physics: Novel Approach to Solving the Neutrino Boltzmann Equation. *Phys. Rev. Lett.*, 134(10):101003, 2025b. doi: 10.1103/PhysRevLett.134.101003.

David Pfau, James S. Spencer, Alexander G. D. G. Matthews, and W. M. C. Foulkes. Ab initio solution of the many-electron Schrödinger equation with deep neural networks. *Phys. Rev. Res.*, 2(3):033429, 2020. doi: 10.1103/PhysRevResearch.2.033429.

B. Pontecorvo. Inverse Beta Processes and Nonconservation of Lepton Charge. *Sov. Phys. JETP*, 7: 172–173, 1958.

H. P. Robertson. Kinematics and World-Structure. *Astrophys. J.*, 82:284–301, 1935. doi: 10.1086/143681.

G. Sigl and G. Raffelt. General kinetic description of relativistic mixed neutrinos. *Nucl. Phys. B*, 406:423–451, 1993. doi: 10.1016/0550-3213(93)90175-O.

Gustaf Söderlind. Digital filters in adaptive time-stepping. *ACM Trans. Math. Software*, 29(1):1–26, 2003. doi: 10.1145/641876.641877.

James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum Natural Gradient. *Quantum*, 4:269, 2020. doi: 10.22331/q-2020-05-25-269.

Eite Tiesinga, Peter J. Mohr, David B. Newell, and Barry N. Taylor. CODATA recommended values of the fundamental physical constants: 2018*. *Rev. Mod. Phys.*, 93(2):025010, 2021. doi: 10.1103/RevModPhys.93.025010.

A. G. Walker. On Milne's Theory of World-Structure. *Proc. Lond. Math. Soc. s*, 2-42(1):90–127, 1937. doi: 10.1112/plms/s2-42.1.90.

Steven Weinberg. A Model of Leptons. *Phys. Rev. Lett.*, 19:1264–1266, 1967. doi: 10.1103/PhysRevLett.19.1264.

Jong-Hyun Yoon. Boltzmann equation solver for thermalization. *Comput. Phys. Commun.*, 327: 110295, 2026. doi: 10.1016/j.cpc.2026.110295.

Cheng-xing Zhai and Boris M. Kastening. The Free energy of hot gauge theories with fermions through g^5. *Phys. Rev. D*, 52:7232–7246, 1995. doi: 10.1103/PhysRevD.52.7232.

P. A. Zyla et al. Review of Particle Physics. *PTEP*, 2020(8):083C01, 2020. doi: 10.1093/ptep/ptaa104.

A METHOD DETAILS

A.1 NEURAL DISTRIBUTION FUNCTIONS

Degrees of freedom. The formalism in Section 4.1 focuses on the case of a particle with a single degree of freedom for simplicity. However, in the neutrino decoupling calculation in Section 5.3 we describe the electron species with four degrees of freedom. As we begin our evolution without asymmetries, and there are no effects that distinguish between particle and antiparticle, or between left and right helicity, all share the same distribution function. This is accounted for by factors of the relativistic degrees of freedom for each particle, g, in Eq. (8),

$$f(y) = \frac{4\pi^2 E}{g y^2} \eta p(y) \,, \qquad \eta = g \int \mathrm{d}y \frac{y^2}{4\pi^2 E} f(y) \,. \tag{18}$$

Electrons, for example, have $g = 4$. The moments of the distribution, defined in Section 2, become $M_a[\phi] = g_a \int \mathrm{d}^3 y/(2\pi)^3 \phi(y) f_a(y)$, and are the summed contributions from each individual degree of freedom. For the collision integral, we work in terms of the spin-summed matrix elements $|\mathcal{M}_k|^2$ which leaves the one-particle Lorentz-invariant phase-space measure $\mathrm{d}\Pi_a$ unchanged, as it still refers to a single degree of freedom.

Equilibrium solutions with neural distribution functions. The ansatz of Eq. (9) using the kinetic energy $\varepsilon = (E - m)/T$ is chosen such that equilibrium distributions can naturally be represented. Expanding the Bose-Einstein and Fermi-Dirac distributions in a geometric series, we find

$$f_{\mathrm{eq}}(y) = \frac{1}{e^{(E-\mu)/T} \pm 1} = \sum_{k=1}^{\infty} (\mp 1)^{k+1} e^{-k(m-\mu)/T} e^{-k\varepsilon} \,, \quad a = \frac{m}{T} \,, \quad \varepsilon = \frac{E-m}{T} \,. \tag{19}$$

Inserting this into Eq. (8) with $y = T\sqrt{\varepsilon(\varepsilon + 2a)}$, $y^2\,\mathrm{d}y/E = Ty\,\mathrm{d}\varepsilon$ gives

$$\begin{aligned} p_{\mathrm{eq}}(y)\,\mathrm{d}y &= \frac{T^2}{4\pi^2 \eta_{\mathrm{eq}}} \sum_{k=1}^{\infty} (\mp 1)^{k+1} e^{-k(m-\mu)/T} \sqrt{\varepsilon(\varepsilon+2a)}\, e^{-k\varepsilon}\,\mathrm{d}\varepsilon \,, \\ \eta_{\mathrm{eq}} &= \frac{mT}{4\pi^2} \sum_{k=1}^{\infty} (\mp 1)^{k+1} \frac{e^{k\mu/T}}{k} K_1(ka) \,. \end{aligned} \tag{20}$$

The fermionic series has alternating signs and can therefore not be parameterised as a gamma mixture distribution, motivating the $f = f_0/(1 + f_0)$ parameterisation. The density $p_{\mathrm{eq}}(y)$ reduces to a series of gamma mixture distributions for massless particles, $a = 0$, and in the non-relativistic limit, $a \to \infty$. For a massless boson with $\mu = 0$, we find

$$p_{\mathrm{eq}}(y)\,\mathrm{d}y = \frac{6}{\pi^2} \sum_{k=1}^{\infty} \frac{1}{k^2}\, \gamma(\varepsilon|2,k)\,\mathrm{d}\varepsilon \,, \qquad \eta_{\mathrm{eq}} = \frac{T^2}{24} \,. \tag{21}$$

In the non-relativistic limit, only the $k = 1$ term survives and we obtain the Maxwell-Boltzmann distribution, which is a single gamma distribution, $p_{\mathrm{eq}}(y)\,\mathrm{d}y = \gamma(\varepsilon|3/2, 1)\mathrm{d}\varepsilon$. For general masses, we interpolate between the two limits,

$$\sqrt{\varepsilon(\varepsilon+2a)} = \left(\varepsilon + \sqrt{2a\varepsilon}\right) A(\varepsilon) \,, \qquad A(\varepsilon) = \frac{\sqrt{\varepsilon+2a}}{\sqrt{\varepsilon}+\sqrt{2a}} \in \left[\frac{1}{\sqrt{2}}, 1\right] \,. \tag{22}$$

The exact equilibrium density is therefore a mixture of $\gamma(\varepsilon|2,k)$ and $\gamma(\varepsilon|3/2,k)$ components, with a bounded correction factor $A(\varepsilon)$. We use this form to sample exact equilibrium distributions by rejection. A learned distribution function instead absorbs the $A(\varepsilon)$ prefactor into the mixture parameters (α_m, λ_m).

For fermions, Eq. (10) maps the Fermi-Dirac distribution onto the Maxwell-Boltzmann distribution as $f_0(y) = e^{-(E-\mu)/T}$. The corresponding rejection step accepts a proposal $y \sim p_0(y)$ with probability $1/(1 + f_0(y))$ and efficiency N. For the massless Fermi-Dirac distribution at $\mu = 0$, this gives $N = \pi^2/12 \approx 0.8$.

Skewed gamma mixture distributions. The design choice $p(y) \propto y^2 f/E$ is based on the invariant measure $\mathrm{d}^3 y/E$, which makes it the natural proposal distribution for the initial states of the collision

integral. However, the fit function $g = y^2 f$ corresponds to the integration measure $\mathrm{d}^3 y$, for which $p(y)$ is not the optimal proposal distribution. To overcome this, we define the skewed density

$$p_\ell(y) = \frac{E^\ell}{\mathbb{E}_{y\sim p(y)}[E^\ell]} p(y) \; . \tag{23}$$

Using the skewed density p_1 to sample $y \sim p_1(\cdot|\omega_{n+1,a})$ in Algorithm 2 improves the sampling efficiency compared to the naive choice $p(\cdot|\omega_{n+1,a})$. We recover the original density with $p_0 = p$, and use p_1 to sample the test momenta in Alg. 2. The skewed density p_ℓ is again a gamma mixture distribution: using $E = m + T\varepsilon$, we find for $\ell = 1$

$$E\gamma(\varepsilon|\alpha,\lambda) = m\gamma(\varepsilon|\alpha,\lambda) + T\frac{\Gamma(\alpha+1)}{\Gamma(\alpha)\lambda}\gamma(\varepsilon|\alpha+1,\lambda) \; . \tag{24}$$

In this way, we obtain from a gamma mixture p another gamma mixture $\propto Ep$ by doubling the number of mixture components and updating the normalisation. Expressions for $\ell > 1$ can be derived in the same way, but require increasingly many mixture components. Note that this trick only works for $\ell \geq 0$, which is the reason why we define the parameterisation based on the $\mathrm{d}^3 y/E$ measure. For fermions, we draw from the skewed variant $p_{0,1}$ of the bosonic density p_0 and apply the standard rejection step with probability $1/(1+f_0)$. We find that this modification improves the efficiency of the Monte Carlo estimate, and use it in all experiments.

Moments of neural distribution functions. With the factorisation of Eq. (8), any moment of f reduces to an expectation under the normalised density,

$$\langle h\rangle_f \equiv \int \frac{\mathrm{d}^3 y}{(2\pi)^3}\, h(y)\, f(y) = \frac{1}{2\pi^2}\int \mathrm{d}y\, y^2\, h(y)\, \frac{4\pi^2 E}{y^2}\, \eta\, p(y) = 2\eta\, \mathbb{E}_{y\sim p(y)}\big[E\, h(y)\big] \; . \tag{25}$$

For fermions, Eq. (10) gives $p = p_0/(N(1+f_0))$ and $\eta = N\eta_0$, so the normalisation drops out, and we find

$$\langle h\rangle_f = 2\eta_0\, \mathbb{E}_{y\sim p_0(y)}[E\, h/(1+f_0)] \; . \tag{26}$$

The gamma mixture is a density in ε, and $E = m + T\varepsilon$, $y^2 = T^2\varepsilon(\varepsilon+2a)$ are polynomial in ε, so every moment with polynomial h follows from

$$\mathbb{E}_{y\sim p(y)}\left[\varepsilon(y)^q\right] = \sum_{m=1}^M w_m \int \mathrm{d}\varepsilon\, \varepsilon^q \gamma(\varepsilon|\alpha_m,\lambda_m) = \sum_{m=1}^M w_m \frac{1}{\lambda_m^q}\frac{\Gamma(\alpha_m+q)}{\Gamma(\alpha_m)} \; . \tag{27}$$

For general h, and for all fermion moments because of the factor $1/(1+f_0)$, we use Gauss-Laguerre quadrature with N_{quad} nodes t_i and weights v_i, $\int_0^\infty \mathrm{d}t\, e^{-t} g(t) \approx \sum_i v_i\, g(t_i)$. Substituting $r = \lambda_m\varepsilon$ in each component,

$$\begin{aligned}
\mathbb{E}_{y\sim p(y)}\left[h(y)\right] &= \sum_{m=1}^M w_m \frac{\lambda_m^{\alpha_m}}{\Gamma(\alpha_m)} \int \mathrm{d}\varepsilon\, h(\varepsilon)\varepsilon^{\alpha_m-1} e^{-\lambda_m\varepsilon} \; , \\
&= \sum_{m=1}^M \frac{w_m}{\Gamma(\alpha_m)} \int \mathrm{d}r\, h(r/\lambda_m) r^{\alpha_m-1} e^{-r} \; , \\
&= \sum_{m=1}^M \frac{w_m}{\Gamma(\alpha_m)} \sum_{i=1}^{N_{\text{quad}}} v_i h(t_i/\lambda_m) t_i^{\alpha_m-1} \; .
\end{aligned} \tag{28}$$

For massive particles the kinematic factor $y = T\sqrt{\varepsilon(\varepsilon+2a)}$ has a $\sqrt{\varepsilon}$ branch point at $\varepsilon = 0$ that spoils the spectral convergence of the Laguerre rule. Substituting $t = u^2$ maps it to a Gaussian integral over the half line, $\int_0^\infty \mathrm{d}t\, e^{-t} g(t) = \int_0^\infty 2u\, \mathrm{d}u\, e^{-u^2} g(u^2)$, where $\sqrt{t} = u$ is polynomial. We evaluate it with the positive nodes u_i of a $2N_{\text{quad}}$-point Gauss-Hermite rule, i.e. we use the rule above with $t_i = u_i^2$ and $v_i = 2u_i v_i^{\text{H}}$. We switch from Laguerre to Hermite nodes for $a = m/T > 0.1$.

A.2 COLLISION INTEGRAL

Here we give additional information on the collision integral evaluation described in Section 4.2, and in particular, in Algorithm 1. We use the `TPropagatorMapping` of Heimel et al. (2026) with coefficient 0.7 as the phase-space mapping, denoted $\mathrm{MadSpace}$ in Algorithm 1.

Distribution function sampling. The factorisation of $f(y)$ into the invariant density, η, and the normalised density, $p(y)$, in Eq. (8) is designed to enable efficient importance sampling for the collision integral initial states. Sampling the angles uniformly $\hat{\Omega} \sim \mathcal{U}(S^2)$, and momenta as $y \sim p(y)$, we obtain for the corresponding phase-space measure

$$\frac{\mathrm{d}\Pi}{(4\pi)^{-1}p(y)} = \frac{\mathrm{d}\Omega}{(4\pi)^{-1}} \frac{y^2 \mathrm{d}y}{(2\pi)^3 2E} \frac{1}{p(y)} = \frac{\mathrm{d}\Omega}{(4\pi)^{-1}} \frac{y^2 \mathrm{d}y}{(2\pi)^3 2E} \frac{4\pi^2 E}{y^2} \frac{\eta}{f(y)} = \mathrm{d}\Omega \, \mathrm{d}y \frac{\eta}{f(y)} \, . \tag{29}$$

The factors in $p(y)$ cancel with those in the Lorentz-invariant phase-space measure, $\mathrm{d}\Pi$, by construction, leaving a factor $\eta/f(y)$, with the $f(y)$ factor naturally cancelling with the factors of $f(y)$ or $1 \pm f(y)$ in the statistical factor $\Lambda[f]$. For neural importance sampling the initially sampled momentum, $\tilde{y}$, is modified with the flow, $\tilde{y} \to y$, and we then obtain $f(y)$ as $f(y) = (4\pi^2 E)/y^2 p(\tilde{y})$, where the prefactor depends on the modified y and the initially sampled $\tilde{y}$ appears in the density.

Neyman allocation. When multiple processes contribute to a collision integral, using the same number of Monte Carlo samples N_i for each process is typically inefficient, because the total variance is dominated by the process with the largest per-sample variance. Instead, we dynamically set the sample count N_i based on the total process standard deviations, σ_i, of the previous time step. For a given sample budget, N_{tot}, we use the so-called Neyman allocation

$$N_i = N_{\mathrm{tot}} \frac{\sigma_i}{\sum_j \sigma_j} \, . \tag{30}$$

Phase-space mappings. Phase-space mappings turn uniform random numbers into kinematic configurations that satisfy on-shell relations and energy-momentum conservation. The mapping is constructed such that the corresponding Jacobian transforms the uniform base density into a phase-space density that is a good approximation to the integrand, minimising integral variance, see Eq. (5). A range of phase-space mappings exist for different structures in the integrand, such as the Rambo mapping (Kleiss et al., 1986) for an integrand that is constant up to the phase-space measure $\mathrm{d}\Phi_k$, or a Breit-Wigner mapping for a resonant structure in the matrix element. Modern event generators build custom phase-space mappings recursively based on Feynman diagram topologies, see Heimel et al. (2026) for a recent review. Our implementation uses the MadSpace package (Heimel et al., 2026) which supports all the options discussed above.

$\mathrm{ToUnitCube}$ mapping. The $\mathrm{ToUnitCube}$ mapping in Alg. 1 maps the momenta $y_i \in \mathbb{R}$, angles $\phi_i \in [0, 2\pi]$, $\cos\theta_i \in [-1, 1]$, and $u \in [0, 1]$ to the unit hypercube $[0, 1]^N$. The optimal choice is to use the corresponding cumulative distribution function, which ensures that the resulting unit hypercube is populated uniformly. The u parameters need no modification and for the uniformly sampled angles, ϕ_i and $\cos\theta_i$, a shift and rescaling is sufficient. For the momenta y, we use a scaled exponential transform as a simple proxy for the full gamma mixture distribution, for which the cumulative distribution function is not available in closed form. Using the same mapping for each species and writing $\hat{\Omega}_i = (\phi_i, \cos\theta_i)$, we arrive at the mapping

$$\mathrm{ToUnitCube}(y_2, \ldots \hat{\Omega}_1, \ldots u) = \left(1 - \exp\left(-\frac{\varepsilon_2}{\mathbb{E}[\varepsilon_2]}\right), \ldots \left(\frac{\phi_1}{2\pi}, \frac{1 + \cos\theta_1}{2}\right), \ldots, u\right) \, . \tag{31}$$

A.3 BOLTZMANN EQUATION TRAINING

Hubble parameter. The Hubble parameter is given in terms of the Planck mass m_{Pl} and the total energy density,

$$H = \sqrt{\frac{8\pi\rho_{\mathrm{tot}}}{3m_{\mathrm{Pl}}^2}} \, . \tag{32}$$

Note that m_{Pl} is the Planck mass in comoving units, related to the physical Planck mass by $m_{\mathrm{Pl}} = (x/m_0) m_{\mathrm{Pl}}^{\mathrm{phys}}$.

Integrator schemes. Here we specify the discretisation used for the time evolution in terms of a generic scheme (A, γ) in Eq. (13). We implement three specific schemes:

$$
\begin{aligned}
\text{Euler}: \quad & A = g_n, \qquad \gamma = 1, \\
\text{BDF2}: \quad & A = a g_n + b g_{n-1}, \qquad \gamma = \frac{1+r}{1+2r}, \\
& a = \frac{(1+r)^2}{1+2r}, \qquad b = -\frac{r^2}{1+2r}, \qquad r = \frac{\log x_{n+1} - \log x_n}{\log x_n - \log x_{n-1}}\ , \\
\text{CN}: \quad & A = g_n + \frac{c_a g_n + c_b g_{n-1} + c_c g_{n-2}}{2}, \qquad \gamma = \frac{1}{2}, \\
& c_a = \frac{r_1(2r_2+1)}{1+r_2}, \qquad c_b = -r_1(1+r_2), \qquad c_c = \frac{r_1 r_2^2}{1+r_2}, \\
& r_1 = \frac{\log x_{n+1} - \log x_n}{\log x_n - \log x_{n-1}}, \qquad r_2 = \frac{\log x_n - \log x_{n-1}}{\log x_{n-1} - \log x_{n-2}}\ .
\end{aligned}
\tag{33}
$$

The Euler scheme is the standard single-step approach. The BDF2 scheme combines information from two previous time steps. Our most powerful and default choice, the CN scheme, starts from $A = g_n + \gamma h s(g_n)$, then improves the collision integral estimate with the trapezoid rule as $s(g_n) \leftarrow (s(g_n) + s(g_{n+1}))/2$, and then inserts an estimate for $s(g_n)$ based on the previous solutions. In each case, the coefficients above are obtained from solving a linear system of conditions that match derivatives at the previous solutions. We find that the CN and BDF2 schemes significantly improve the solver precision.

Collision operator derivative estimate. The iteration speed when solving $g'_{n+1} = T(g_{n+1}) = g_{n+1} - M^{-1}F(g_{n+1})$ of Section 4.3 depends on the choice of M. The ideal choice $M = \partial F/\partial g$, following from the optimal update to leading order $\partial T/\partial g = 1 - M^{-1}\partial F/\partial g = 0$, is impractical due to the large time and memory cost of differentiating the collision operator. On the other hand, the naive choice $M = 1$, corresponding to $\partial s/\partial g = 0$, suffers from unstable numerical behaviour. To find an optimal middle ground, we approximate the collision operator derivative of species a:

$$
\begin{aligned}
\frac{\partial s_a}{\partial g_a} &= \frac{\partial}{\partial f_a}\frac{1}{2E_a H}\sum_k \int \mathrm{d}\Phi_{k\setminus a}|\mathcal{M}_k|^2 \Lambda_k[f]\ , \\
&= \frac{\partial}{\partial f_a}\frac{1}{2E_a H}\sum_k \int \mathrm{d}\Phi_{k\setminus a}|\mathcal{M}_k|^2 \left((1\mp f_a)\prod_{b\in k_{\text{out}}} f_b \prod_{b\in k_{\text{in}}\setminus 1}(1\mp f_b) - f_a(\text{in}\leftrightarrow\text{out})\right)\ , \\
&= -\frac{\partial}{\partial f_a}\left[\frac{1}{2E_a H}\sum_k \int \mathrm{d}\Phi_{k\setminus a}|\mathcal{M}_k|^2 f_a \left(\pm\prod_{b\in k_{\text{out}}} f_b \prod_{b\in k_{\text{in}}\setminus 1}(1\mp f_b) + (\text{in}\leftrightarrow\text{out})\right) + \text{const}\right]\ , \\
&\approx -\frac{1}{2E_a H}\sum_k \int \mathrm{d}\Phi_{k\setminus a}|\mathcal{M}_k|^2 \left(\pm\prod_{b\in k_{\text{out}}} f_b \prod_{b\in k_{\text{in}}\setminus 1}(1\mp f_b) + (\text{in}\leftrightarrow\text{out})\right) \equiv -\kappa_a\ .
\end{aligned}
\tag{34}
$$

In the last step, we have neglected f_a occurrences in the matrix element and on other legs of the collision operator. The resulting approximation to $\partial s(y)/\partial g(y')$ is exact on the diagonal, $y = y'$. For bosons, stimulated emission allows $\kappa < 0$, which can lead to unstable regimes. We set $\kappa = \max(\kappa, 0)$ to avoid this.

Natural gradient update with conserved moments. If certain moments of a species such as the number density n or energy density ρ are conserved, then the corresponding direction in the neural distribution function parameter landscape can suffer from systematic shifts from the natural gradient step that build up over time steps. This does not happen for non-conserved moments, because collisions damp the effect. To correct for this, we first construct a natural gradient update that acts on the parameter directions that leave conserved moments unchanged, and afterwards do a low-dimensional refinement update on the conserved moments.

We consider a list of weighting functions $\psi = (\psi_1, \ldots, \psi_n)$ with moments $\Psi[g] = \int \mathrm{d}y\, \psi(y) g(y)$ that are conserved by collisions, $\Psi[s(g)] = 0$. We use $\psi = (1, E)$ throughout this work to handle

number and energy density conservation. Including moments in Ψ that are not conserved does not reduce the precision of the update, but marginally increases the cost.

To remove this shift, we want to enforce $\Delta\Psi = \Psi[g - g_{\text{target}}] = 0$ in addition to the standard objective of Eq. (6). We estimate the moments $\Psi[g]$ with Monte Carlo integration, $\Psi[g] = \mathbb{E}_{y\sim p(y)}[\psi(y)g(y)/p(y)]$. Defining the corresponding Jacobian $D = \partial\Delta\Psi/\partial\theta$, we obtain $D\delta\theta = -\Delta\Psi$ as a constraint on the parameter update $\delta\theta = \theta_* - \theta_0$ in Eq. (7). We extend the objective $L(\theta)$ of Eq. (6) with a Lagrange multiplier λ to enforce this condition, and obtain

$$\begin{pmatrix} F & D^\top \\ D & 0 \end{pmatrix} \begin{pmatrix} \delta\theta \\ \lambda \end{pmatrix} = \begin{pmatrix} -J^\top W r \\ -\Delta\Psi \end{pmatrix} . \tag{35}$$

Inserting the first row, $\delta\theta = \delta\theta_{\text{NG}} - F^{-1}D^\top\lambda$ with $\delta\theta_{\text{NG}} = -F^{-1}J^\top W r$, into the second row yields the solution $\lambda = S^{-1}(D\delta\theta_{\text{NG}} + \Delta\Psi)$ for the Lagrange multiplier, with the matrix $S = DF^{-1}D^\top$. Substituting this back into the first row yields the solution for the constrained natural gradient step

$$\delta\theta = \left(\mathbb{I} - F^{-1}D^\top S^{-1}D\right)\delta\theta_{\text{NG}} - F^{-1}D^\top S^{-1}\Delta\Psi \equiv \delta\theta_{\text{shape}} + \delta\theta_{\text{moment}} . \tag{36}$$

The shape step $\delta\theta_{\text{shape}}$ leaves the moments unchanged, $D\delta\theta_{\text{shape}} = 0$, and the moment step $\delta\theta_{\text{moment}}$ shifts the moments by the required amount, $D\delta\theta_{\text{moment}} = -\Delta\Psi$, using a minimal step in parameter space as defined by the Fisher metric. The two steps commute only at linear order in $\delta\theta$, and we therefore first apply the shape step $\delta\theta_{\text{shape}}$ and do the moment step $\delta\theta_{\text{moment}}$ afterwards to ensure that the moment change is correct. In practice, we first determine D and S in addition to F, construct F^{-1} and S^{-1} with the usual regularised matrix inversion, then construct and apply the shape step $\delta\theta_{\text{shape}}$, and finally perform the moment step $\delta\theta_{\text{moment}}$ by iteratively refining the parameters and evaluating $\Delta\Psi$ at updated parameters until the moments match.

See Appendix D.4 for a study on the effect of this extension of the natural gradient update. We find that this constraint removes systematic shifts in conserved moments, and therefore use it in all our experiments. The computational cost is negligible compared to the collision integral evaluation.

Time step convergence criteria. In each step of Algorithm 2, we test the following two convergence criteria after constructing the residual r:

1. Noise floor: The residual is dominated by Monte Carlo noise, $\text{median}_y[r^2(y)/\sigma_r^2(y)] \leq c\chi^2_{1,\text{median}}$, where $\chi^2_{1,\text{median}} \approx 0.455$ is the median of the χ^2_1 distribution and we use $c = 1.2$.
2. Plateau: The objective $r^\top W r$ decreased by less than a factor r_{plateau} over the last three iterations with $r_{\text{plateau}} = 0.05$, indicating that the residual is limited by the expressivity of the ansatz or the solver.

If either criterion is satisfied, we accept the current parameters θ_{n+1} instead of performing another natural gradient update. This typically occurs after 5-10 iterations for unconditional trainings. For conditional trainings, we additionally require a minimum number of iterations before accepting convergence, typically 15, to ensure that the network sees a sufficient range of conditions. Larger condition spaces and stronger dependences on the condition require a larger batch size or more startup iterations.

Step size controller. We use a dynamic step size controller (Gustafsson, 1991; Söderlind, 2003) to determine the step size $h_n = \log x_{n+1} - \log x_n$ in Eq. (13), based on the deviation of the fixed-point target from the linear extrapolation of the previous solutions, $e(y) = g_{\text{target}}(y) - g_n(y) - r_n(g_n(y) - g_{n-1}(y))$, $r_n = h_n/h_{n-1}$. We use this error to construct the error ratio ζ_n which is used to compute the next step size, h_{n+1},

$$\zeta_n = \max_{\text{species}} \mathbb{E}_{y\sim p(y)} \left[\left(\frac{e(y)}{\epsilon_{\text{abs}} + \epsilon_{\text{rel}}|g_n(y)| + c\sigma_g(y)} \right)^2 \right]^{1/2} ,$$

$$h_{n+1} = h_n \beta \zeta_n^{-k_I/2} \left(\frac{\zeta_{n-1}}{\zeta_n} \right)^{k_P/2} . \tag{37}$$

We use the values $\beta = 0.9$, $k_I = 0.7$, $k_P = 0.4$. The Monte Carlo uncertainty of the target $g_{\text{target}}(y)$ is given by $\sigma_g(y) = \gamma h_n \sigma_s(y)/(1 + \gamma h_n \kappa)$, where $\sigma_s(y)$ is the Monte Carlo uncertainty

of the collision integral estimate $s(y)$. The σ_g term in ζ_n implements a noise floor that prevents step size decrease if $e(y)$ is dominated by noise. We set $c = 2$ but find that this parameter has little impact in our experiments. The absolute and relative tolerances $\epsilon_{\rm abs}$ and $\epsilon_{\rm rel}$ set the tolerance band for $e(y)$, typically $\epsilon_{\rm abs} \sim 10^{-8}$, $\epsilon_{\rm rel} \sim 10^{-6}$. We further enforce $h_{n+1} \in [\Delta \log x_{\rm min}, \Delta \log x_{\rm max}]$ to prevent problems from very large or small steps.

B NEUTRINO DECOUPLING FORMALISM

Standard Model neutrino decoupling serves as a precision test of Boltzmann equation solvers. We use the fully momentum-dependent, three-flavour transport treatments introduced in several previous works (Mangano et al., 2005; de Salas & Pastor, 2016; Akita & Yamaguchi, 2020; Froustey et al., 2020; Bennett et al., 2021), and assume FLRW spacetime, no lepton asymmetry, no CP-violating phase in the lepton sector, massless neutrinos for the collision kinematics, and truncate weak collision rates at leading order in the Fermi constant, G_F. We include finite-temperature quantum electrodynamic (QED) corrections to the equation of state up to $\mathcal{O}(e^5)$, and assume that photons and electrons remain in exact equilibrium at all times. The comoving variables introduced in Section 2 are used throughout.

Quantum kinetic equation. Accounting for the effects of neutrino mixing promotes the neutrino distribution functions, f_α, $\alpha \in \{e, \mu, \tau\}$, to the Hermitian density matrix F, whose components are defined by the two-point correlation function

$$\langle a^\dagger_\beta(\vec{y}) a_\alpha(\vec{y}')\rangle = (2\pi)^3 2E_y \delta^{(3)}(\vec{y} - \vec{y}') F_{\alpha\beta}(y) \, . \tag{38}$$

The diagonal elements of F are the occupations of each flavour eigenstate, whilst the off-diagonal components encode coherence. The antiparticle density matrix, $\bar{F}$, has components $\bar{F}_{\alpha\beta}(y) \sim \langle b^\dagger_\alpha b_\beta\rangle$, and is related to the particle density matrix by $\bar{F} = F^\top$ in the absence of CP violation and lepton asymmetry.

The density matrix evolves according to the quantum kinetic equation (QKE) (Sigl & Raffelt, 1993)

$$Hx\frac{\partial F}{\partial x} = -i[\mathcal{H}_{\rm eff}, F] + \mathcal{C}_\nu[F], \quad \mathcal{H}_{\rm eff} = \frac{U\Delta M^2 U^\dagger}{2y} - \frac{4G_F y}{\sqrt{2}m_W^2}(\rho_e + P_e)\Pi_e - \frac{8\sqrt{2}G_F y}{3m_Z^2}\varrho_\nu \, , \tag{39}$$

where U is the Pontecorvo–Maki–Nakagawa–Sakata neutrino mixing matrix (Pontecorvo, 1958; Maki et al., 1962), $\Delta M^2 = {\rm diag}(0, \Delta m^2_{21}, \Delta m^2_{31})$, $\Pi_e = {\rm diag}(1, 0, 0)$, and m_W and m_Z are the weak gauge boson masses. The first term in $\mathcal{H}_{\rm eff}$ governs neutrino oscillations, transferring occupation between flavour eigenstates. The remaining terms describe coherent forward scattering in the early-universe plasma, which modify the oscillation frequencies. The ϱ_ν term is proportional to the identity, and so vanishes under the commutator, when the small differences between neutrino flavours are neglected (Froustey et al., 2020). As such, we will not include it going forward. Neither term modifies the total occupation, captured by ${\rm Tr}[F]$.

The energy density, ρ_a, and pressure, P_a, of a species are moments of the distribution functions

$$\rho_a = g_a \int \frac{{\rm d}^3 y}{(2\pi)^3} E_a f_a, \qquad P_a = g_a \int \frac{{\rm d}^3 y}{(2\pi)^3} \frac{y^2}{3E_a} f_a, \qquad \varrho_\nu = 2\int \frac{{\rm d}^3 y}{(2\pi)^3} y \, {\rm Re}[F] \, , \tag{40}$$

with $g_e = 4$ for electrons, and $g_\gamma = 2$ for photons. The full matrix ϱ_ν contributes to coherent forward scattering, but only its trace, $\rho_\nu = {\rm Tr}[\varrho_\nu]$, proportional to the physical neutrino energy density, contributes to gravitational interactions. If we neglect neutrino mixing, F is diagonal and the commutator $[\mathcal{H}_{\rm eff}, F]$ vanishes, recovering the Boltzmann equation of Eq. (1).

Collision operator. For a process $r = a(y_1) + b(y_2) \leftrightarrow c(y_3) + d(y_4)$, the QKE collision operator is the matrix-valued expression

$$\begin{aligned} \mathcal{C}^{(a)}_\nu[F] = & \frac{1}{2E_1}\sum_r S^{(r)} \int \left(\prod_{i=2}^4 {\rm d}\Pi_i\right) (2\pi)^4 \delta^{(4)}(y_1 + y_2 - y_3 - y_4) \\ & \times \left[\frac{1}{2}\left\{\mathcal{K}^{(r)}_{\mathcal{G}}, \widetilde{F}^{(a)}_1\right\} - \frac{1}{2}\left\{\mathcal{K}^{(r)}_{\mathcal{L}}, F^{(a)}_1\right\}\right], \qquad \widetilde{F} = \mathbb{I} - F \, , \end{aligned} \tag{41}$$

with $S^{(r)}$ the symmetry factor. The superscript (a) denotes the particle species, (r) the process, and the subscript the momentum label, *e.g.* $F_1^{(a)} = F^{(a)}(y_1)$. The gain and loss kernels sum over all spin and flavour interaction channels:

$$\left(\mathcal{K}_{\mathcal{G}}^{(r)}\right)_{\alpha\beta} = \sum_{\text{spins}} \sum_{\mu,\mu'} \sum_{\nu,\nu'} \sum_{\rho,\rho'} \mathcal{M}^{(ab;cd)*}_{\alpha\mu;\nu\rho} \mathcal{M}^{(ab;cd)}_{\beta\mu';\nu'\rho'} \left(\widetilde{F}_2^{(b)}\right)_{\mu'\mu} \left(F_3^{(c)}\right)_{\nu\nu'} \left(F_4^{(d)}\right)_{\rho\rho'} , \tag{42}$$

where $\mathcal{M}^{(ab;cd)}_{\alpha\mu;\nu\rho}$ is the amplitude for $a_\alpha b_\mu \to c_\nu d_\rho$ with momenta assigned as above. The loss kernel follows from the replacements $F \leftrightarrow \widetilde{F}$, and $f \leftrightarrow \widetilde{f}$ for scalar distributions.

There are five processes contributing to neutrino decoupling, with gain kernels (Froustey et al., 2020; Bennett et al., 2021)

$$\begin{aligned}
\mathcal{K}_{\mathcal{G}}^{(\nu e;\nu e)} &= 32 G_F^2 \widetilde{f}_2^{(e)} f_4^{(e)} W_{\nu e}\left[F_3^{(\nu)}\right] , \\
\mathcal{K}_{\mathcal{G}}^{(\nu \bar{e};\nu \bar{e})} &= 32 G_F^2 \widetilde{f}_2^{(e)} f_4^{(e)} W_{\nu \bar{e}}\left[F_3^{(\nu)}\right] , \\
\mathcal{K}_{\mathcal{G}}^{(\nu \bar{\nu};e \bar{e})} &= 32 G_F^2 f_3^{(e)} f_4^{(e)} W_{\nu \bar{\nu}}\left[\left(\widetilde{F}_2^{(\nu)}\right)^\top\right] , \\
\mathcal{K}_{\mathcal{G}}^{(\nu\nu;\nu\nu)} &= 8 G_F^2 s^2 \left(F_3^{(\nu)} \mathcal{T}\left[\widetilde{F}_2^{(\nu)} F_4^{(\nu)}\right] + F_4^{(\nu)} \mathcal{T}\left[\widetilde{F}_2^{(\nu)} F_3^{(\nu)}\right]\right) , \\
\mathcal{K}_{\mathcal{G}}^{(\nu \bar{\nu};\nu \bar{\nu})} &= 8 G_F^2 u^2 \left(\mathcal{T}\left[\left(\widetilde{F}_2^{(\nu)}\right)^\top \left(F_4^{(\nu)}\right)^\top\right] F_3^{(\nu)} + \mathcal{T}\left[F_3^{(\nu)} \left(F_4^{(\nu)}\right)^\top\right] \left(\widetilde{F}_2^{(\nu)}\right)^\top\right) ,
\end{aligned} \tag{43}$$

in terms of the Mandelstam variables $s = (y_1 + y_2)^2$, $t = (y_1 - y_3)^2$, and $u = (y_1 - y_4)^2$, and matrix maps

$$\begin{aligned}
W_{\nu e}[Y] &= (s - m_e^2)^2 G_L Y G_L + (u - m_e^2)^2 G_R Y G_R + m_e^2 t (G_L Y G_R + G_R Y G_L) , \\
W_{\nu \bar{e}}[Y] &= (u - m_e^2)^2 G_L Y G_L + (s - m_e^2)^2 G_R Y G_R + m_e^2 t (G_L Y G_R + G_R Y G_L) , \\
W_{\nu \bar{\nu}}[Y] &= (u - m_e^2)^2 G_L Y G_L + (t - m_e^2)^2 G_R Y G_R + m_e^2 s (G_L Y G_R + G_R Y G_L) .
\end{aligned} \tag{44}$$

The left and right coupling matrices are functions of the Weinberg angle (Weinberg, 1967; Glashow, 1961), θ_W, and the remaining matrix map is

$$G_L = \left(-\frac{1}{2} + \sin^2\theta_W\right)\mathbb{I} + \Pi_e , \qquad G_R = \sin^2\theta_W \mathbb{I} , \qquad \mathcal{T}[Y] = Y + \text{Tr}[Y]\mathbb{I} . \tag{45}$$

Our implementation instead uses the effective low-energy couplings at zero momentum transfer (Kumar et al., 2013; Erler & Su, 2013) following Bennett et al. (2021), $G_L = \text{diag}(g_L^e, g_L^\mu, g_L^\tau)$, $G_R = g_R \mathbb{I}$, see Table 2. All processes have symmetry factor $S^{(r)} = 1$, with the exception of $S^{(\nu\nu;\nu\nu)} = \frac{1}{2}$. The standard collision operator of Eq. (3) is recovered when all F are diagonal. We additionally follow Bennett et al. (2021) and include the momentum-independent part of the electron thermal mass in the collision operator.

Adiabatic transfer of averaged oscillations. Neutrino oscillations proceed over much shorter timescales than cosmic expansion or collisions, allowing several simplifications to the QKEs. Following Froustey et al. (2020), we rewrite the QKE in terms of the instantaneous (mass) eigenstates of the effective Hamiltonian as

$$Hx \frac{\partial F_m}{\partial x} = -i[\mathcal{H}_{\text{diag}}, F_m] - Hx\left[V^\dagger \frac{\partial V}{\partial x}, F_m\right] + V^\dagger \mathcal{C}_\nu[F] V , \tag{46}$$

with V a unitary diagonalising matrix, and where the diagonal Hamiltonian and mass-basis density matrices are

$$\mathcal{H}_{\text{eff}} = V \mathcal{H}_{\text{diag}} V^\dagger , \qquad F_m = V^\dagger F V . \tag{47}$$

Adiabaticity, the slow evolution of the background, and hence the eigenbasis, allows us to neglect the second commutator in Eq. (46). Additionally, as many oscillations occur between successive collisions, complex phases in the off-diagonal components of F_m average to zero. Consequently, we set

$$F_m \to F_{\text{diag}} = \text{diag}(f_1, f_2, f_3) , \tag{48}$$

in terms of the instantaneous mass eigenstate occupations, f_i, $i \in \{1,2,3\}$. The resulting QKE takes the form of a Boltzmann equation

$$Hx\frac{\partial f_i}{\partial x} = (\mathcal{C}_{\nu,\mathrm{diag}})_i[f] \,, \tag{49}$$

with the diagonal collision operator

$$(\mathcal{C}_{\nu,\mathrm{diag}})_i[f] = \frac{1}{2E_1}\sum_r S^{(r)} \int \mathrm{d}\Phi_2 \left[G_i^{(r)}(1-f_i(y_1)) - L_i^{(r)} f_i(y_1)\right] \,,$$
$$G_i^{(r)} = (V^\dagger \mathcal{K}_{\mathcal{G}}^{(r)} V)_{ii} \,, \qquad L_i^{(r)} = (V^\dagger \mathcal{K}_{\mathcal{L}}^{(r)} V)_{ii} \,, \tag{50}$$

where $\mathrm{d}\Phi_2$ is the usual $2 \to 2$ invariant phase-space measure.

The transformation matrix is a function of the evolution variable x and of the momentum y appearing in $\mathcal{H}_{\mathrm{eff}}$, $V = V(x,y)$. Rather than explicitly diagonalising for each momentum value, which would be prohibitively expensive, we instead define $\mathcal{H}'_{\mathrm{eff}} = 2y\mathcal{H}_{\mathrm{eff}}$:

$$\mathcal{H}'_{\mathrm{eff}}(x,y) = A(x) - 2y^2 B(x) \,, \qquad A = U\Delta M^2 U^\dagger \,, \qquad B = \frac{4G_F}{\sqrt{2}m_W^2}(\rho_e + P_e)\Pi_e \,. \tag{51}$$

As rescaling a matrix does not change its eigenvectors, it suffices to find the matrix diagonalising $\mathcal{H}'_{\mathrm{eff}}$. To construct the transformation matrix $V(x,y)$, we first apply a 2×2 block rotation, $W(x)$, to diagonalise the μ-τ components, under which $\Pi_e = \mathrm{diag}(1,0,0)$ is invariant. The resulting matrix, $H(x,y) = W^\dagger(x)\mathcal{H}'_{\mathrm{eff}}(x,y)W(x)$, only has non-zero off-diagonal components in its electron row and column, which are related by Hermiticity across the diagonal. The eigenvalues λ, and eigenvectors u, of H follow from solving the secular equation

$$H_{ee}(y) - \lambda(y) + \frac{|H_{e\mu}|^2}{\lambda(y) - H_{\mu\mu}} + \frac{|H_{e\tau}|^2}{\lambda(y) - H_{\tau\tau}} = 0 \,, \qquad u_i(y) = \frac{1}{\mathcal{N}_i(y)} \begin{pmatrix} 1 \\ \frac{H^*_{e\mu}}{\lambda_i(y) - H_{\mu\mu}} \\ \frac{H^*_{e\tau}}{\lambda_i(y) - H_{\tau\tau}} \end{pmatrix} \,, \tag{52}$$

where all x dependence is implicit and $\mathcal{N}_i$ normalises each u_i. The diagonalising matrix of interest is then constructed as $V(x,y) = W(x)(u_1(x,y), u_2(x,y), u_3(x,y))$.

Flavour-diagonal limit. Neutrino mixing is only expected to give a tiny correction to N_{eff}, as coherent evolution conserves $\mathrm{Tr}[F]$, and consequently ρ_ν in the ultrarelativistic limit. Mixing does not therefore directly transfer energy to or from the neutrino sector, and so can only affect N_{eff} indirectly through redistributing interaction rates. Indeed, Bennett et al. (2021) find a correction $\Delta N_{\mathrm{eff}} = 0.0005$ due to these effects. As such, the flavour-diagonal limit serves as a precision benchmark of NBEs.

In the flavour-diagonal limit, the neutrino density matrix and effective Hamiltonian satisfy

$$F_{\alpha\beta} = f_{\nu_\alpha}\delta_{\alpha\beta} \,, \qquad [\mathcal{H}_{\mathrm{eff}}, F] = 0 \,. \tag{53}$$

It follows that the QKE reduces to the Boltzmann equation

$$Hx\frac{\partial f_{\nu_\alpha}}{\partial x} = (\mathcal{C}_\nu[F])_{\alpha\alpha} \equiv \mathcal{C}_{\nu_\alpha}[f] \,. \tag{54}$$

Identical initial conditions for all three neutrino flavours, assumed here, additionally imply that $f_{\nu_\mu} = f_{\nu_\tau}$. We therefore evolve two independent neutrino distributions, f_{ν_e}, and f_{ν_β}, with $\beta \in \{\mu,\tau\}$.

Equilibrium electromagnetic bath. Electromagnetic interactions are far more efficient than weak interactions, keeping electrons and photons in equilibrium throughout neutrino decoupling. Consequently, the evolution of the combined electromagnetic system is completely described by its temperature, T_{EM}, in the absence of chemical potentials.

The equation governing the evolution of T_{EM} follows from the conservation of the stress-energy tensor, $\nabla_\mu T^{\mu\nu} = 0$, where ∇ denotes the covariant derivative. Splitting the stress-energy tensor into its components for each sector, $T_A^{\mu\nu}$, $A \in \{\mathrm{EM}, \nu\}$, the timelike component of each divergence satisfies

$$\nabla_\mu T^{\mu 0}_{\mathrm{EM}} = Hx\left[\frac{\partial \rho_{\mathrm{EM}}}{\partial x} - \frac{\rho_{\mathrm{EM}} - 3P_{\mathrm{EM}}}{x}\right] \,, \qquad \nabla_\mu T^{\mu 0}_\nu = 2\int \frac{\mathrm{d}^3 y}{(2\pi)^3} y\,\mathrm{Tr}\left[\mathrm{Re}\left[\mathcal{C}_\nu[F]\right]\right] \,, \tag{55}$$

where the latter equation follows from $\rho_\nu = 3P_\nu$, and the combination of Eq. (39) and Eq. (40). The electromagnetic energy density therefore evolves according to

$$\frac{\partial \rho_{\rm EM}}{\partial x} = \frac{\rho_{\rm EM} - 3P_{\rm EM}}{x} - \frac{2}{Hx}\int \frac{{\rm d}^3 y}{(2\pi)^3} y \,{\rm Tr}\left[{\rm Re}\left[\mathcal{C}_\nu[F]\right]\right] \,. \tag{56}$$

The electromagnetic pressure and energy density are decomposed into those of the photon, electron, and finite-temperature QED corrections

$$\rho_{\rm EM} = \rho_\gamma + \rho_e + \sum_{n>0} \rho_{\rm QED}^{(e^n)} \,, \qquad P_{\rm EM} = P_\gamma + P_e + \sum_{n>0} P_{\rm QED}^{(e^n)} \,. \tag{57}$$

We now define the generalised electron energy moment

$$\mathcal{E}_{m,n} = \frac{2}{\pi^2}\int {\rm d}y \, y^2 E_y^m \, \mathcal{F}_e^{(n)} \,, \qquad \mathcal{F}_e^{(0)} = f_e, \qquad \mathcal{F}_e^{(n+1)} = f_e(1-f_e)\frac{\partial \mathcal{F}_e^{(n)}}{\partial f_e} \,, \tag{58}$$

where $E_y = E_e(y)$, such that *e.g.* $\mathcal{E}_{1,0} = \rho_e$, the logarithmic moment

$$\mathcal{L}_n = \int_0^\infty {\rm d}y \int_0^\infty {\rm d}q \, \frac{yq}{E_y E_q} \ln\left|\frac{y+q}{y-q}\right| \sum_{k=0}^{n} \binom{n}{k} E_y^k E_q^{n-k} \mathcal{F}_e^{(k)}(y) \mathcal{F}_e^{(n-k)}(q) \,, \tag{59}$$

and the stress-energy tensor trace, $\Theta_e = \rho_e - 3P_e$. These satisfy the relations

$$\frac{\partial \mathcal{E}_{m,n}}{\partial T_{\rm EM}} = \frac{\mathcal{E}_{m+1,n+1}}{T_{\rm EM}^2} \,, \qquad \frac{\partial \mathcal{L}_n}{\partial T_{\rm EM}} = \frac{\mathcal{L}_{n+1}}{T_{\rm EM}^2} \,, \qquad \frac{\partial \Theta_e}{\partial T_{\rm EM}} = \frac{m_e^2}{T_{\rm EM}^2}\mathcal{E}_{0,1} \,. \tag{60}$$

In terms of these moments, the finite-temperature QED corrections up to $\mathcal{O}(e^3)$ are (Bennett et al., 2020)

$$\begin{aligned} P_{\rm QED}^{(e^2)} &= -\frac{e^2 T_{\rm EM}^2}{12 m_e^2}\Theta_e - \frac{e^2}{8 m_e^4}\Theta_e^2 + \frac{e^2 m_e^2}{4\pi^4}\mathcal{L}_0 \,, \\ P_{\rm QED}^{(e^3)} &= \frac{e^3}{12\pi\sqrt{T_{\rm EM}}}\mathcal{E}_{0,1}^{\frac{3}{2}} \,, \\ \rho_{\rm QED}^{(e^n)} &= T_{\rm EM}\frac{\partial P_{\rm QED}^{(e^n)}}{\partial T_{\rm EM}} - P_{\rm QED}^{(e^n)} \,. \end{aligned} \tag{61}$$

At $\mathcal{O}(e^4)$ and $\mathcal{O}(e^5)$, the pressure corrections are only available for $T_{\rm EM} \gg m_e$, and contain a logarithm of the renormalisation scale, $\bar{\mu}$, and a constant (Arnold & Zhai, 1995; Zhai & Kastening, 1995). We follow Escudero et al. (2026a) and set $\bar{\mu} = m_e$ and relate the logarithmic terms to lower-order contributions,

$$\begin{aligned} P_{\rm QED}^{(e^4)} &= -\frac{e^2}{6\pi^2}\ln\frac{m_e}{T_{\rm EM}}\,\bar{P}_{\rm QED}^{(e^2)} + c_4\, e^4\,\frac{\pi^2 T_{\rm EM}^4}{45} \,, \\ P_{\rm QED}^{(e^5)} &= -\frac{e^2}{4\pi^2}\ln\frac{m_e}{T_{\rm EM}}\,P_{\rm QED}^{(e^3)} + c_5\, e^5\,\frac{\pi^2 T_{\rm EM}^4}{45} \,, \end{aligned} \tag{62}$$

where $\bar{P}_{\rm QED}^{(e^2)}$ denotes $P_{\rm QED}^{(e^2)}$ without the $\mathcal{L}_0$ term, and $c_4 \approx -1.594 \cdot 10^{-3}$, $c_5 \approx -6.972 \cdot 10^{-4}$ are dimensionless constants. To solve Eq. (56), the neural distribution function outputs $T_{\rm EM}$, which we use to construct $\rho_{\rm EM}$ and $P_{\rm EM}$. Network parameters are then updated using the natural gradient update through the energy conservation residual.

C Experiment details

We use the same neural distribution function hyperparameters for all experiments. We use $M = 50$ gamma mixture components, and use $N_{\rm quad} = 16$ or 32 quadrature nodes for the moment evaluation. The network that predicts the mixture parameters ω in conditional runs is a standard multilayer perceptron with 3 layers, 64 channels, and tanh activations. Our implementation uses double precision (float64) throughout.

The matrix inversion for the natural gradient update uses a Tikhonov damping of 10^{-8} on the diagonal, and drops eigenvalues that are lower than a factor 10^{-12} of the largest eigenvalue. We increase the Tikhonov damping to 10^{-5} when inverting the $B \times B$ Gram matrix in conditional trainings. To avoid overshooting in the natural gradient update, we introduce a learning rate γ, with default $\gamma = 1$, which is decreased by a factor of 2 until the update decreases the objective $L(\theta)$ of Eq. (6).

We use the Crank-Nicolson (CN) integrator scheme of Eq. (33) to refine the solver trajectory. The resulting uncertainty estimate is used in a standard PI step size controller to determine the size of the next step, with a specified minimum and maximum step size picked separately for each experiment. The error tolerance involves a Monte Carlo uncertainty floor term in addition to the standard absolute and relative tolerances.

The fixed-point iteration of Algorithm 2 typically converges within 10 iterations for unconditional trainings, and up to 50 iterations for conditional trainings. For conditional trainings, we specify a minimum number of iterations for each step to be conservative. For unconditional trainings, we specify a maximum budget of 30 iterations, but complete the fixed-point iteration earlier if either the residual plateaus below a certain target precision for a specified number of iterations, or if the residual is dominated by Monte Carlo noise for the same number of iterations.

We follow the MadNIS defaults of Heimel et al. (2024) for our normalising flow, using 10 bins, and an MLP with 3 layers and 32 channels per layer. We use the Adam optimiser with learning rate $\gamma = 10^{-3}$ to train the normalising flow.

C.1 Equilibrium distributions

Hyperparameters. Neural distribution functions trained with the natural gradient method follow the description above, whereas Adam trainings use a learning rate of $\gamma = 0.03$ with a cosine schedule, on the same mean-squared-error objective. We use batch size $B = 300$. Training for 100 iterations takes around one second on a GPU.

C.2 Thermalisation

Theory. The formalism of Yoon (2026) maps onto our expanding universe formalism with $H \to 1$, $\log x \to t$, allowing us to reuse our framework without major modifications,

$$\frac{\partial f}{\partial \log x} = \frac{\mathcal{C}[f]}{H} \quad \to \quad \frac{\partial f}{\partial t} = \mathcal{C}[f] \,. \tag{63}$$

We focus on their most challenging case of an $m = 1$ massive particle with the processes $2 \leftrightarrow 2$ and $2 \leftrightarrow 3$. Both processes share the same coupling λ, such that the collision operator is directly proportional to λ^2, $\mathcal{C}[f] = \lambda^2 \mathcal{C}_0[f]$. This allows us to absorb the coupling into the time variable, $t \to t' \equiv \lambda^2 t$:

$$\frac{\partial f}{\partial t} = \lambda^2 \mathcal{C}_0[f] \qquad \to \qquad \mathcal{C}_0[f] = \frac{\partial f}{\lambda^2 \partial t} \equiv \frac{\partial f}{\partial t'} \,. \tag{64}$$

The evolution in terms of t' is therefore independent of the coupling variable λ. We use this property of the model to test the precision of our coupling-conditional training in Figure 2.

Yoon (2026) specifies a non-thermal initial condition, $f_{\text{init}}(y) = (e^{(y-3)/2} + 1)^{-1}$. While evolving the system via the Boltzmann equation, the coupling of different momentum modes in the collision operator pushes the species towards the Bose-Einstein distribution which is the equilibrium configuration. The energy density of the species is conserved in the process. However, the number density can be changed by the $2 \leftrightarrow 3$ process. The known form of the distribution functions at $t = 0$ and $t \to \infty$ combined with energy conservation allows us to compute the asymptotic number density analytically. We find $n_{t\to\infty}/n_{t=0} \approx 0.900272$, and highlight this limit in Figure 2.

We use a log-uniform prior for the condition, $p_{\text{prior}}(\log_{10} \lambda^2) = \mathcal{U}(-0.25, 0.25)$.

Baselines. The semi-analytical and BEST baselines in Figure 2 are evaluated using the public implementation of Yoon (2026). We found that the semi-analytical result is biased at $y \to 0$ and refined their quadrature integration to reduce the shift. The tilt towards larger deviations for the NBE results at $y = 10^{-1}$ is caused by this suboptimal quadrature result. BEST samples the gain and loss terms independently and uses two VEGAS iterations, so it effectively uses $4\times$ more samples

Table 1: Decomposition of $N_{\rm eff}$ contributions, as discussed in the text.

Description	NBE result	$\Delta N_{\rm eff}$
Instantaneous decoupling	3.00000	–
+ ν-$e^{\pm}$ interactions	3.03414	+0.03414
+ spectral distortions	3.03407	−0.00007
+ finite-temperature QED	3.04336	+0.00929
+ flavour oscillations	3.04392	+0.00056

per step compared to our approach for the same number of integral evaluations. The gain factors in Section 5.2 do not include this extra factor to be conservative, and would be $4\times$ larger if the factor was included.

Hyperparameters. We use batch size $B = 1000$ for both pretraining and evolution, and use $N_s = 1.2{\cdot}10^4$ Monte Carlo samples for the evolution in Figure 2. We use $\Delta \log x_{\rm min} = 2.5$, $\Delta \log x_{\rm max} = 25$, and find that the code uses on average 6 fixed-point iterations per step. A full training takes around 2 hours on an A30 GPU.

C.3 Neutrino decoupling

Theory. We discuss the theory behind neutrino decoupling in more detail in Appendix B, and summarise the essential aspects in the following. We assume thermal equilibrium for the electromagnetic plasma, including photons, electrons, and positrons, and evolve their distribution functions following the differential equation for $\rho_{\rm EM}$, Eq. (56). For the three neutrino species, we instead allow a general distribution function, leading to the spectral distortions visualised in Figure 3. If no mixing is included, we evolve the muon and tau neutrinos jointly because they interact through the same processes, and so share a distribution function in the absence of initial asymmetries. We implement neutrino oscillations using the approach of Froustey et al. (2020).

In Table 1 we study the effect of the individual physics contributions on $N_{\rm eff}$ by turning on the contributions one after the other. The non-instantaneous effects on $N_{\rm eff}$ from the neutrino-electron interactions through the neutrino collision integral move $N_{\rm eff}$ away from $N_{\rm eff} = 3$. We keep the Fermi-Dirac equilibrium shape for the second row of Table 1, whereas the following rows use the fully flexible shape that allows them to express spectral distortions. We find that runs with Fermi-Dirac distribution shape carry a larger systematic uncertainty than the runs with fully flexible distribution shape, because they cannot express the target distribution function during training, and the way the network determines the free temperature and chemical potential depends on how the fit is performed. The finite-temperature QED corrections contribute a $+0.01025$ correction at $\mathcal{O}(e^2)$, -0.00095 at $\mathcal{O}(e^3)$, and $+0.00003$ at $\mathcal{O}(e^4) + \mathcal{O}(e^5)$. We also include the electron thermal mass from there on, which contributes a -0.00003 correction. Finally, we include flavour oscillations following the formalism in Appendix B.

Experimental inputs. Our $N_{\rm eff}$ calculation relies on the set of input parameters listed below. For the measured quantity $Q^{\rm phys}$ with mass dimension d, we use $Q = (x/m_0)^d Q^{\rm phys}$, $m_0 = m_e^{\rm phys}$ to obtain the corresponding comoving quantity Q. In Table 2 we list all parameters where uncertainties can be neglected, following Bennett et al. (2021).

The neutrino sector parameters with 1σ uncertainties are taken from the more recent NuFIT-6.1 fit (Esteban et al., 2024; 2025), and we assume $\delta_{\rm CP} = 0$ in the PMNS matrix,

$$
\begin{aligned}
& \sin^2\theta_{12} = 0.3088^{+0.0067}_{-0.0066}\,, \quad \Delta m^{2,\rm phys}_{21} = 7.537^{+0.094}_{-0.10}\cdot 10^{-5}\,\mathrm{eV}^2\,, \\
\mathrm{NO}: \quad & \sin^2\theta_{13} = 0.02249 \pm 0.00057\,, \quad \sin^2\theta_{23} = 0.470^{+0.017}_{-0.014}\,, \\
& \Delta m^{2,\rm phys}_{31} = 2.521^{+0.026}_{-0.018}\cdot 10^{-3}\,\mathrm{eV}^2\,, \\
\mathrm{IO}: \quad & \sin^2\theta_{13} = 0.02261 \pm 0.00056\,, \quad \sin^2\theta_{23} = 0.555^{+0.013}_{-0.016}\,, \\
& \Delta m^{2,\rm phys}_{32} = -2.500^{+0.024}_{-0.023}\cdot 10^{-3}\,\mathrm{eV}^2\,.
\end{aligned}
\tag{65}
$$

Table 2: Experimental inputs used in the N_{eff} precision calculation.

Quantity	Value	Reference
α	$1/137.035999084$	Tiesinga et al. (2021)
m_e^{phys}	$0.51099895\,\text{MeV}$	Tiesinga et al. (2021)
G_F^{phys}	$1.1663787 \cdot 10^{-5}\,\text{GeV}^{-2}$	Zyla et al. (2020)
m_W^{phys}	$80.379\,\text{GeV}$	Zyla et al. (2020)
$m_{\text{Pl}}^{\text{phys}}$	$1.22089 \cdot 10^{19}\,\text{GeV}$	Zyla et al. (2020)
g_L^e	0.727	Kumar et al. (2013); Erler & Su (2013)
$g_L^\mu = g_L^\tau$	-0.273	Kumar et al. (2013); Erler & Su (2013)
g_R	0.233	Kumar et al. (2013); Erler & Su (2013)

Hyperparameters. We use a batch size $B = 4000$, and $N_s = 4000$ Monte Carlo samples for the collision integral. We evolve the system from $x = 10^{-2}$ to $\log x = 5.0$. We use a minimum step size of $\Delta \log x = 0.0025$, a maximum step size of $\Delta \log x = 0.025$, alongside relative and absolute tolerances of $\epsilon_{\text{rel}} = 10^{-6}$ and $\epsilon_{\text{abs}} = 10^{-8}$, respectively, for the step size controller. For each training, we iterate on the fixed-point objective until the objective improves by less than $r_{\text{plateau}} = 0.05$ over 3 iterations, and terminate after 30 iterations at most. We use 64 quadrature nodes for the electron moment integrals and 32 quadrature nodes for the neutrinos. We do not use the MadNIS neural importance sampling refinement for this calculation. A single training with these settings takes around 4 hours on an NVIDIA A30 GPU.

Solver uncertainty. Here we describe how we have quantified the solver uncertainty, which is significantly smaller for NBEs as compared to existing approaches, and as such a key result of our work. We first performed $n = 20$ trainings on independent seeds with the default hyperparameters to extract the mean, μ_0, and per-run standard deviation, σ_0, reference values for N_{eff}:

$$\mu_0 = 3.0439206\,, \qquad \sigma_0 = 1.3 \cdot 10^{-6}\,. \tag{66}$$

Any digits beyond these varied with different seeds or hyperparameters, and so we do not report them in our main result, Eq. (17). We next varied each of the hyperparameters above to increase precision, and estimated uncertainties based on the mean $\mu_{\delta,i}$ and per-run standard deviation $\sigma_{\delta,i}$ from $m = 6$ runs. Based on these values, we extract the statistical uncertainty σ_{stat} as the standard deviation of the mean μ_0 and the systematic uncertainties $\sigma_{\text{syst},i}$ from varying each of the hyperparameters,

$$\sigma_{\text{stat}} = \frac{\sigma_0}{\sqrt{n}}\,, \qquad \sigma_{\text{syst},i} = \max\left(|\mu_0 - \mu_{\delta,i}|, \sigma_{\delta,i}\sqrt{1/n + 1/m}\right)\,. \tag{67}$$

The systematic effects are summarised in Table 3. For Monte Carlo sampling, we vary the batch size $4000 \to 8000$ and the number of collision integral samples $4000 \to 8000$. We increase the number of quadrature nodes by a factor of 2. We start the evolution at $x = 10^{-2} \to 10^{-3}$. For the step size controller, we decrease the minimum step size $\Delta \log x_{\min} = 0.0025 \to 0.000625$, the maximum step size $\Delta \log x_{\max} = 0.025 \to 0.00625$, and the tolerances $\epsilon_{\text{rel}} = 10^{-6} \to 4 \cdot 10^{-7}$, $\epsilon_{\text{abs}} = 10^{-8} \to 2.5 \cdot 10^{-9}$. For the fixed-point iteration convergence criterion, we increase the maximum number of iterations $30 \to 60$ and decrease the early exit plateau $r_{\text{plateau}} = 0.05 \to 0.0125$. All hyperparameters are varied independently and added in quadrature, $\sigma_{\text{tot}} = (\sum_i \sigma_i^2)^{1/2}$, which corresponds to the assumption of independent Gaussian uncertainties. For the step size controller and fixed-point iteration convergence values in Table 3 we already combined several individual effects. Finally, we combine all statistical and systematic effects, and obtain a total solver uncertainty of $2.0 \cdot 10^{-6}$. To further validate this procedure, we repeat the procedure for a lower-precision calculation in Appendix D.5 and find that this uncertainty estimate covers the result of the higher-precision calculation performed here.

Experimental uncertainty. We now study how uncertainties from the NuFIT-6.1 global fits on neutrino mixing angles $\sin^2\theta_{ij}$ and mass differences Δm_{ij}^2 propagate into uncertainties on the N_{eff} determination. To this end, we follow Bennett et al. (2021) and run solvers with input parameters varied by 3σ upwards and downwards. We then average the two resulting shifts in N_{eff} to obtain an uncertainty estimate. We also perform one run with inverted neutrino mass ordering, finding a one-sided shift. Table 4 shows the resulting uncertainties, which are dominated by the solar mixing

Table 3: Contributions to the $\pm 2 \cdot 10^{-6}$ solver uncertainty in the $N_{\rm eff}$ calculation in Eq. (17). We quantify the effects by increasing the precision for every parameter individually, and combine the resulting uncertainties as independent Gaussian effects.

Effect	$10^6 \cdot \delta N_{\rm eff}$
Seed statistics	± 0.3
Monte Carlo sampling (batch size)	± 0.6
Monte Carlo sampling (collision integral)	± 0.6
Moment quadrature	± 0.7
Evolution starting point	± 0.6
Step size controller	± 1.4
Fixed-point iteration convergence	± 0.6
Total	± 2.0

Table 4: Contributions to the $\pm 2 \cdot 10^{-5}$ experimental uncertainty in the $N_{\rm eff}$ calculation in Eq. (17). We quantify the effects by varying the input parameters within their reported experimental uncertainty band, and for the neutrino mass ordering we report the effect of flipping the choice.

Effect	$10^6 \cdot \delta N_{\rm eff}$
$\sin^2\theta_{12}$ (3σ band)	± 18
$\sin^2\theta_{13}$ (3σ band)	± 4
Δm^2_{21} (3σ band)	± 3
Neutrino mass ordering (NO $\to$ IO)	$+7$
Total	± 20

angle θ_{12}. This is in agreement with Bennett et al. (2021), where the authors found that the dominating uncertainty comes from the solar mixing angle, although as a 10^{-4} effect due to the larger experimental uncertainty on θ_{12} when the study was performed.

Theory uncertainty. On the theory side, there are several effects that are expected to yield contributions at the level of experimental uncertainty, $2 \cdot 10^{-5}$, or above. Quantifying this uncertainty is challenging, as it ultimately requires the full calculation. A first estimate can be obtained from approximate calculations, however we do not include these estimated effects as uncertainties in the same way as the solver or experimental uncertainties as they strongly depend on the quality of these approximations, and instead discuss them here. Most importantly, QED corrections to the neutrino collision integral are expected to contribute at the 10^{-4} level, based on approximations ranging from $7 \cdot 10^{-4}$ (Cielo et al., 2023) to $2 \cdot 10^{-5}$ (Drewes et al., 2024). Binder et al. (2025) estimate the impact of positronium formation and find corrections to $N_{\rm eff}$ at the level of 10^{-4} to 10^{-6} depending on the prescription. Finite-temperature corrections to the QED equation of state are known up to $\mathcal{O}(e^3)$ including electron mass effects, and up to $\mathcal{O}(e^5)$ in the $T \gg m_e$ limit, see Appendix B. Finally, our approximation of adiabatic neutrino oscillations in Appendix B is estimated to affect the value of $N_{\rm eff}$ below the 10^{-6} level. We emphasise that the significantly reduced solver uncertainty of the NBE approach opens the door for probing many of these precision effects.

Comparison with literature results. We now discuss the differences between our approach and the implementation of Bennett et al. (2021), which we used as a reference. First, our flavour oscillation treatment follows Froustey et al. (2020), see Appendix B, which induces a deviation at the 10^{-6} level or below. Second, we include finite-temperature QED corrections up to $\mathcal{O}(e^5)$, instead of only $\mathcal{O}(e^3)$, see Appendix B. Third, we use the recent NuFIT-6.1 values for the neutrino masses and mixing angles, with significantly reduced uncertainties compared to the values used by Bennett et al. (2021), translating into reduced uncertainties in $N_{\rm eff}$. As a check, we repeat the $N_{\rm eff}$ calculation following the Bennett et al. (2021) choices in all the cases above except the flavour oscillations, finding

$$N_{\rm eff} = 3.04390 \ \pm 2 \cdot 10^{-6}\,(\text{solver}) \ \pm 4 \cdot 10^{-5}\,(\text{experiment})\ . \tag{68}$$

This is in agreement with the $N_{\text{eff}} = 3.0440 \pm 0.0001(\text{solver}) \pm 0.0001(\text{experiment})$ result of Bennett et al. (2021). However, we find a smaller experimental uncertainty from varying $\sin^2\theta_{12} = 0.318 \pm 0.048$ on a 3σ band.

New physics parameter scan. The new physics parameter scan in Figure 3 does not require the 10^{-6} precision of the calculation described above, here 10^{-2} precision is sufficient to understand whether the new theory with dark photon A' and complex scalar ϕ can shift the value of N_{eff} by the correct amount. We therefore use the 'fast' hyperparameters of Appendix D.5 for this study, see Table 5. We use a uniform prior for $r \in [2.01, 2.7]$ and a log-uniform prior for $m_\phi \in [1, 20]\,\text{MeV}$. The conditional training requires an accurate evolution of the neural distribution function over the full prior range, and we therefore additionally increase the batch size to 1000. We also require at least 15 iterations per step to ensure that the prior space is sufficiently explored. A single run with these settings takes around 5 hours on an NVIDIA A30 GPU.

We validated the results of the conditional run by comparing with unconditional runs using the 'fast' hyperparameters of Table 5, which are expected to have a solver uncertainty of $\delta N_{\text{eff}} \approx 10^{-5}$. The N_{eff} results of the conditional training vary from this reference by 10^{-3} to 10^{-4} depending on the theory parameters, which is sufficient for phenomenological studies.

D Additional results

D.1 Number of mixture components in neural distribution functions

Fermionic and bosonic equilibrium distribution functions for massless particles can be exactly represented using the gamma mixture series in Section 4.1, using a single mixture component for fermions and an infinite number of mixture components for bosons, see Appendix A.1. We visualise the bosonic case in Figure 4, where we show the exact Bose-Einstein shape compared with the shapes obtained by truncating the gamma mixture series after M terms. The series converges slowly for small y values due to the $e^{-k\varepsilon}$ term in the series representation, Eq. (19).

In Figure 5 we show the neural distribution functions learned with NBE, for the four cases of fermionic/bosonic and massless/massive particles in equilibrium, extending the left panel of Figure 1. We follow the same training as in Section 5.1, described in Appendix C. We vary the number of mixture components between $M = 3$ and our default $M = 50$, and compare to the truth distribution. We find that $M = 3$ already achieves 10^{-4} precision, and the precision plateaus around $M = 10$, with no significant improvements when going to our default $M = 50$ in this case. For massless fermions which can be expressed with a single mixture component, we instead find that $M = 3$ achieves the highest precision at 10^{-10}, and more mixture components degrade to 10^{-6} due to the noise in the higher-dimensional matrix inversion required for the natural gradient step. Compared to Figure 4, we see that neural distribution functions do not learn the gamma mixture series representation of the distribution function which converges slowly for small y, but a representation with the highest precision at intermediate y values, which receive the largest weight in the

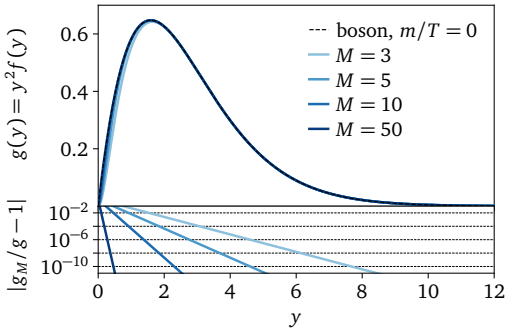

Figure 4: Convergence speed of the gamma mixture series for massless bosons. We compare the exact Bose-Einstein distribution with the shapes obtained after truncating the series after M terms.

objective. For $M = 3$, the learned representation in Figure 5 is on average more precise than the series representation in Figure 4.

D.2 Longer Adam training for equilibrium distributions

Section 5.1 discusses the benefits of natural gradient training for neural distribution functions compared to training with the Adam optimiser. We find in Figure 1 that Adam training converges slowly, whereas natural gradient training reaches a plateau at 10^{-6} deviation within 10 iterations. To understand the limitations of the Adam optimiser for this problem, we perform a longer training with 10,000 instead of 100 iterations, see the left part of Figure 6. We find that Adam then reaches 10^{-4} deviation, still significantly above the value reached by natural gradient after only 10 iterations.

D.3 Gain from neural importance sampling

Neural importance sampling can be used to refine the Monte Carlo samples in Algorithm 1. We study the gain from neural importance sampling for the thermalisation experiment, training on a single coupling and otherwise using the same hyperparameters as Section 5.2, described in Appendix C.

We study the gain of neural importance sampling over the thermalisation process for the full $\phi\phi \leftrightarrow \phi\phi$ and $\phi\phi \leftrightarrow \phi\phi\phi$ interactions in the right part of Figure 6. In the top panel, we show the effective sample gain compared to the NBE baseline obtained by requiring $\sigma/\sqrt{N} = \sigma_{\text{NBE}}/\sqrt{N_{\text{NBE}}}$, to get effective sample gain $G = N_{\text{NBE}}/N$. Whereas we find only $G \approx 5$ at $t = 0$, the gain G increases to $G \approx 20$ while the system thermalises. This is due to the change of the integrand and neural distribution function density, which modify the gap between the default and optimal proposal density that can be closed by neural importance sampling. In the lower panel of Figure 6, we study the unweighting efficiency $\langle w \rangle / w_{\text{max}}$ as a direct measure of the distribution of integral weights w. We see the same $10\times$ increase in sample efficiency as in the first panel.

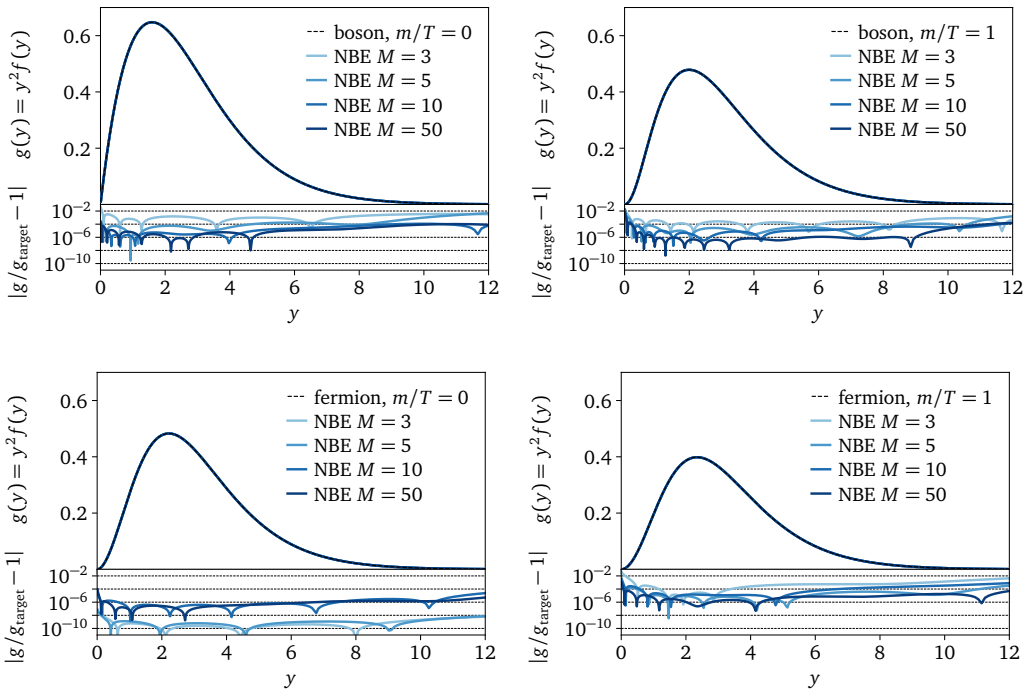

Figure 5: Trained neural distribution functions with different numbers of mixture components M, for massless and massive fermions and bosons.

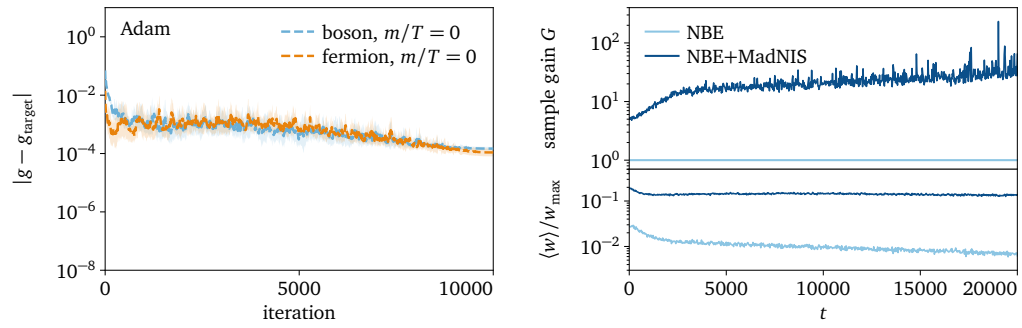

Figure 6: Left: Longer Adam training for massless equilibrium distribution functions. Right: Benefit of neural importance sampling for the thermalisation experiment in Section 5.2, comparing NBE with and without the neural importance sampling refinement using MadNIS.

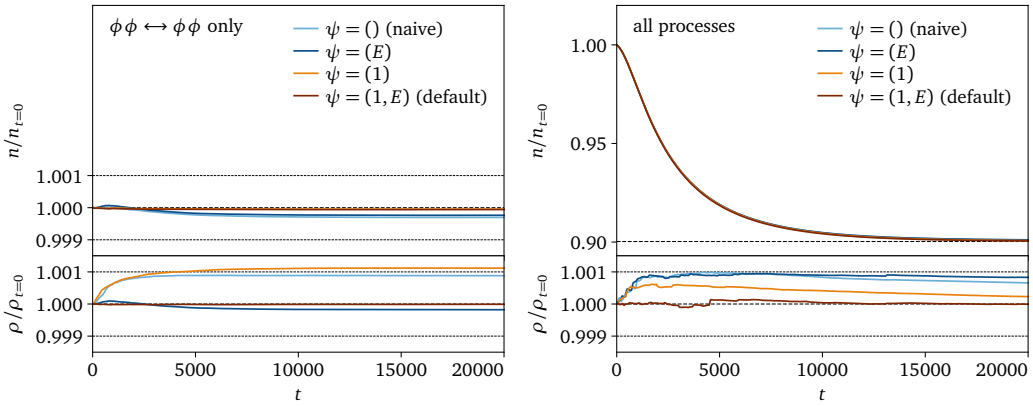

Figure 7: Effect of the moment conservation trick of Appendix A.3 for scalar thermalisation, including only $2 \leftrightarrow 2$ processes (left) and also $2 \leftrightarrow 3$ processes (right).

D.4 Natural gradient update with conserved moments

In Appendix A.3 we discussed a modification tothe natural gradient update that improves the handling of conserved moments. We test the effect of this trick by evolving the thermalisation experiment of Section 5.2 with different choices of conserved moments $\psi = (\psi_1, \dots \psi_n)$, see Appendix A.3 for the formalism. The naive baseline $\psi = ()$ does not use the trick, and the extensions $\psi = (E)$, $\psi = (1)$, $\psi = (1, E)$ improve the handling of energy density conservation, number density conservation, and both. The left panel of Figure 7 includes only the $\phi\phi \leftrightarrow \phi\phi$ process, such that both energy and number density are conserved, whereas the right panel includes also $\phi\phi \leftrightarrow \phi\phi\phi$ processes such that number density is no longer conserved. For both systems, we find that the $\psi = (1, E)$ run achieves the highest precision, with 10^{-4} or lower relative deviation, whereas the naive approach only achieves 10^{-3} deviation.

D.5 Fast neutrino decoupling solver

To validate the solver uncertainty quantification procedure used in Appendix C.3, we repeat the procedure for a calculation with hyperparameters that achieve reduced precision. We can then use the result of the precision calculation to validate the uncertainty estimate of the lower-precision calculations.

We specify the default and varied parameters in Table 5, with 'precision' for our defaults used in Appendix C.3, and 'fast' for the additional study in this section. A 'precision' run takes around 4 hours on an NVIDIA A30 GPU, whereas a 'fast' run only takes 40 minutes. Both runs use the experimental uncertainty and the same assumptions from the theory side. We follow the procedure of Appendix C.3 to obtain the solver uncertainty estimates, and find the results in Table 6. In total,

Table 5: Solver settings varied for the uncertainty estimate of Table 6. We write $a \to b$ for each entry, with a the default value and b the increased-precision value used for the solver uncertainty estimate.

Setting	precision	fast
Batch size	$4000 \to 8000$	$500 \to 1000$
Collision integral samples	$4000 \to 8000$	$4000 \to 8000$
Neutrino moment quadrature	$32 \to 64$	same
EM-sector moment quadrature	$64 \to 128$	same
Photon series truncation	$1000 \to 2000$	same
QED two-dimensional quadrature	$32 \to 64$	same
$x_{\min}$	$10^{-2} \to 10^{-3}$	same
$\Delta \log x_{\max}$	$0.025 \to 0.00625$	same
$\Delta \log x_{\min}$	$0.0025 \to 6.25 \cdot 10^{-4}$	$0.005 \to 1.25 \cdot 10^{-3}$
Relative tolerance ϵ_{rel}	$10^{-6} \to 4 \cdot 10^{-7}$	$6.25 \cdot 10^{-6} \to 1.5625 \cdot 10^{-6}$
Absolute tolerance ϵ_{abs}	$10^{-8} \to 2.5 \cdot 10^{-9}$	same
Fixed-point plateau threshold r_{plateau}	$0.05 \to 0.0125$	same
Fixed-point maximum iterations	$30 \to 60$	same

Table 6: Contributions to the solver uncertainty in the N_{eff} calculation, for the two solver settings of Table 5. We quantify the effects by increasing the precision for every parameter individually and combine the resulting uncertainties as independent Gaussian effects.

Effect	$10^6 \cdot \delta N_{\mathrm{eff}}$ (precision)	$10^6 \cdot \delta N_{\mathrm{eff}}$ (fast)
Seed statistics	± 0.3	± 1.1
Monte Carlo sampling (batch size)	± 0.6	± 3.4
Monte Carlo sampling (collision integral)	± 0.6	± 2.4
Moment quadrature	± 0.7	± 4.9
Evolution starting point	± 0.6	± 2.4
Step size controller	± 1.4	± 6.5
Fixed-point iteration convergence	± 0.6	± 2.4
Total	± 2.0	± 9.8

we find

$$\begin{aligned} N_{\mathrm{eff}} &= 3.043915 \ \pm 0.000010\,(\mathrm{solver}) \quad (\mathrm{fast})\,, \\ N_{\mathrm{eff}} &= 3.043921 \ \pm 0.000002\,(\mathrm{solver}) \quad (\mathrm{precision})\,. \end{aligned} \tag{69}$$

The solver uncertainty of the fast run covers the result of the more precise calculation, validating the uncertainty estimate procedure of Appendix C.3.