Normalizing Flows for Microscopic Many-Body Calculations: An Application to the Nuclear Equation of State

Jack Brady,[1,*] Pengsheng Wen,[2,3,†] and Jeremy W. Holt[2,3,‡]

[1]*Texas A&M University, College Station, Texas 77843, USA*
[2]*Cyclotron Institute, Texas A&M University, College Station, Texas 77843, USA*
[3]*Department of Physics and Astronomy, Texas A&M University, College Station, Texas 77843, USA*

(Received 8 February 2021; revised 21 May 2021; accepted 29 June 2021; published 3 August 2021)

Normalizing flows are a class of machine learning models used to construct a complex distribution through a bijective mapping of a simple base distribution. We demonstrate that normalizing flows are particularly well suited as a Monte Carlo integration framework for quantum many-body calculations that require the repeated evaluation of high-dimensional integrals across smoothly varying integrands and integration regions. As an example, we consider the finite-temperature nuclear equation of state. An important advantage of normalizing flows is the ability to build highly expressive models of the target integrand, which we demonstrate enables precise evaluations of the nuclear free energy and its derivatives. Furthermore, we show that a normalizing flow model trained on one target integrand can be used to efficiently calculate related integrals when the temperature, density, or nuclear force is varied. This work will support future efforts to build microscopic equations of state for numerical simulations of supernovae and neutron star mergers that employ state-of-the-art nuclear forces and many-body methods.

DOI: 10.1103/PhysRevLett.127.062701

Introduction.—The hot and dense matter equation of state (EOS) is of fundamental importance for interpreting observations of neutron stars, core-collapse supernovae, and neutron star mergers in terms of the underlying nuclear microphysics [1,2]. Because of the complexity of computing the free energy and its derivatives (to obtain the pressure, entropy, chemical potentials, etc.) across the wide range of ambient conditions encountered during simulations of supernovae and neutron star mergers, most equations of state in wide use by the simulation community are based on simplified mean field models of the nuclear force [3–6]. Since mean field theory is grounded in effective interactions fitted to the bulk properties of medium-mass and heavy nuclei, one loses connection to fundamental nuclear two- and many-body forces and the ability to estimate systematic uncertainties [7]. In addition, certain thermodynamic properties that are important for understanding the evolution of core-collapse supernovae, such as the temperature-dependent nucleon effective mass [8,9], are quite different in microscopic and mean field models [10]. For these reasons, there is strong motivation to develop more microscopic descriptions of the nuclear equation of state based on realistic nuclear forces in beyond-mean-field-theory quantum many-body calculations.

Microscopic calculations of the free energy $F(n, T, Y_p)$ as a function of density n, temperature T, and composition (e.g., the proton fraction Y_p) have in recent years been computed from realistic two- and three-body chiral effective field theory (EFT) nuclear forces [11–13]. However, the inclusion of the most sophisticated three-body forces [14,15] and important high-order many-body perturbation theory corrections (such as third-order particle-hole diagrams) [16,17] require the evaluation of technically challenging multidimensional integrations and therefore have not yet been achieved in finite-temperature calculations. Moreover, the tabulation of an astrophysical equation of state (the free energy and its first and second derivatives) involves the repeated evaluation of these integrals across more than 1 000 000 phase space points in order to ensure numerical stability of supernova and neutron star merger simulations [18]. Microscopic EOS tabulations suitable for astrophysical simulations are therefore computationally demanding and only recently have been carried out [19,20] using the Argonne v_{18} NN potential and the Urbana IX three-body force, supplemented by a liquid drop model for describing the low-density inhomogeneous phase of nuclear matter.

A potential solution to the numerical challenges outlined above is adaptive Monte Carlo methods based on importance sampling [21], which have recently been employed [17] to calculate high-order perturbation theory corrections to the cold dense matter equation of state that were previously intractable. In importance sampling an estimate for the integral

$$I[\psi] = \int_D \psi(\vec{x}) d\vec{x} \tag{1}$$

is obtained as a Monte Carlo estimate under a proposal distribution $p(x)$ as

$$I \simeq \langle I \rangle_N = \frac{1}{N}\sum_{i=1}^{N} \frac{\psi(\vec{x}_i)}{p(\vec{x}_i)}. \tag{2}$$

The precision of this estimator, however, is dependent on how well $p(x)$ is able to match the normalized target $|\psi(\vec{x})|/\tilde{I}$, where $\tilde{I} \equiv I[|\psi|]$. In particular, if $p(x)$ matches $|\psi(\vec{x})|/\tilde{I}$ exactly, we obtain an ideal estimator [22]. Consequently, when a precise estimate for the integral is required, as is the case when computing numerical derivatives in EOS tabulations, the proposal distribution must have sufficient expressive capacity to match the target. Popular adaptive importance sampling methods [23,24], however, often make restrictive assumptions on the integrand such as factorizability, thus limiting the precision of the estimator.

In the present work, we leverage normalizing flows [25–27] as a means for constructing efficient importance sampling estimators for microscopic EOS tabulations. Normalizing flows have recently emerged as a highly expressive method for modeling complex proposal distributions [28–37] using deep neural networks. We demonstrate that this expressivity allows for precise first- and second-order numerical derivatives of the free energy over a relatively coarse density and temperature grid. Furthermore, we show that a normalizing flow model trained on one target integrand transfers remarkably when the density, temperature, or nuclear force is varied, thus providing a compelling framework moving forward for including high-order many-body perturbation theory corrections for tabulated astrophysical equations of state and assessing associated uncertainties. Although the focus of the present work is on the nuclear matter equation of state, we note that normalizing flow based importance sampling could also be applied in condensed matter physics and related fields, where many-body perturbation theory has recently received renewed interest [38,39].

As a concrete test case, we consider the second-order perturbation theory contribution to the grand canonical potential Ω of isospin-symmetric nuclear matter from an antisymmetrized two-body force $\bar{V}_{NN}$:

$$\Omega^{(2)} = -\frac{1}{8}\sum_{1234} \bar{V}_{NN}^{12;34}\bar{V}_{NN}^{34;12}\frac{f_1 f_2 \bar{f}_3 \bar{f}_4 - \bar{f}_1 \bar{f}_2 f_3 f_4}{\epsilon_3 + \epsilon_4 - \epsilon_1 - \epsilon_2}, \tag{3}$$

where $f_i = 1/(1 + e^{(\epsilon_i - \mu)/T})$ is the Fermi-Dirac distribution function for particles with chemical potential μ, $\bar{f}_i = 1 - f_i$, $\epsilon_i = k_i^2/(2M)$ is the free-particle spectrum, and the sums are taken over spin, isospin, and momentum. On the one hand, the contribution in Eq. (3) is sufficiently complex to demonstrate the efficiency of normalizing flow based importance sampling, and on the other hand it is amenable to nearly exact evaluation using Gaussian quadrature for benchmarking our results. Since our focus is on the momentum-space integrations inherent in Eq. (3), we begin with a particularly simple model of the nuclear force

$$V(q) = \frac{g^2}{m_\phi^2 + q^2} \tag{4}$$

associated with scalar-isoscalar boson exchange, where q is the magnitude of the momentum transfer, and we take $m_\phi = 600$ MeV and $g = 1$. Performing the spin and isospin sums in Eq. (3) and choosing $\vec{k}_3 = k_3 \hat{i}_z$, we obtain

$$\Omega^{(2)}(n,T) = -\frac{Mg^4}{64\pi^8}\int_0^\infty dk_1 \int_0^\infty dk_2 \int_0^\infty dk_3 \int_0^\pi d\theta_1 \int_0^\pi d\theta_2 \int_0^{2\pi} d\phi_1 \int_0^{2\pi} d\phi_2 \frac{k_1^2 k_2^2 k_3^2 \sin\theta_1 \sin\theta_2 (f_1 f_2 \bar{f}_3 \bar{f}_4 - \bar{f}_1 \bar{f}_2 f_3 f_4)}{k_1^2 + k_2^2 - k_3^2 - k_4^2} \\ \times \left[\frac{4}{(m_\phi^2 + q_1^2)^2} - \frac{1}{(m_\phi^2 + q_1^2)(m_\phi^2 + q_2^2)}\right] e^{-2(p/\Lambda)^6 - 2(p'/\Lambda)^6}, \tag{5}$$

where $\vec{k}_4 = \vec{k}_1 + \vec{k}_2 - \vec{k}_3$, $\vec{q}_1 = \vec{k}_1 - \vec{k}_3$, $\vec{q}_2 = \vec{k}_1 - \vec{k}_4$, $\vec{p} = \frac{1}{2}(\vec{k}_1 - \vec{k}_2)$, and $\vec{p}' = \frac{1}{2}(\vec{k}_3 - \vec{k}_4)$. We have included the multiplicative function $g(\vec{p}, \vec{p}') = e^{-(\vec{p}/\Lambda)^6 - (\vec{p}'/\Lambda)^6}$ in the definition of the potential in Eq. (4) as is common in the literature to regulate the unresolved high-momentum components of chiral nuclear forces [40]. We choose $\Lambda = 450$ MeV as the high-momentum cutoff scale. In practice, we have replaced the upper integration limits of ∞ with $k_{\max} = 2\Lambda$, which we have tested is sufficient to achieve converged results.

Methods.—A normalizing flow [25–27] defines a complex distribution $p(\vec{x})$ by applying a learnable, bijective mapping $\vec{h}$ to a simple base distribution $\pi(\vec{u})$. The probability density of a sample $\vec{x} \coloneqq \vec{h}(\vec{u})$ under a flow can then be obtained analytically using the change of variables formula

$$p(\vec{x}) = \pi(\vec{h}^{-1}(\vec{x}))\left|\det\left(\frac{\partial \vec{h}^{-1}}{\partial \vec{x}}\right)\right|. \tag{6}$$

In the present case, $\vec{h}$ should transform $\pi(\vec{u})$ such that the resulting distribution $p(\vec{x})$ matches our target distribution as closely as possible. This can be achieved by optimizing the parameters of $\vec{h}$ using gradient-based methods to minimize a suitable divergence metric between our model and target distributions. To optimize this objective in practice, however, certain conditions must be satisfied when choosing a parametrization for $\vec{h}$.

First, the transformation $\vec{h}$ must have sufficient expressive capacity to model the target distribution and at the same time have a Jacobian determinant that is tractable to compute. To satisfy these requirements, we implement $\vec{h}$ using a sequence of coupling transforms [41,42]. For a given coupling transform $\vec{\phi}$, an n-dimensional vector $\vec{x}$ is first partitioned into two parts: $\vec{x} = (x_1, \ldots, x_d, 0, \ldots 0) + (0, \ldots, 0, x_{d+1}, \ldots, x_n)$, which we refer to as the base input and updated input vectors, respectively. The d coordinates of the base input vector are then passed through a neural network which outputs a set of parameters $\vec{\theta}$ that define an invertible transformation $g_{\theta_i}(x_i)$ dimensionwise on the $n - d$ updated input coordinates x_i. The coordinates of the base input vector $x_1, \ldots, x_d$ are then passed unchanged through the coupling transform: $x_i \to x_i$, for $i \leq d$. This results in a lower-triangular Jacobian matrix where the determinant can easily be computed as the product of the diagonals:

$$\det\left(\frac{\partial \vec{\phi}}{\partial \vec{x}}\right) = \prod_{i=d}^{n} \frac{\partial g_{\theta_i}}{\partial x_i}. \tag{7}$$

Furthermore, $\vec{h}$ can be made highly expressive by composing a sequence of k coupling transformations $\vec{h} \coloneqq \vec{\phi}_k \circ \ldots \circ \vec{\phi}_1$ with different choices of base input and updated input coordinates such that all variables are allowed to interact.

The second condition $\vec{h}$ must satisfy is that its output range needs to respect the specified boundary conditions for a given integral. For this purpose, we choose to implement each coupling transform using rational-quadratic spline flows [31,43,44], which define $g_{\theta_i}(x_i)$ piecewise on an interval $[A_i, B_i]$ by partitioning the interval into K bins and defining the transformation in each bin as a monotonic rational-quadratic function. The rational-quadratic functions are parametrized by a set of $K + 1$ knots $\{(x^{(k)}, y^{(k)})\}_{k=0}^{K}$, which define the boundaries for the domain and range of each transformation, and a set of $K + 1$ derivatives $\{\delta^{(k)}\}_{k=0}^{K}$ defined at each knot. The knots monotonically increase within the interval $[A_i, B_i]$ where $(x_i^{(0)}, y_i^{(0)}) = (A_i, A_i)$ and $(x_i^{(K)}, y_i^{(K)}) = (B_i, B_i)$ such that $g_{\theta_i}(x_i)$ is a mapping from $[A_i, B_i]$ to $[A_i, B_i]$. Thus, by setting $[A_i, B_i]$ to be the boundaries for each dimension of a given integral, we can restrict $\vec{h}$ to only be defined on the integration region.

We implement our flow using a composition of 6 rational-quadratic spline coupling transforms which is the minimum number required to account for correlations among all the variables in our seven-dimensional integral in Eq. (5) [32]. For each transform, we use $K = 16$ bins and implement each neural network using a residual network [45] with two residual blocks and 32 hidden features. Our base distribution is chosen to be uniform over the integration region for each dimension, respectively. To train our flow, we minimize the Pearson χ^2 divergence between our model distribution $p(\vec{x})$ and target distribution $|\psi(\vec{x})|/\tilde{I}$. This divergence is estimated as an expectation under our model through importance sampling as

$$D_{\chi^2} \simeq \langle D_{\chi^2} \rangle_N = \frac{1}{N} \sum_{i=1}^{N} \frac{\left[\frac{|\psi(\vec{x}_i)|}{\tilde{I}} - p(\vec{x}_i)\right]^2}{p(\vec{x}_i)} / p(\vec{x}_i), \tag{8}$$

where the normalizing constant $\tilde{I}$ is additionally estimated through sampling. At each training iteration, we minimize this expectation with respect to the parameters of our flow using gradient descent on batches of N samples, where sampling from our flow amounts to first sampling from the base distribution $\pi(\vec{u})$ and then passing these samples through $\vec{h}$ to obtain $\vec{x}$. The gradient descent optimization algorithm we employ is Adam [46]. All models were implemented using PyTorch [47], and the open-source implementation for the spline transformations in Ref. [44] was used for our coupling layers.

Results.—We start by training the flow on the target integral in Eq. (5) using batches of $N = 5000$ samples drawn randomly at each iteration from our base distribution and passed through our model. We initially fix the density at $n = n_0$, where $n_0 = 0.16\ \mathrm{fm}^{-3}$ is the saturation density of nuclear matter, and the temperature at $T = 25$ MeV. The learning rate for the Adam optimizer was set to 10^{-3} until 200 iterations passed without an improvement in the χ^2 loss function, at which point a cosine scheduler was initiated with maximum learning rate 10^{-3}, minimum learning rate 10^{-4}, and period of 200 iterations. For comparison, we have computed the integral in Eq. (5) using the adaptive Monte Carlo integrators Divonne, Suave, and VEGAS in the Cuba multidimensional integration library [21]. Only VEGAS [23,24] was found to give a high-quality estimate of the integral as well as a reliable associated uncertainty, and therefore it will be the standard benchmark used throughout this work. Suave yielded inaccurate integral estimates, while Divonne was found to underestimate its actual error (see discussion below).

In the top panel of Fig. 1 we compare the VEGAS (green) and normalizing flow (red) total relative uncertainty

$$\frac{\sigma_t}{\Omega^{(2)}} = \frac{1/\sqrt{\sum_i \frac{1}{\sigma_i^2}}}{\sum_i \frac{\Omega_i^{(2)}}{\sigma_i^2} / \sum_i \frac{1}{\sigma_i^2}} \tag{9}$$

over the first 3000 iterations, where in Eq. (9) i is the training iteration, σ_i is the batch standard error, and $\Omega_i^{(2)}$ is the batch mean. We observe that VEGAS outperforms the normalizing flow early in the training, but the greater expressive capacity of the normalizing flow leads to a smaller total relative uncertainty past 500 iterations. Moreover, in stark contrast to VEGAS, the normalizing

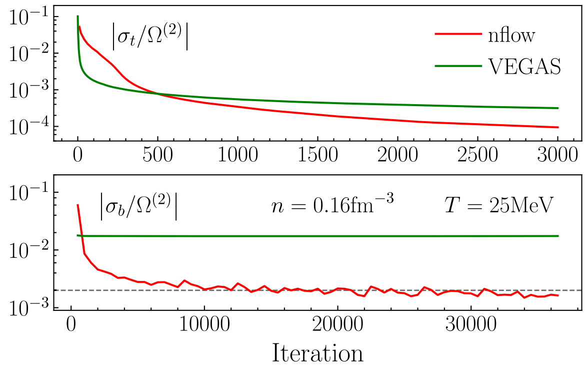

FIG. 1. (top) Total relative uncertainty $|\sigma_t/\Omega^{(2)}|$ as a function of iteration at the beginning of training for the evaluation of $\Omega^{(2)}(n_0, T = 25\ \mathrm{MeV})$ in Eq. (5) using the normalizing flow (red) and VEGAS (green) Monte Carlo integration algorithms. (bottom) Same as top panel but for the batch relative uncertainty $|\sigma_b/\Omega^{(2)}|$ as a function of iteration over the full training time. The number of batch samples per iteration is 5000. The gray dashed line is added to help guide the eye.

flow continues to learn over many training iterations, as shown in the bottom panel of Fig. 1, where we plot the batch relative uncertainty $|\sigma_b/\Omega^{(2)}|$, which reflects the performance of each model at a particular training iteration with 5000 samples. We observe that the VEGAS batch uncertainty saturates after less than one hundred iterations, while the normalizing flow batch uncertainty continues to decrease throughout training. We stopped training the normalizing flow when the total relative uncertainty reached 10^{-5} (after 36 000 iterations) with associated batch relative uncertainty of $\sim 1.5 \times 10^{-3}$, which is an order of magnitude improvement over VEGAS.

When precise integral estimates are required (as is the case for computing numerical derivatives of the free energy), it is clear that normalizing flows are able to outperform VEGAS, with the caveat that a high sample complexity is required to reach this low precision. We now demonstrate that this initial high sample complexity is a one-time cost, and that normalizing flow models transfer exceptionally well when either the density, temperature, or even the nuclear potential in Eq. (5) is varied. In the left and right panels of Fig. 2 we show the temperature and density dependence, respectively, of the total relative uncertainty $|\sigma_t/\Omega^{(2)}|$ in the evaluation of Eq. (5). Both VEGAS (green) and the normalizing flow (red) were first trained at the phase space point indicated by the star and then transferred sequentially to different densities and temperatures using just 100 additional training steps (500 000 samples). At the starting point ($n = n_0, T = 25$ MeV), the normalizing flow begins with more than an order of magnitude better uncertainty estimate compared to VEGAS. We can see that this improvement in the precision (relative to VEGAS) persists as both the density and temperature are varied. We note that the increase in total relative uncertainty as temperature decreases is not unexpected, since the sharpening of the Fermi distribution functions becomes difficult to model within any adaptive Monte Carlo method, as evidenced by the similar behavior demonstrated by VEGAS.

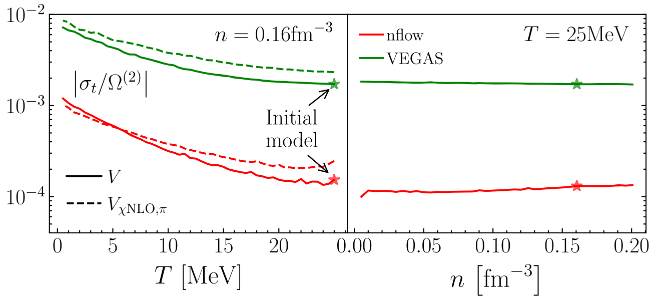

FIG. 2. Total relative uncertainty for the evaluation of $\Omega^{(2)}(n, T)$ in Eq. (5) using the VEGAS (green solid line) and normalizing flow (red solid line) Monte Carlo integration algorithms. Both models were initially trained at $n = n_0$ and $T = 25$ MeV (denoted by the star) for 36 000 iterations and then transferred to nearby phase space points using only 100 additional training iterations (500 000 samples). Also shown are the total relative uncertainties (dashed lines) for both models trained for 100 iterations when the potential in Eq. (4) is replaced by the pion-exchange terms up to next-to-leading-order in the chiral expansion.

We also show in Fig. 2 the ability for each model to adapt to nontrivial changes in the choice of nuclear potential. In particular, the dashed lines in the left panel of Fig. 2 denote the total relative uncertainty in the evaluation of Eq. (5) after replacing the simple potential in Eq. (4) with the sum of the leading-order (LO) and next-to-leading-order (NLO) pion-exchange contributions in realistic chiral effective field theory nuclear forces [40,48]. In obtaining the integral estimates for $V_{\chi\mathrm{NLO},\pi}$, we included one extra training run (100 iterations) at the phase space point ($n = n_0, T = 25$ MeV) to reorient the normalizing flow model before using the standard 100 iterations to train and evaluate at all phase space points.

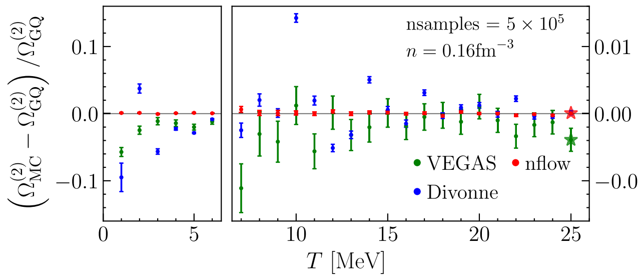

FIG. 3. Relative error for the evaluation of $\Omega^{(2)}_{\mathrm{MC}}(n, T)$ in Eq. (5) using the VEGAS (green), Divonne (blue), and normalizing flow (red) Monte Carlo integration algorithms compared to the exact evaluation $\Omega^{(2)}_{\mathrm{GQ}}(n, T)$ using Gaussian quadrature. The VEGAS and normalizing flow models were initially trained at $n = n_0$ and $T = 25$ MeV and then transferred to lower temperatures using 100 additional training iterations (500 000 samples).

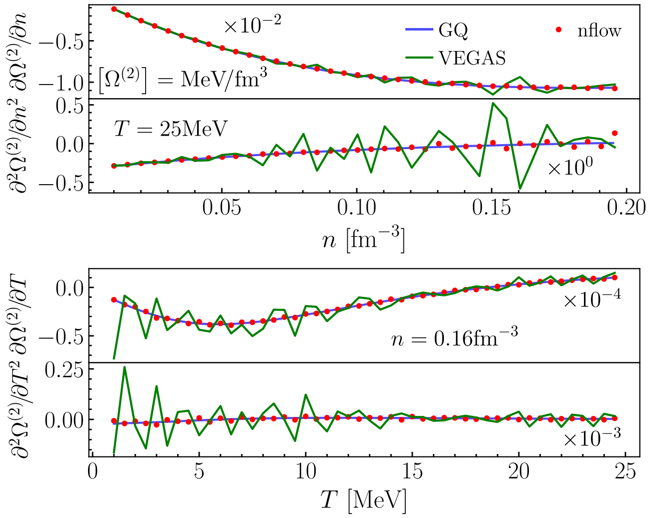

FIG. 4. 1st and 2nd-order derivatives of $\Omega^{(2)}$ with respect to the density n and temperature T from VEGAS (green line), normalizing flows (red dots), and exact Gaussian quadrature (blue line). The numerical derivatives are calculated by the central finite difference method of order 2 generated from the data shown in Fig. 2.

We observe that the normalizing flow is able to efficiently transfer even when a highly nontrivial change to the nuclear potential is introduced.

In Fig. 3 we show the true Monte Carlo integration errors for Divonne, VEGAS, and normalizing flows by comparing to exact results obtained from Gaussian quadrature (GQ). In general, we observe that the integral estimates for $\Omega^{(2)}_{\mathrm{MC}}$ from the normalizing flow and VEGAS are within 1 or 2 standard deviations of the exact result $\Omega^{(2)}_{\mathrm{GQ}}$, while Divonne significantly underestimates its actual error.

We now estimate how the uncertainties in the Monte Carlo estimates for the grand canonical potential from the VEGAS and normalizing flow models propagate to the calculation of numerical derivatives. In the top and lower panels of Fig. 4 we show the 1st and 2nd-order derivatives of $\Omega^{(2)}$ with respect to the density and temperature, respectively. The central finite difference method of order 2 is applied to calculate the numerical derivatives from the VEGAS (green) and normalizing flow (red) datasets generated in Fig. 2 as well as the exact results obtained through Gaussian quadrature (blue). We see that the improved numerical precision from the normalizing flow leads to significantly better estimates of free energy derivatives. In particular, we can see that in the low-temperature region the derivatives from VEGAS fluctuate strongly about the true value, while the results from the normalizing flow are stable and match the exact values well, even for the 2nd-order derivatives.

Outlook.—In the present work, we have performed proof-of-principle calculations demonstrating the potential of normalizing flow-based importance sampling in the context of nuclear many-body perturbation theory. In particular, we have shown that normalizing flows are able to learn models of the target integrand which allow for precise integral estimates and which can be transferred to related integrals with minimal additional computational cost. Ultimately, this leads to speedup factors on the order of 100 compared to VEGAS when precise integral evaluations must be repeated across a multi-dimensional phase space, such as in calculations of the free energy and its numerical derivatives in astrophysical equation of state tables. Numerous extensions and further applications are envisioned. One important application is the nucleon single-particle energy, a quantity that varies over the four-dimensional parameter space $\{n, T, Y_p, q\}$, where q is the nucleon momentum. First- and second-order derivatives of this quantity are needed when computing, e.g., the nucleon effective mass. Another application is to nuclear matter response functions, which vary over the five-dimensional parameter space $\{n, T, Y_p, q, \omega\}$, where q and ω represent the momentum and energy transfer to the medium. In all of these cases, normalizing flows may allow for the inclusion of perturbation theory contributions that at present are too computationally demanding to map over the full phase space needed in astrophysical applications.

J. B. would like to thank Prafulla Choubey for helpful collaboration in the early stages of this project. Work supported by the National Science Foundation under Grant No. PHY1652199 and by the U.S. Department of Energy National Nuclear Security Administration under Grant No. DE-NA0003841. Portions of this research were conducted with the advanced computing resources provided by Texas A&M High Performance Research Computing.

[*] jack.brady@tamu.edu
[†] pswen2019@tamu.edu
[‡] holt@physics.tamu.edu

[1] J. M. Lattimer and M. Prakash, Phys. Rep. **333–334**, 121 (2000).

[2] J. M. Lattimer and M. Prakash, Phys. Rep. **621**, 127 (2016).

[3] J. Lattimer and D. Swesty, Nucl. Phys. **A535**, 331 (1991).

[4] H. Shen, H. Toki, K. Oyamatsu, and K. Sumiyoshi, Nucl. Phys. **A637**, 435 (1998).

[5] G. Shen, C. J. Horowitz, and S. Teige, Phys. Rev. C **83**, 035802 (2011).

[6] A. W. Steiner, M. Hempel, and T. Fischer, Astrophys. J. **774**, 17 (2013).

[7] C. Drischler, J. A. Melendez, R. J. Furnstahl, and D. R. Phillips, Phys. Rev. C **102**, 054315 (2020).

[8] H. Yasin, S. Schäfer, A. Arcones, and A. Schwenk, Phys. Rev. Lett. **124**, 092701 (2020).

[9] A. S. Schneider, L. F. Roberts, C. D. Ott, and E. O'Connor, Phys. Rev. C **100**, 055802 (2019).

[10] P. Donati, P. M. Pizzochero, P. F. Bortignon, and R. A. Broglia, Phys. Rev. Lett. **72**, 2835 (1994).

[11] C. Wellenhofer, J. W. Holt, and N. Kaiser, Phys. Rev. C **92**, 015801 (2015).

[12] A. Carbone, A. Polls, and A. Rios, Phys. Rev. C **98**, 025804 (2018).

[13] C. Drischler, J. W. Holt, and C. Wellenhofer, arXiv:2101.01709 [Annu. Rev. Nucl. Part. Sci. (to be published)].

[14] V. Bernard, E. Epelbaum, H. Krebs, and U.-G. Meissner, Phys. Rev. C **77**, 064004 (2008).

[15] V. Bernard, E. Epelbaum, H. Krebs, and U.-G. Meissner, Phys. Rev. C **84**, 054001 (2011).

[16] J. W. Holt and N. Kaiser, Phys. Rev. C **95**, 034326 (2017).

[17] C. Drischler, K. Hebeler, and A. Schwenk, Phys. Rev. Lett. **122**, 042501 (2019).

[18] S. Typel, M. Oertel, and T. Klaehn, Phys. Part. Nucl. **46**, 633 (2015).

[19] H. Togashi, K. Nakazato, Y. Takehara, S. Yamamuro, H. Suzuki, and M. Takano, Nucl. Phys. **A961**, 78 (2017).

[20] J.-J. Lu, Z.-H. Li, G. F. Burgio, A. Figura, and H.-J. Schulze, Phys. Rev. C **100**, 054335 (2019).

[21] T. Hahn, Comput. Phys. Commun. **168**, 78 (2005).

[22] A. B. Owen, *Monte Carlo Theory, Methods and Examples* (2013).

[23] G. Lepage, J. Comput. Phys. **27**, 192 (1978).

[24] G. Lepage, J. Comput. Phys. **439**, 110386 (2021).

[25] E. G. Tabak and E. Vanden-Eijnden, Commun. Math. Sci. **8**, 217 (2010).

[26] E. Tabak and C. Turner, Commun. Pure Appl. Math. **66**, 145 (2013).

[27] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, J. Mach. Learn. Res. **22**, 1 (2021), http://jmlr.org/papers/v22/19-1028.html.

[28] D. J. Rezende and S. Mohamed, arXiv:1505.05770.

[29] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, arXiv:1606.04934.

[30] A. Golinski, F. Wood, and T. Rainforth, arXiv:1907.08082.

[31] T. Müller, B. Mcwilliams, F. Rousselle, M. Gross, and J. Novák, ACM Trans. Graph. **38**, 145 (2019).

[32] C. Gao, J. Isaacson, and C. Krause, Mach. Learn.: Sci. Technol. **1**, 045023 (2020).

[33] E. Bothmann, T. Janßen, M. Knobbe, T. Schmale, and S. Schumann, SciPost Phys. **8**, 69 (2020).

[34] P. Wirnsberger, A. Ballard, G. Papamakarios, S. Abercrombie, S. Racanière, A. Pritzel, D. J. Rezende, and C. Blundell, J. Chem. Phys. **153**, 144112 (2020).

[35] G. Kanwar, M. S. Albergo, D. Boyda, K. Cranmer, D. C. Hackett, S. Racanière, D. J. Rezende, and P. E. Shanahan, Phys. Rev. Lett. **125**, 121601 (2020).

[36] J. Bendavid, arXiv:1707.00028.

[37] M. D. Klimek and M. Perelstein, SciPost Phys. **9**, 53 (2020).

[38] R. Rossi, Phys. Rev. Lett. **119**, 045701 (2017).

[39] M. Maček, P. T. Dumitrescu, C. Bertrand, B. Triggs, O. Parcollet, and X. Waintal, Phys. Rev. Lett. **125**, 047702 (2020).

[40] R. Machleidt and D. R. Entem, Phys. Rep. **503**, 1 (2011).

[41] L. Dinh, D. Krueger, and Y. Bengio, arXiv:1410.8516.

[42] L. Dinh, J. Sohl-Dickstein, and S. Bengio, arXiv:1605.08803.

[43] J. A. Gregory and R. Delbourgo, IMA J. Numer. Anal. **2**, 123 (1982).

[44] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, arXiv:1906.04032.

[45] K. He, X. Zhang, S. Ren, and J. Sun, *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, New York, 2016), p. 770.

[46] D. P. Kingma and J. Ba, arXiv:1412.6980.

[47] A. Paszke *et al.*, in *Advances in Neural Information Processing Systems 32* (Curran Associates, Inc., 2019) p. 8024.

[48] E. Epelbaum, H.-W. Hammer, and U.-G. Meißner, Rev. Mod. Phys. **81**, 1773 (2009).