SOLVING ODE WITH UNIVERSAL FLOWS: APPROXIMATION THEORY FOR FLOW-BASED MODELS

Chin-Wei Huang[1], **Laurent Dinh**[2], **Aaron Courville**[1,3]
[1]Mila (University of Montreal), [2]Google Brain, [3]CIFAR Fellow

ABSTRACT

Normalizing flows are powerful invertible probabilistic models that can be used to translate two probability distributions, in a way that allows us to efficiently track the change of probability density. However, to trade for computational efficiency in sampling and in evaluating the log-density, special parameterization designs have been proposed at the cost of representational expressiveness. In this work, we propose to use ODEs as a framework to establish universal approximation theory for certain families of flow-based models.

1 INTRODUCTION

Deep invertible models have recently gained increasing interest among machine learning researchers as they constitute a powerful probabilistic toolkit. They allow for the tracking of changes in probability density and have been widely applied in many tasks, including

(i) generative models (Dinh et al., 2017; Kingma & Dhariwal, 2018; Chen et al., 2019),

(ii) variational inference (Rezende & Mohamed, 2015; Kingma et al., 2016; Berg et al., 2018),

(iii) density estimation (Papamakarios et al., 2017; Huang et al., 2018),

(iv) reinforcement learning (Mazoure et al., 2019; Ward et al., 2019), etc.

The main challenges in designing an invertible model for the above use cases are to ensure (1) the mapping f is invertible, (2) the log-determinant of the Jacobian of f is cheap to compute, and (3) f is expressive. For use case (i), ideally we would also like to (4) invert f efficiently. In general, it is hard to design a family of functions that satisfy all of the above. Most work within this line of research is dedicated to improving the expressivity of the bijective mapping while maintaining the computational tractability of the log-determinant of Jacobian (Dinh et al., 2014; Kingma et al., 2016; Dinh et al., 2017; Berg et al., 2018; Huang et al., 2018; Chen et al., 2019).

Huang et al. (2018) propose to approximate a universal triangular map proposed by Hyvarinen & Pajunen (1998), which is also known as the Knothe-Rosenblatt transformation in the optimal transport literature (Villani, 2008). Huang et al. (2018) show that if one can approximate such a triangular map pointwise (using a family of monotonic neural networks), then one can universally transform one random variable into another. This is extended by Jaini et al. (2019) who propose to use sum-of-square polynomial functions as approximations.

In this work, we consider a different approach that relies on building a transport map which solves an ordinary differential equation (ODE). We show that (1) if the solution of an ODE x_T with $x_0 \sim q(x)$ converges in distribution to $x_\infty \sim p_\infty(x)$ as $T \to \infty$, and if (2) if one can approximate the function x_T to any arbitrary precision (pointwise), then we can approximate the distribution p_∞ by transforming x_0 using the approximating function.

2 NORMALIZING FLOWS

Assume $y \sim \mathcal{N}(0, I)$. Assume the data is generated via a bijective mapping $x = f_\theta(y)$. Then the probability density function of $f_\theta(y)$ evaluated at x can be written as

$$p_\theta(x) = \mathcal{N}(f_\theta^{-1}(x); 0, I) \left| \det \frac{\partial f_\theta^{-1}(x)}{\partial x} \right| \tag{1}$$

Equivalently, one can parameterize the inverse transformation $x \mapsto g_\theta(x)$ with invertible mapping g_θ, and define the generative transformation as $f_\theta = g_\theta^{-1}$.

Much of the design effort has been dedicated to ensuring (1) the invertibility of the transformation g, and (2) efficiency in computing the log-determinant of the Jacobian in Equation 1.

Block-wise Affine Coupling For example, Dinh et al. (2017) propose the affine coupling:

$$g_\theta(x_a, x_b) = \texttt{concat}(x_a,\ s_\theta(x_a) \odot x_b + m_\theta(x_a))$$

where s_θ and m_θ are parameterized by neural networks and x_a and x_b are two partitioning of the data vector, and compose multiple layers of transformations intertwined with permutation of elements of x. Since an affine transformation is applied to modify one block of the data conditioned on the other block, we refer to this type of transformation as the *block-wise affine coupling* (BWAC). While BWAC is computationally efficient for evaluation and for sampling (inversion), the common criticism is that it requires a longer chain of transformations to reach the same performance level as other flow methods (Rezende & Mohamed, 2015; Kingma et al., 2016), and that the affine transformation is not capable of redistributing the probability mass non-uniformly (Huang et al., 2018), which is essential for multimodal distribution.

The BWAC can be extended so that all of the data is transformed in one pass (Kingma et al., 2016):

$$g_\theta(x)_i = s_{\theta,i}(x_{<i}) \cdot x_i + m_{\theta,i}(x_{<i})$$

for i iterating over the indices of the features. Later on, Huang et al. (2018) generalize the affine coupling to invertible neural transformation:

$$g_\theta(x)_i = \text{NN}(x_i; \pi_i(x_{<i}))$$

where NN is a neural network with positive weights and strictly monotonic activation functions; π_i is a hyper-network outputing the parameters of NN which takes all the preceding $x_{<i}$ as input. This non-affine autoregressive transformation is shown to be universal transport map between any probability distributions, by approximating the Knothe-Rosenblatt rearrangement between two random variables.

Residual Flows Another family of flows which do not rely on the partial dependency to have tractable Jacobian determinant computation are of a residual form

$$g_\theta(x) = x + h_\theta(x)$$

Behrmann et al. (2018) show that if h_θ is contractive (i.e. h_θ is C-Lipschitz for some $C < 1$), then g_θ is invertible. Residual flows can be seen as a discretized neural ODE (Chen et al., 2018; Grathwohl et al., 2019):

$$g_\theta(x; T) = \int_0^T h(g_\theta(x; t), t, \theta)\, dt$$

3 APPROXIMATING SOLUTIONS OF ODE AS UNIVERSAL TRANSPORT MAP

In this section, we demonstrate how to use ODEs as a tool to show certain families of flows are universal density approximators. We do so in two steps: (1) we identify a universal transport map in the form of a solution to an ODE

$$\dot{x}_t = k(x_t, t)$$

An ODE is called a **transport map** if $x_0 \sim p_0$ (e.g. data distribution) and x_T following the dynamics converges in distribution to some limiting random variable of interest (such as standard Gaussian). (2) we approximate the ODE using invertible neural networks.

Assume x_n solves some ODE at some time step indexed by n. The following lemma shows that if x_n converges in distribution to x_∞ as n approaches infinity, and if x_n and y_n are asymptotically indistinguishable, then y_n also converges in distribution to x_∞.

Lemma 1. *Let x_∞, $(x_n : n \geq 0)$ and $(y_n : n \geq 0)$ be random variables. If $x_n \to x_\infty$ in distribution and if $||x_n - y_n|| \to 0$ almost surely as $n \to \infty$, then $y_n \to x_\infty$ in distribution.*

Below, we give two examples of ODEs that converge in distribution to some asymptotic limit that we can approximate using universal flows.

Example 1. *(Deterministic Langevin diffusions) Consider the following overdamped Langevin stochastic differential equation (aka Brownian dynamics)*

$$\dot{x}_t = \nabla \log p_\infty(x_t) + \sqrt{2}\dot{w}_t$$

where w_t is the standard Brownian motion, which has p_∞ as its stationary distribution. It is a well known result (Roberts et al., 1996) that under some mild smoothness condition on $\log p_\infty$, the diffusion above satisfies $x_t \overset{TV}{\to} x_\infty$. Notably, replacing the Brownian motion term with $-\nabla \log p_t(x)$ (where p_t is the density of x_t) corresponds to the functional gradient of the KL divergence

$$\nabla \log p_\infty - \nabla \log p_t = -\frac{\delta}{\delta r_t(\epsilon)} \int p_\epsilon(\epsilon)[\log p_t(r_t(\epsilon)) - \log p_\infty(r_t(\epsilon))]d\epsilon$$

for some reparameterization r_t which satisfies $P_\epsilon \circ r_t^{-1} = P_t$, where P_t is the law of x_t and P_ϵ is the law of ϵ that corresponds to the density p_ϵ; this deterministic modification does not change the Fokker Planck representation of the original Langevin SDE (Wang & Li, 2019; Hoffman & Ma, 2019), which means its solution will be statistically indistinguishable from the original one if the same initial law P_0 is chosen.

Two results follow immediately from this example. First, one can approximate the gradient flow $k_t := \nabla \log p_\infty - \nabla \log p_t$ using a neural network $h_t := h(\cdot, t)$. The corresponding solution of the neural ODE would converge to the solution of k_t if the approximation error can be somehow controlled; informally,

$$\left| \int h_t - \int k_t \right| \leq \int |h_t - k_t| \to 0 \qquad \text{if} \qquad h_t \to k_t$$

Second, one can numerically integrate the ODE and approximate the numerical integration using discrete flows. Using the Euler method, we have

$$x_{t+\epsilon} \approx x_t + \epsilon k_t(x_t)$$

Provided that k_t is Lipschitz, then for sufficiently small ϵ, ϵk_t is contractive, which can then be approximated by certain family of Lipschitz neural networks (Anil et al., 2019).

Additionally, it can be shown (with some care) that, combining standard methods of numerical integration (such as Euler's method and the midpoint method) with function approximation using neural networks leads to an "approximate integration error" d_n that satisfies the premise of the following lemma.

Lemma 2. *If for any $N > 0$, $\{d_n : 0 \leq n \leq N\}$ is a sequence of real numbers satisfying*

$$d_n \leq \frac{c}{N^2} + \frac{c}{N^2} \sum_{t=1}^{n-1} \sum_{s=1}^{t} d_s$$

for some constant c, then

$$\max_{0 \leq n \leq N} d_n \;\to\; 0 \quad \textit{as} \quad N \to \infty$$

Lemma 1 and lemma 2 imply that infinitely deep residual flows are universal density approximators.

Example 2. *(Hamiltonian ODE) Define the following scaling coefficients $\alpha_t = \log \frac{2}{t}$ and $\beta_t = \gamma_t = \log t^2$. Let $p_\infty(x)$ be the standard normal density, and $q(x)$ be the data distribution. Let $q_0 = q$ and $\Phi : \mathcal{X} \to \mathbb{R}$ be some convex function. Define the Hamiltonian ODE:*

$$\dot{x}_t = e^{\alpha_t - \gamma_t} e_t, \qquad x_0 \sim q_0$$

$$\dot{e}_t = -e^{\alpha_t + \beta_t + \gamma_t} \nabla \log \frac{q_t(x_t)}{p_\infty(x_t)}, \qquad e_0 = \nabla \Phi(x_0)$$

where $\dot{x}_t$ and $\dot{e}_t$ are the time derivatives of x and e at time t, and q_t is the marginal density of x_t. It follows by Taghvaei & Mehta (2019); Wang & Li (2019) that for some Φ, (x_t, e_t) converges in distribution to (x_∞, e_∞) where $x_\infty \sim p_\infty$ and $e_\infty \sim \delta_0$ (point mass at 0).

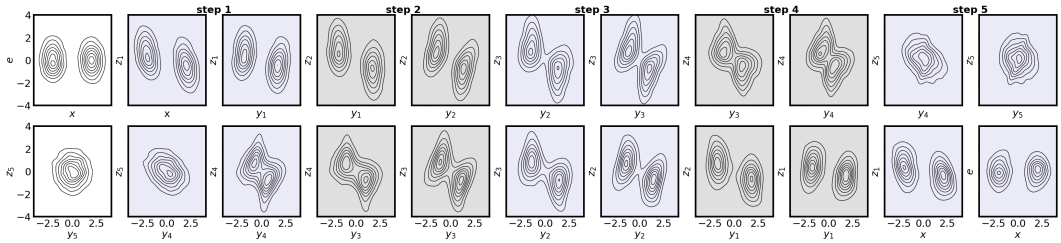

Figure 1: 5-step ANF on 1D MoG. In the inference path (top row), we start with an encoding transform that maps e to z_1 conditioned on x, followed by a decoding transform that maps x into y_1 conditioned on z_1. We reuse the same encoder and decoder to refine the joint variable repeatedly to obtain y_5 and z_5. In the generative path (bottom row), we reverse the process, starting with the inverse transform of the decoding, followed by the inverse transform of the encoding, etc.

ODE of this form has a desired property, since it is "conditionally affine", and thus invertible by construction. This means the Hamiltonian ODE can be approximated by BWAC.

Specifically, let us construct a sequence of "encoding" and "decoding" functions m_n^{enc} and m_n^{dec} parameterized by neural networks, and define the following (additive) invertible mappings

$$e_1^\pi = e_0^\pi + m_1^{\text{enc}}(x_0^\pi)$$
$$x_{n+1}^\pi = x_n^\pi + 2\epsilon \cdot m_{n+1}^{\text{dec}}(e_{n+1}^\pi) \qquad \forall\, n \geq 0 \tag{2}$$
$$e_{n+1}^\pi = e_n^\pi + 2\epsilon \cdot m_{n+1}^{\text{enc}}(x_n^\pi) \qquad \forall\, n \geq 1 \tag{3}$$

with $e_0^\pi = 0$ and $x_0^\pi \sim q_0$. The step size parameter ϵ will be chosen to depend on the depth coefficient N, i.e. the number of steps of the joint transformation.

Assume our target distribution lies within a family of distributions $\mathcal{Q}$ satisfying Assumption 1 in the Appendix A (some smoothness condition on the time derivatives and Φ). We can then set the encoding and decoding functions to be arbitrarily close to the time derivatives by the universal approximation of neural networks (Cybenko, 1989), and by taking the depth N to be arbitrarily large, we can approximate the transport map induced by the Hamiltonian ODE arbitrarily well, which gives rise to the following universal approximation theorem (the proof is relegated to the Appendix A):

Theorem 1. *For any $q \in \mathcal{Q}$, we can find a sequence (x_N^π, e_N^π) of ANFs of the additive form (2,3), such that if $x_0^\pi, e_0^\pi \sim q(x)\delta_0(e)$ and $x_\infty, e_\infty \sim p_\infty(x)\delta_0(e)$, then $(x_N^\pi, e_N^\pi) \to (x_\infty, e_\infty)$ in distribution.*

The theorem suggests BWAC can be made expressive by augmented the data x with an auxiliary variable. However, training a model with a Dirac prior δ_0 is problematic because the loss is not smooth. To remedy this problem, we consider using a non-degenerate augmented data distribution $q(e)$ (taken to be the standard normal) and maximizing the joint likelihood of data (x, e) sampled from $q(x)q(e)$ under a generative flow with the joint prior being standard normal as well. We call this the **augmented normalizing flow** (ANF). We use the BWAC as suggested by Theorem 1. Figure 1 demonstrates that ANFs are capable of transforming the marginal of a 1-D mixture of Gaussian non-uniformly into a standard normal prior, which is not possible with regular BWAC (since for this 1-D problem BWAC amounts to mere shifting and scaling, which only modifies the first two moments of the data distribution).

4 Conclusion

In this work, we propose to use ODEs that are universal transport map to establish universality of flow-based methods (in the space of probability distributions). The takeaway is that composing certain families of flows can be shown to be universal using this technique and that some other techniques to parameterize flows can also be motivated by our new theory, such as augmentation.

References

Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pp. 291–301, 2019.

Jens Behrmann, Will Grathwohl, Ricky T. Q. Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, 2018.

Rianne van den Berg, Leonard Hasenclever, Jakub M Tomczak, and Max Welling. Sylvester normalizing flows for variational inference. In *International Conference on Machine Learning*, 2018.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Neural Information Processing Systems*, pp. 6571–6583, 2018.

Ricky T. Q. Chen, Jens Behrmann, David Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. In *Neural Information Processing Systems*, 2019.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

James F Epperson. *An introduction to numerical methods and analysis*. John Wiley & Sons, 2013.

Will Grathwohl, Ricky T. Q. Chen, Jesse Betterncourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2019.

Matthew D Hoffman and Yian Ma. Langevin dynamics as nonparametric variational inference. 2019.

Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, 2018.

A Hyvarinen and Petteri Pajunen. On existence and uniqueness of solutions in nonlinear independent component analysis. In *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36227)*, volume 2, pp. 1350–1355. IEEE, 1998.

Priyank Jaini, Kira A Selby, and Yaoliang Yu. Sum-of-squares polynomial flow. In *International Conference on Machine Learning*, pp. 3009–3018, 2019.

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Neural Information Processing Systems*, pp. 10215–10224, 2018.

Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Neural Information Processing Systems*, pp. 4743–4751, 2016.

Bogdan Mazoure, Thang Doan, Audrey Durand, R Devon Hjelm, and Joelle Pineau. Leveraging exploration in off-policy algorithms via normalizing flows. In *Conference on Robot Learning*, 2019.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Neural Information Processing Systems*, pp. 2338–2347, 2017.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, 2015.

Gareth O Roberts, Richard L Tweedie, et al. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996.

Amirhossein Taghvaei and Prashant Mehta. Accelerated flow for probability distributions. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6076–6085, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Yifei Wang and Wuchen Li. Accelerated information gradient flow, 2019.

Patrick Nadeem Ward, Ariella Smofsky, and Avishek Joey Bose. Improving exploration in soft-actor-critic with normalizing flows policies. *arXiv preprint arXiv:1906.02771*, 2019.

Herbert S Wilf. *generatingfunctionology*. AK Peters/CRC Press, 2005.

A PROOFS

Define the scaling coefficients $\alpha_t = \log \frac{2}{t}$ and $\beta_t = \gamma_t = \log t^2$. Let $p(x)$ be the standard normal density, and $q(x)$ be the data distribution. Let $q_0 = q$ and $\Phi : \mathcal{X} \to \mathbb{R}$ be some continuous function. Define the following Hamiltonian ordinary differential equation (ODE):

$$\dot{x}_t = e^{\alpha_t - \gamma_t} e_t, \qquad x_0 \sim q_0 \tag{4}$$

$$\dot{e}_t = -e^{\alpha_t + \beta_t + \gamma_t} \log \frac{q_t(x_t)}{p(x_t)}, \qquad e_0 = \nabla \Phi(x_0) \tag{5}$$

where $\dot{x}_t$ and $\dot{e}_t$ are the time derivatives of x and e at time t, and q_t is the marginal density of x_t.

Proposition 1. *For some convex Φ, the trajectories of x_t and e_t following (4,5) converge in distribution to x_∞ and e_∞, respectively, where $x_\infty \sim p(x)$ and $e_\infty \sim \delta_0$ (i.e. a point mass at 0).*

Proof. By Theorem 1 of Taghvaei & Mehta (2019) and Appendix C.4 of Wang & Li (2019) (for an extension to high dimensional cases), since α_t, β_t and γ_t satisfy the scaling condition in Taghvaei & Mehta (2019) and $\log p$ is convex, x_t converges in KL divergence to x_∞ and e_t converges to 0 almost surely (which implies convergence in distribution). Pinsker's inequality implies $x_t \to x_\infty$ in total variation, d_{TV}, which has a dual representation:

$$d_{\mathrm{TV}}(x_t, x_\infty) = \sup_{f : \mathcal{X} \to [-1,1]} \mathbb{E}[f(x_t)] - \mathbb{E}[f(x_\infty)]$$

This implies for any bounded, continuous f,

$$|\mathbb{E}[f(x_t)] - \mathbb{E}[f(x_\infty)]| \le d_{\mathrm{TV}}(x_t, x_\infty) \cdot ||f||_\infty$$

which converges to 0 as $t \to \infty$. By Portmanteau's Lemma, $x_t \to x_\infty$ in distribution. □

We first construct a sequence of encoding functions m_n^{enc} and decoding functions m_n^{dec} parameterized by neural networks, and define the following (volume preserving) invertible mappings

$$\begin{aligned} e_1^\pi &= e_0^\pi + m_1^{\mathrm{enc}}(x_0^\pi) \\ x_{n+1}^\pi &= x_n^\pi + 2\epsilon \cdot m_{n+1}^{\mathrm{dec}}(e_{n+1}^\pi) && \forall\, n \ge 0 \\ e_{n+1}^\pi &= e_n^\pi + 2\epsilon \cdot m_{n+1}^{\mathrm{enc}}(x_n^\pi) && \forall\, n \ge 1 \end{aligned}$$

with $e_0^\pi = 0$ and $x_0^\pi \sim q_0$. The step size parameter ϵ will be chosen to depend on the depth coefficient N, i.e. the number of layers of the joint transformation.

Below we prove ANF of the above form can universally transform $q(x)\delta_0(e)$ into $p(x)\delta_0(e)$. We make the following assumption on the family of q:

Assumption 1. *We assume the gradient of the convex function in Proposition (1) $\nabla\Phi$ is continuous, and that $f(e,t) := e^{\alpha_t - \gamma_t} e$ and $g(x,t) := -e^{\alpha_t + \beta_t + \gamma_t} \log \frac{q_t(x)}{p(x)}$ have a bounded second time derivative (on the trajectories x_t and e_t which are also functions of time), and are uniformly Lipschitz; that is,*

$$\max \left\{ ||f''||, \quad ||g''||, \quad \sup_{e \neq e', t > 0} \frac{||f(e,t) - f(e',t)||}{||e - e'||}, \sup_{x \neq x', t > 0} \frac{||g(x,t) - g(x',t)||}{||x - x'||} \right\} \le K$$

for some $K \ge 0$, where we define the single-argument vector functions $f(t) = f(e_t, t)$ and $g(t) = g(x_t, t)$ as the time derivatives of the trajectories (x_t, e_t).

We denote by $\mathcal{Q}$ the family of probability measures that satisfies this assumption.

Before we move on to approximation, we start with a lemma (restated) for bounding approximation error by solving recursion using the technique of generating functions.

Lemma 2. *If for any $N > 0$, $\{d_n : 0 \le n \le N\}$ is a sequence of real numbers satisfying*

$$d_n \le \frac{c}{N^2} + \frac{c}{N^2} \sum_{t=1}^{n-1} \sum_{s=1}^{t} d_s$$

for some constant c, then

$$\max_{0 \le n \le N} d_n \;\to\; 0 \quad \textit{as} \quad N \to \infty$$

Proof. We would like to bound the error d_n explicitly. To do so, we first note that the sequence $\{d_n\}$ is no larger than $\{D_n\}$, which is recursively defined as

$$D_0 = 0$$
$$D_{n+1} = C + C\sum_{t=1}^{n}\sum_{s=1}^{t} D_s \tag{6}$$

for $n \geq 0$, where for simplicity we let $C = c/N^2$.

Now to express D_{n+1} explicitly, we use the method of generating function, following the recipe of Wilf (2005) (see Chapter 1 for a brief introduction). Define function f to be a power series whose coefficients are D_n's; that is, $f(x) = \sum_{n\geq 0} D_n x^n$. Multiply both sides of (6) by x^n and summing over the indices of non-negative integers $n \geq 0$ give us

$$\frac{f(x)}{x} = \frac{C}{1-x} + \frac{Cf(x)}{(1-x)^2}$$

After rearrangement, we have

$$f(x)\left(\frac{x^2-(2+C)x+1}{x(1-x)^2}\right) = \frac{C}{1-x} \quad \Rightarrow \quad \frac{f(x)}{x} = \frac{C(1-x)}{x^2-(2+C)+1}$$

which can be decomposed into the partial fractions

$$\frac{f(x)}{x} = \frac{\frac{C}{1+a_2}}{a_1-x} + \frac{\frac{C}{1+a_1}}{a_2-x} \tag{7}$$

where a_1 and a_2 are the roots of the quadratic function $x^2-(2+C)x+1$, which satisfy $a_1+a_2 = 2+C$ and $a_1a_2 = 1$.

For sufficiently small x, we can break (7) into the geometric series

$$\frac{f(x)}{x} = \frac{C}{a_1(1+a_2)}\sum_{n\geq 0}\left(\frac{x}{a_1}\right)^n + \frac{C}{a_2(1+a_1)}\sum_{n\geq 0}\left(\frac{x}{a_2}\right)^n$$

This means for $n > 0$, since $a_1a_2 = 1$, the coefficient of $f(x)$ can be expressed as

$$D_n = \frac{C}{1+a_2}\frac{1}{a_1^n} + \frac{C}{1+a_1}\frac{(a_1a_2)^n}{a_2^n} = C\left(\frac{1}{(1+a_1)a_1^{n-1}} + \frac{a_1^n}{1+a_1}\right) \tag{8}$$

Now let a_1 be the larger root. Solving $x^2-(2+C)x+1$ yields

$$a_1 = \frac{2+C+\sqrt{C^2+4C}}{2} =: 1+r$$

where $r := \frac{C}{2} + \sqrt{\frac{C^2}{4}+C}$.

We show that the parenthesis in (8) can be controlled asymptotically (i.e. does not exceed certain constant for sufficiently large N), and that since C diminishes, D_n converges. First, since $r > 0$, $a_1 > 1$ and

$$\frac{1}{(1+a_1)a_1^{n-1}} < \frac{1}{2}$$

Second, since $(1+r)^n \leq e^{nr}$ for $n \geq 0$ and $r \geq -1$,

$$\begin{aligned} a_1^n = (1+r)^n &\leq e^{nr} \\ &\leq \exp\left(\frac{CN}{2} + \sqrt{\frac{C^2N^2}{4}+CN^2}\right) \\ &= \exp\left(\frac{c}{2N} + \sqrt{\frac{c^2}{4N^2}+c}\right) \end{aligned}$$

which converges to $\exp(\sqrt{c})$ as $N \to \infty$.

Finally, since $C \to 0$ as $N \to \infty$ and $d_n \leq D_n$, $d_n \to 0$ for all $n \leq N$ as $N \to \infty$. □

We are now ready to show the result of the pointwise approximation of the Hamiltonian ODE using ANFs with affine (more specifically, additive) coupling.

Proposition 2. *Let x_t and e_t be trajectories (mappings of $x_0 \in \mathcal{X} = \mathbb{R}^d$) following the Hamiltonian ODE (4,5) described in Proposition 1 dependent on some initial distribution $q_0 \in \mathcal{Q}$. For each $T > 0$, we can choose some number of layers N of the joint transformation and a sequence of pairs of m_n^{enc} and m_n^{dec} (dependent on T) for $1 \leq n \leq N$, such that $||x_N^\pi - x_T|| \to 0$ and $||e_N^\pi - e_T|| \to 0$ as $T \to \infty$ pointwise for $x_0 \in \mathcal{X} = \mathbb{R}^d$.*

Proof. Fix $q_0 \in \mathcal{Q}$ and $T > 0$ and some compact subset $\mathcal{X}_0 \subset \mathcal{X}$. We first consider all points x_0 in $\mathcal{X}_0$, and show that (x_n^π, e_n^π) can be used to approximate (x_T, e_T) uniformly well.

We consider a N-step joint transformation, and set $\epsilon = \frac{T}{2N} > 0$. We start with approximating e_ϵ by e_1^π. Since e_0^π is 0, by the universal approximation theorem (UAT) of neural networks (Cybenko, 1989), we can choose some m_1^{enc} such that $||e_\epsilon - e_1^\pi|| = ||e_\epsilon - m_1^{\text{enc}}|| \leq \epsilon^2$ for all $x_0 \in \mathcal{X}_0$.

We proceed with an approximate leap-frog integration of the dynamic, using the neural encoders and decoders to approximate the time derivatives. Let $\mathcal{E}_1 := e_1^\pi(\mathcal{X}_0)$ where $e_1^\pi := m_1^{\text{enc}}$, which is compact, since $\mathcal{X}_0$ is compact and e_1^π is continuous wrt $\mathcal{X}_0$. Again, by the UAT, we can choose some m_1^{dec} such that $||f(e, \epsilon) - m_1^{\text{dec}}(e)|| < \epsilon^2$ for all $e \in \mathcal{E}_1$. Likewise, we let $\mathcal{X}_1 := x_1^\pi(\mathcal{X}_0)$ where $x_1^\pi := (2\epsilon m_1^{\text{dec}} \circ e_1^\pi + Id)(\mathcal{X}_0)$ with Id being the identity map, such that $\mathcal{X}_1$ is also compact since x_1^π is continuous wrt $\mathcal{X}_0$, and choose m_2^{enc} such that $||g(x, 2\epsilon) - m_2^{\text{enc}}(x)|| < \epsilon^2$ for all $x \in \mathcal{X}_1$.

Repeating the same construction for m_n^{dec} and m_n^{enc} for $n \leq N$, we have

$$x_{n+1}^\pi = x_n^\pi + 2\epsilon m_{n+1}^{\text{dec}}(e_{n+1}^\pi) \tag{9}$$

$$e_{n+1}^\pi = e_n^\pi + 2\epsilon m_{n+1}^{\text{enc}}(x_n^\pi) \tag{10}$$

with m_n^{dec} and m_n^{enc} chosen such that

1. $||f(e, 2n\epsilon + \epsilon) - m_{n+1}^{\text{dec}}(e)|| < \epsilon^2$ for all $e \in \mathcal{E}_{n+1} := e_{n+1}^\pi(\mathcal{X}_0)$ where $e_{n+1}^\pi := 2\epsilon m_{n+1}^{\text{enc}} \circ x_n^\pi + e_n^\pi$ is a continuous map of $\mathcal{X}_0$; and
2. $||g(x, 2n\epsilon) - m_{n+1}^{\text{enc}}(x)|| < \epsilon^2$ for all $x \in \mathcal{X}_n := x_n^\pi(\mathcal{X}_0)$ where $x_n^\pi := 2\epsilon m_n^{\text{dec}} \circ e_n^\pi + x_{n-1}^\pi$ is a continuous map of $\mathcal{X}_0$.

Such choices of m_n^{enc} and m_n^{dec} are possible since by construction $\mathcal{X}_{n-1}$ and $\mathcal{E}_n$ are compact.

Equations (9,10) are approximate midpoint methods as they use functions to approximate the time derivatives evaluated at midpoints of their counterparts. The exact midpoint method has a cubic error rate of $\frac{h^3}{24} f''(\xi)$, for some ξ between the midpoint and the approximating point, where h is the interval width of each iteration; see Section 5.4 of Epperson (2013). That is,

$$x_{2n\epsilon+2\epsilon} = x_{2n\epsilon} + 2\epsilon f(e_{2n\epsilon+\epsilon}, 2n\epsilon + \epsilon) + \frac{\epsilon^3}{3} f''(\xi_{n+1}^x) \tag{11}$$

for some ξ_{n+1}^x between the two steps. Similarly,

$$e_{2n\epsilon+\epsilon} = e_{2n\epsilon-\epsilon} + 2\epsilon g(x_{2n\epsilon}, 2n\epsilon) + \frac{\epsilon^3}{3} g''(\xi_{n+1}^e) \tag{12}$$

for some ξ_{n+1}^e between the two steps.

Subtracting (9) from (11) yields

$$x_{2n\epsilon+2\epsilon} - x_{n+1}^\pi = x_{2n\epsilon} - x_n^\pi + 2\epsilon f(e_{2n\epsilon+\epsilon}, 2n\epsilon + \epsilon) - 2\epsilon m_{n+1}^{\text{dec}}(e_{n+1}^\pi) + \frac{\epsilon^3}{3} f''(\xi_{n+1}^x)$$

By triangle inequality, we have

$$\begin{aligned}
\left\|x_{2n\epsilon+2\epsilon} - x_{n+1}^\pi\right\| &\leq \|x_{2n\epsilon} - x_n^\pi\| + \left\|2\epsilon f(e_{2n\epsilon+\epsilon}, 2n\epsilon + \epsilon) - 2\epsilon m_{n+1}^{\text{dec}}(e_{n+1}^\pi)\right\| + \left\|\frac{\epsilon^3}{3} f''(\xi_{n+1}^x)\right\| \\
&\leq \underbrace{\|x_{2n\epsilon} - x_n^\pi\| + 2\epsilon \left\|f(e_{2n\epsilon+\epsilon}, 2n\epsilon + \epsilon) - m_{n+1}^{\text{dec}}(e_{n+1}^\pi)\right\|}_{\text{propagated error}} + \underbrace{\frac{\epsilon^3}{3} \left\|f''(\xi_{n+1}^x)\right\|}_{\text{truncated error}}
\end{aligned}$$

The error on the RHS consists of two parts: (1) the first two terms constitute the propagated error from the previous steps and (2) the third term is a newly introduced truncation error due to the Taylor expansion.

By triangle inequality again,

$$\begin{aligned}\left\|f(e_{2n\epsilon+\epsilon}, 2n\epsilon+\epsilon) - m^{\text{dec}}_{n+1}(e^{\pi}_{n+1})\right\| &= \left\|f(e_{2n\epsilon+\epsilon}, 2n\epsilon+\epsilon) - f(e^{\pi}_{n+1}, 2n\epsilon+\epsilon) + f(e^{\pi}_{n+1}, 2n\epsilon+\epsilon) - m^{\text{dec}}_{n+1}(e^{\pi}_{n+1})\right\| \\ &\leq \underbrace{\left\|f(e_{2n\epsilon+\epsilon}, 2n\epsilon+\epsilon) - f(e^{\pi}_{n+1}, 2n\epsilon+\epsilon)\right\|}_{\text{midpoint deviation}} + \underbrace{\left\|f(e^{\pi}_{n+1}, 2n\epsilon+\epsilon) - m^{\text{dec}}_{n+1}(e^{\pi}_{n+1})\right\|}_{\text{approximation error}}\end{aligned}$$

Again the RHS can be decomposed into two error parts: (1) a midpoint deviation resulting from performing midpoint numerical integration which would not vanish even if the neural network is replaced with the true time derivative, and (2) an approximation error due to the inaccuracy of approximating the time derivative.

Letting $d^x_n = ||x_{2n\epsilon} - x^{\pi}_n||$ and $d^e_n = ||e_{2n\epsilon-\epsilon} - e^{\pi}_n||$, and applying the properties of the Assumption 1, we have

$$d^x_{n+1} \leq d^x_n + 2\epsilon(Kd^e_{n+1} + \epsilon^2) + \frac{\epsilon^3 K}{3} = d^x_n + 2\epsilon K d^e_{n+1} + \epsilon^3\left(\frac{K}{3} + 2\right)$$

owing to the uniform error bound of the neural decoder $||f(e, 2n\epsilon+\epsilon) - m^{\text{dec}}_{n+1}(e)|| < \epsilon^2$ for all $e \in \mathcal{E}_{n+1}$ and the fact that $e^{\pi}_{n+1}(x_0) \in \mathcal{E}_{n+1}$ since $x_0 \in \mathcal{X}_0$.

The same can be done to obtain a bound on d^e_{n+1} by subtracting (10) from (12), which yields

$$d^e_{n+1} \leq d^e_n + 2\epsilon K d^x_n + \epsilon^3\left(\frac{K}{3} + 2\right)$$

To summarize, we have

$$d^e_1 \leq \epsilon^2 \tag{13}$$

$$d^x_{n+1} \leq d^x_n + 2\epsilon K' d^e_{n+1} + \epsilon^3 K' \qquad \text{for } n \geq 0 \tag{14}$$

$$d^e_{n+1} \leq d^e_n + 2\epsilon K' d^x_n + \epsilon^3 K' \qquad \text{for } n \geq 1 \tag{15}$$

where $K' = \max\{K, \frac{K}{3} + 2\}$.

Summing $d^x_1, ..., d^x_n$ and subtracting $d^x_1 + ... + d^x_{n-1}$ from both sides yield

$$d^x_n \leq 2\epsilon K' \sum_{t=1}^{n} d^e_t + n\epsilon^3 K' \tag{16}$$

Note that $d^x_0 = 0$. Similarly, summing $d^e_2, ..., d^e_n$ and subtracting $d^e_2 + ... + d^e_{n-1}$ from both sides yield

$$d^e_n \leq d^e_1 + 2\epsilon K' \sum_{t=1}^{n-1} d^x_t + (n-1)\epsilon^3 K' \tag{17}$$

To recursively express d^x_n in terms of itself (except for d^e_1), we sum over the sequence $d^e_1, ..., d^e_n$ again

$$\sum_{t=1}^{n} d^e_t \leq n d^e_1 + 2\epsilon K' \sum_{t=2}^{n}\sum_{s=1}^{t-1} d^x_s + \sum_{t=1}^{n}(t-1)\epsilon^3 K'$$

Substituting into (16) yields

$$d^x_n \leq 2\epsilon K'\left(n d^e_1 + 2\epsilon K' \sum_{t=2}^{n}\sum_{s=1}^{t-1} d^x_s + \sum_{t=1}^{n}(t-1)\epsilon^3 K'\right) + n\epsilon^3 K'$$

Since $n \leq N$, $\sum_{t=1}^{n} t \leq n^2$, $d^e_1 \leq \epsilon^2$ and $\epsilon = \frac{T}{2N}$, the above can be rearranged and further bounded by

$$d^x_n \leq \left(\frac{T^3 K'^2}{4} + \frac{T^4 K'^2}{8} + \frac{T^3 K'}{8}\right)\frac{1}{N^2} + \frac{T^2 K'^2}{N^2}\sum_{t=1}^{n-1}\sum_{s=1}^{t} d^x_s \tag{18}$$

The same can be done for (17) to analyze d_n^e.

$$\sum_{t=1}^{n-1} d_t^x \leq 2\epsilon K' \sum_{t=1}^{n-1}\sum_{s=1}^{t} d_s^e + \sum_{t=1}^{n-1} t\epsilon^3 K'$$

$$d_n^e \leq d_1^e + 2\epsilon K' \left(2\epsilon K' \sum_{t=1}^{n-1}\sum_{s=1}^{t} d_s^e + \sum_{t=1}^{n-1} t\epsilon^3 K' \right) + (n-1)\epsilon^3 K'$$

$$d_n^e \leq \left(\frac{T^2}{4} + \frac{T^4 K'^2}{8} + \frac{T^3 K'}{8} \right) \frac{1}{N^2} + \frac{T^2 K'^2}{N^2} \sum_{t=1}^{n-1}\sum_{s=1}^{t} d_s^e \tag{19}$$

By Lemma 2, we know that the elements of both sequences of error d_n^x and d_n^e converge uniformly on $1 \leq n \leq N$ to 0 as $N \to \infty$. In particular, for all $T > 0$, $\delta > 0$ and compact subset $\mathcal{X}_0$ of $\mathbb{R}^d$, there exists some large enough integer $N(T, \delta, \mathcal{X}_0) > 0$ for which a joint transformation of $N(T, \delta, \mathcal{X}_0)$ layers parameterized by some neural encoders and decoders satisfies $d^x_{N(T,\delta,\mathcal{X}_0)} \leq \delta$ and $d^e_{N(T,\delta,\mathcal{X}_0)} \leq \delta$ for all $x_0 \in \mathcal{X}_0$.

Consider some positive value $B > 0$. We let $\mathcal{X}_0 = [-B, B]^d$, $T = B$ and $\delta = \frac{1}{B}$. We can find a sequence of models with an error rate $d^x_{N(B,1/B,[-B,B]^d)} \leq 1/B$ and $d^e_{N(B,1/B,[-B,B]^d)} \leq 1/B$ converging pointwise on $\mathbb{R}^d$ to 0 as $B \to \infty$. This implies

$$d^x_{N(B,1/B,[-B,B]^d)} = \left\| x_B - x^{\pi}_{N(B,1/B,[-B,B]^d)} \right\| \to 0$$

pointwise as $B \to \infty$. The same holds for the augmented variable e. □

The lemma (restated) below shows if one can approximate the solution of an ODE ($||y_n - x_n|| \to 0$, i.e. x_n and y_n are asymptotically indistinguishable) and if the limit of the solution is a transport map ($x_n \xrightarrow{d} x_\infty$), then the approximation also forms a transport map ($y_n \xrightarrow{d} x_\infty$).

Lemma 1. *Let x_∞, $(x_n : n \geq 0)$ and $(y_n : n \geq 0)$ be random variables. If $x_n \to x_\infty$ in distribution and if $||x_n - y_n|| \to 0$ almost surely as $n \to \infty$, then $y_n \to x_\infty$ in distribution.*

Proof. Let $\Lambda : \mathbb{R}^d \to \mathbb{R}$ be an arbitrary *bounded* and *Lipschitz continuous* function. Then

$$\begin{aligned} |\mathbb{E}\left[\Lambda\left(x_\infty\right) - \Lambda\left(y_n\right)\right]| &\leq |\mathbb{E}\left[\Lambda\left(x_\infty\right) - \Lambda(x_n) + \Lambda(x_n) - \Lambda\left(y_n\right)\right]| \\ &\leq |\mathbb{E}\left[\Lambda\left(x_\infty\right) - \Lambda(x_n)\right]| + \mathbb{E}\left[|\Lambda\left(x_n\right) - \Lambda\left(y_n\right)|\right] \end{aligned}$$

First, since $x_n \to x_\infty$ in distribution and since Λ is bounded and continuous, by the Portmanteau Lemma the first term of the RHS converges to 0 as $n \to \infty$. Second, since y_n is almost surely asymptotically indistinguishable from x_n (let Ω be the almost sure set), and since the Lipschitzness of Λ implies uniform continuity, the following are true

- For all $\epsilon > 0$, there exists a $\delta > 0$ such that $||x - y|| \leq \delta$ implies $|\Lambda(x) - \Lambda(y)| \leq \epsilon$.
- For any $\delta > 0$, there exists a integer $N > 0$ such that for all $n \geq N$, $||x_n - y_n|| \leq \delta$ for all $\omega \in \Omega$.

These imply $||\Lambda(x_n) - \Lambda(y_n)|| \to 0$ on Ω. Then

$$\mathbb{E}\left[|\Lambda\left(x_n\right) - \Lambda\left(y_n\right)|\right] = \underbrace{\mathbb{E}_\Omega\left[|\Lambda\left(x_n\right) - \Lambda\left(y_n\right)|\right]}_{E_1} + \underbrace{\mathbb{E}_{\Omega^c}\left[|\Lambda\left(x_n\right) - \Lambda\left(y_n\right)|\right]}_{E_2}$$

converges to 0, since (1) boundedness of Λ and the *Bounded Convergence Theorem* imply $E_1 \to 0$ and (2) $\sup_x \Lambda(x) < \infty$ implies $E_2 \leq 2 \sup_x \Lambda(x) \mathbb{P}(\Omega^c) = 0$. Finally, since Λ is arbitrary, by the Portmanteau Lemma again, y_n converges in distribution to x_∞ as $n \to \infty$. □

We now are ready to prove Theorem 1, which we restate below. The main idea is to notice that ANFs can be made pointwise inseparable from the Hamiltonian ODE, which implies weak convergence since the Hamiltonian ODE converges in distribution.

Theorem 1. *For any $q \in \mathcal{Q}$, we can find a sequence (x_N^π, e_N^π) of ANFs of the additive form (2,3), such that if $x_0^\pi, e_0^\pi \sim q(x)\delta_0(e)$ and $x_\infty, e_\infty \sim p_\infty(x)\delta_0(e)$, then $(x_N^\pi, e_N^\pi) \to (x_\infty, e_\infty)$ in distribution.*

Proof. First, by Proposition 1, $x_B \to x_\infty$ in distribution as $B \to \infty$. Second, x_B and $x^\pi_{N(B,1/B,[-B,B]^d)}$ chosen from Proposition 2 are almost surely asymptotically indistinguishable. Thus, by Lemma 1, $x^\pi_{N(B,1/B,[-B,B]^d)}$ converges in distribution to x_∞. The same holds for the augmented variable e. Let (x_N^π) and (e_N^π) denote such sequences. By Theorem 2.7 of Van der Vaart (2000), $(x_N^\pi, e_N^\pi) \to (x_\infty, e_\infty)$ in distribution (as $e_\infty = 0$ is a constant). □