LEARNING CONTINUOUS NORMALIZING FLOWS FOR FASTER CONVERGENCE TO TARGET DISTRIBUTION VIA ASCENT REGULARIZATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Normalizing flows (NFs) have been shown to be advantageous in modeling complex distributions and improving sampling efficiency for unbiased sampling. In this work, we propose a new class of continuous NFs, ascent continuous normalizing flows (ACNFs), that makes a base distribution converge faster to a target distribution. As solving such a flow is non-trivial and barely possible, we propose a practical implementation to learn flexibly parametric ACNFs via ascent regularization and apply in two learning cases: maximum likelihood learning for density estimation and minimizing reverse KL divergence for unbiased sampling and variational inference. The learned ACNFs demonstrate faster convergence towards the target distributions, therefore, achieving better density estimations, unbiased sampling and variational approximation at lower computational cost. Furthermore, the flows show to stabilize themselves to mitigate performance deterioration and are less sensitive to the choice of training flow length T.

1 INTRODUCTION

Normalizing flows (NFs) provide a flexible way to define an expressive but tractable distribution which only requires a base distribution and a chain of bijective transformations (Papamakarios et al., 2021). Neural ODE (Chen et al., 2018) extends discrete normalizing flows (Dinh et al., 2014; 2016; Papamakarios et al., 2017; Ho et al., 2019) to a new continuous-time analogue by defining the transformation via a differential equation, substantially expanding model flexibility in comparison to the discrete alternatives. (Grathwohl et al., 2018; Chen and Duvenaud, 2019) propose a computationally cheaper way to estimate the trace of Jacobian to accelerate training, while other methods focus on increasing flow expressiveness by e.g. augmenting with additional states (Dupont et al., 2019; Massaroli et al., 2020), or adding stochastic layers between discrete NFs to alleviate the topological constraint (Wu et al., 2020). Recent diffusion models like (Hodgkinson et al., 2020; Ho et al., 2020; Song et al., 2020; Zhang and Chen, 2021) extend the scope of continuous normalizing flows (CNFs) with stochastic differential equations (SDEs). Although these diffusion models significantly improve the quality of the generated images, introduced diffusion comes with some costs: some models do not allow for tractable density estimation; or the practical implementations of these models rely on a long chain of discretizations, thus needing relatively more computations than tractable CNF methods, which can be critical for some use cases such as online inference.

(Finlay et al., 2020; Onken et al., 2021; Yang and Karniadakis, 2020) introduce several regularizations to learn simpler dynamics using optimal transport theory, which decrease the number of discretization steps in integration and thus reduce training time. (Kelly et al., 2020) extends the L_2 transport cost to regularize any arbitrary order of dynamics. Although these regularizations are beneficial for decreasing computational costs, it does not improve the slow convergence of density to the target distributions like vanilla CNF models shown in Figure 1. To accelerate the flow convergence, STEER (Ghosh et al., 2020) and TO-FLOW (Du et al., 2022) propose to optimize flow length T in two different approaches: STEER randomly samples the length during training while TO-FLOW establishes a subproblem for T during training. To understand the effectiveness of these methods, we train multiple Neural ODE models with different flow length T_n for a 2-moon distribution. Figure 2 shows the estimated log-likelihoods of all learned flows. Although optimizing T dynamically performs a model selection during training and makes models reach higher estimates at shorter flows,

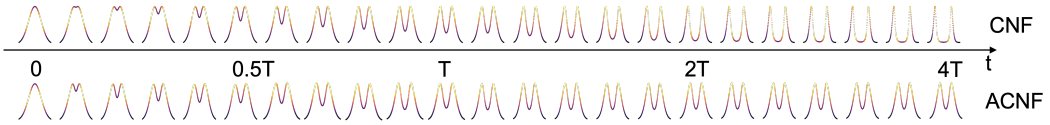

Figure 1: Distribution transformations of two learned flows for 1d Gaussian mixture from a Gaussian distribution at multiple $t \in [0, 4T]$. Although the two flows have similar densities at T, the density of ACNF converges faster to the target distribution before T and diverges slower after T than that of CNF. Color indicates the density of true Gaussian mixture.

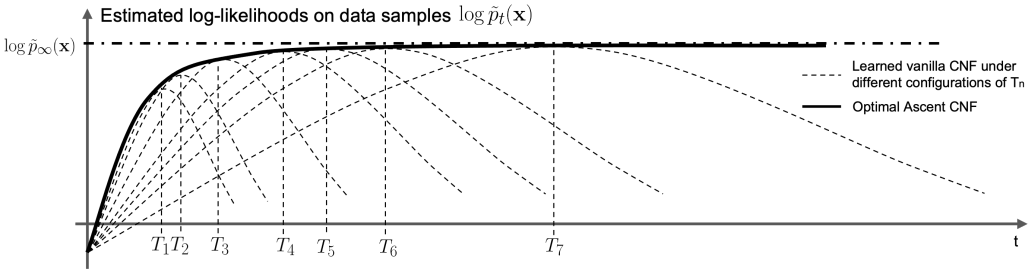

Figure 2: The log-likelihood estimates of trained vanilla CNF models with various flow length T_n and steepest ACNF with dynamics defined in eq.(6) at different t on 2-moon distribution. All vanilla CNF models reach their maximum around T_n and deteriorate rapidly afterwards while the log-likelihood estimate of ACNF elevates rapidly at initial and increases monotonically.

it cannot prevent the divergence of flows after T_n. Furthermore, flows with shorter length are more limited in expressiveness and sensitive to flow length.

In this work, we present a new family of CNFs, ascent continuous normalizing flows (ACNFs), to address the aforementioned problems. ACNF concerns a flow that makes a base distribution monotonically converge to a target distribution, and the dynamics is imposed to follow the steepest ACNF. However, solving such a steepest flow is non-trivial and barely possible. We propose a practical implementation to learn parametric ACNFs via ascent regularization. The learned ACNFs exhibit three main distinct behaviors: 1) faster convergence to target distribution with less computation; 2) self-stabilization to mitigate flow deterioration; and 3) insensitivity to flow training length T. We demonstrate the beneficial behaviors in three use cases: modeling data distributions; learning annealed samplers for unbiased sampling; and learning a tractable but more flexible variational approximation.

2 Continuous Normalizing Flows

Considering a time-t transformation $\mathbf{z}(t) = \mathbf{\Phi}_t(\mathbf{x})$ on the initial value $\mathbf{x}$, i.e. $\mathbf{z}(0) = \mathbf{x}$, the change of variable theorem reveals the relation between the transformed distribution $p_t(\mathbf{z}(t))$ and $p(\mathbf{x})$:

$$p_t(\mathbf{z}(t)) = \left|\det\left(\mathcal{J}_{\mathbf{\Phi}_t}^{-1}(\mathbf{x})\right)\right| p(\mathbf{x}), \tag{1}$$

where $\mathcal{J}_{\mathbf{\Phi}}$ is the Jacobian matrix of $\mathbf{\Phi}$. As $\mathbf{\Phi}_t$ normalizes $\mathbf{x}$ towards some base distribution, $p_t(\mathbf{z}(t))$ is referred as normalized distribution, starting from the data distribution $p(\mathbf{x})$.

Continuous normalizing flow is the infinitesimal limit of the chain of discrete flows and the infinitesimal transformation is an ordinary differential equation (ODE):

$$\frac{\mathrm{d}\mathbf{z}(t)}{\mathrm{d}t} = \frac{\mathrm{d}\mathbf{\Phi}_t(\mathbf{x})}{\mathrm{d}t} = \mathbf{f}(\mathbf{z}(t), t). \tag{2}$$

The instantaneous change of variable theorem (Chen et al., 2018, theorem 1) shows the infinitesimal changes of $\log p_t(\mathbf{z}(t))$:

$$\frac{\mathrm{d}\log p_t(\mathbf{z}(t))}{\mathrm{d}t} = -\nabla \cdot \mathbf{f}(\mathbf{z}(t), t). \tag{3}$$

Thus, the normalized distribution $\log p_t(\mathbf{z}(t))$ at t can be obtained by integrating eq.(3) backwards with a common approximation to the base distribution μ, i.e. $p_T \approx \mu$:

$$\log p_t(\mathbf{z}(t)) = \log p_T(\mathbf{z}(T)) - \int_t^T \nabla \cdot \mathbf{f}(\mathbf{z}(\tau), \tau)\mathrm{d}\tau \approx \log \mu(\mathbf{z}(T)) - \int_t^T \nabla \cdot \mathbf{f}(\mathbf{z}(\tau), \tau)\mathrm{d}\tau,$$

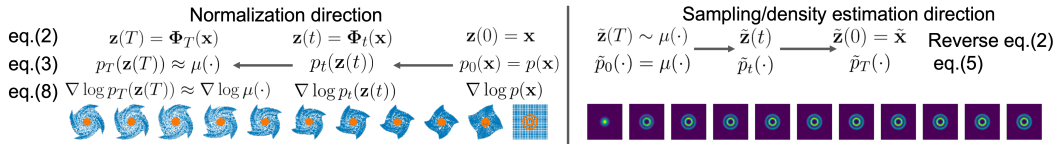

Figure 3: Upper: transformations on variables and densities on normalization and sampling directions. Lower left: data samples (orange) and the grid of states (blue) transformations along normalization direction. Lower right: density estimation $\tilde{p}_t$ along sampling direction.

where $\mathbf{z}(t) = \mathbf{x} + \int_0^t \mathbf{f}(\mathbf{z}(\tau), \tau)\mathrm{d}\tau$. The accuracy of $\log p_0(\mathbf{x})$, obtained by the right hand side, depends on the approximation error of p_T to μ and the error varies at different $\mathbf{z}(T)$. To avoid the problem and investigate how the flow length affects on the distribution modeling, we introduce $\tilde{p}_t(\mathbf{x})$ to estimate the data distribution $p(\mathbf{x})$ by a t-length flow $\mathbf{\Phi}_t$, via the change of variable theorem:

$$\tilde{p}_t(\mathbf{x}) = \left|\det\left(\mathcal{J}_{\mathbf{\Phi}_t}(\mathbf{x})\right)\right| \mu(\mathbf{\Phi}_t(\mathbf{x})). \tag{4}$$

Shown by eq.(4) and Figure 3, $\tilde{p}_t$ initiates from the base distribution, i.e. $\tilde{p}_0(\mathbf{x}) = \mu(\mathbf{\Phi}_0(\mathbf{x})) = \mu(\mathbf{x})$.Combining eq.(4) and eq.(1), the estimated likelihood of time-t flow is related to the normalized distribution $p_t(\mathbf{z}(t))$:

$$\frac{\tilde{p}_t(\mathbf{x})}{p(\mathbf{x})} = \frac{\mu(\mathbf{\Phi}_t(\mathbf{x}))}{p_t(\mathbf{\Phi}_t(\mathbf{x}))} = \frac{\mu(\mathbf{z}(t))}{p_t(\mathbf{z}(t))}.$$

It shows that as $p_t \to \mu$, $\tilde{p}_t(\mathbf{x}) \to p(\mathbf{x})$. When there exists a flow, of which the normalized density p_T is equal to the base distribution, i.e. $p_T = \mu$, then the estimated likelihood of time-T flow is exact to the true data distribution, i.e. $\tilde{p}_T(x) = p(x)$.

Like the instantaneous change of variable theorem, we derive the infinitesimal change of time-t estimated log-likelihood:

Proposition 1 (Instantaneous Change of Log-likelihood Estimate)**.** *Let $\mathbf{z}(t)$ be a finite continuous random variable at time t as the solution of a differential equation $\frac{\mathrm{d}\mathbf{z}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{z}(t), t)$ with initial value $\mathbf{z}(0) = \mathbf{x}$. Assuming that $\tilde{p}_0 = \mu$ at $t = 0$ and $\mathbf{f}$ is uniformly Lipschitz continuous in $\mathbf{z}$ and t, then the change in estimated log-likelihood $\log \tilde{p}_t(\mathbf{x})$ at t follows a differential equation:*

$$\frac{\mathrm{d}\log \tilde{p}_t(\mathbf{x})}{\mathrm{d}t} = \boldsymbol{\nabla} \cdot \mathbf{f}(\mathbf{z}(t), t) + \nabla \log \mu(\mathbf{z}(t)) \cdot \mathbf{f}(\mathbf{z}(t), t). \tag{5}$$

Proof. See Appendix A.1 for derivation and its relation to eq.(3). Unlike the integral for $p_t(\mathbf{z}(t))$ that relies on the approximation and requires to solve the whole trajectory $\mathbf{z}(\tau), \tau \in [0, T]$, the logarithm of density estimation, $\log \tilde{p}_t(\mathbf{x})$, at any or/and multiple t can be evaluated exactly in one single forward integration:

$$\log \tilde{p}_t(\mathbf{x}) = \log \mu(\mathbf{x}) + \int_0^t \left(\boldsymbol{\nabla} \cdot \mathbf{f}(\mathbf{z}(\tau), \tau) + \nabla \log \mu(\mathbf{z}(\tau)) \cdot \mathbf{f}(\mathbf{z}(\tau), \tau)\right) \mathrm{d}\tau.$$

3 Ascent Continuous Normalizing Flows

Using KL divergence as distance measure of distributions, we have the following duality:

$$\mathrm{KL}(p(\mathbf{x})||\tilde{p}_t(\mathbf{x})) = \text{const} - \int p(\mathbf{x}) \log \tilde{p}_t(\mathbf{x}) \mathrm{d}\mathbf{x} = \mathrm{KL}(p_t(\mathbf{z}(t))||\mu(\mathbf{z}(t))),$$

so that maximum likelihood learning to fit $\tilde{p}_t(\mathbf{x})$ for data from $p(\mathbf{x})$ is equivalent to minimizing the reverse KL divergence in normalization direction. Besides, we can measure the convergence rates of both KL divergences by their time derivative. Therefore, we define ascent continuous normalizing flows (ACNF) that decrease $\mathrm{KL}(p_t(\mathbf{z}(t))||\mu(\mathbf{z}(t)))$ or equivalently increase the expectation of $\log \tilde{p}_t(\mathbf{x})$ monotonically:

$$\frac{\partial}{\partial t} \int p(\mathbf{x}) \log \tilde{p}_t(\mathbf{x}) \mathrm{d}\mathbf{x} \geq 0; \text{ or } \frac{\partial}{\partial t} \mathrm{KL}(p_t(\mathbf{z}(t))||\mu(\mathbf{z}(t))) \leq 0.$$

By applying total variation, we can find the dynamics for the steepest descent of reverse KL divergence or the steepest ascent of the log-likelihood expectation:

Theorem 1 (Dynamics for Steepest Ascent Continuous Normalizing Flows)**.** *Let $\mathbf{z}(t)$ be a finite continuous random variable and the solution of a differential equation $\frac{\mathrm{d}\mathbf{z}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{z}(t), t)$ with initial value $\mathbf{z}(0) = \mathbf{x}$. Its probability $p_t(\mathbf{z}(t))$ subjects to the continuity equation $\partial_t p_t + \boldsymbol{\nabla} \cdot (p_t \mathbf{f}) = 0$. The dynamics of the steepest flow for decreasing KL$(p_t(\mathbf{z}(t))||\mu(\mathbf{z}(t)))$ is*

$$\mathbf{f}^*(\mathbf{z}(t), t) = \nabla \log \mu(\mathbf{z}(t)) - \frac{\nabla p_t(\mathbf{z}(t))}{p_t(\mathbf{z}(t))} = \nabla \log \mu(\mathbf{z}(t)) - \nabla \log p_t(\mathbf{z}(t)). \tag{6}$$

Proof. See Appendix A.2 for detailed derivation. The steepest dynamics is defined by the difference between two gradients: $\nabla \log \mu$ and $\nabla \log p_t$ w.r.t. $\mathbf{z}(t)$. There are a few important implications of eq.(6): 1) the dynamics is time-variant as p_t simultaneously evolves with the flow; 2) at time t, it only depends on the current state $\mathbf{z}(t)$, thus no history is needed; 3) the flow is initiated by the difference between $\nabla \log \mu(\mathbf{x})$ and $\nabla \log p(\mathbf{x})$, and it gradually slows down and eventually stops when $p_t(\mathbf{z}(t))$ converges to $\mu(\mathbf{z}(t))$.The convergence rate of the steepest flow can also be proven as the negative Fisher divergence, $\partial \mathrm{KL}(p_t \| \mu)/\partial t = -\mathbb{F}(p_t||\mu) = -\mathbb{E}_{p_t} \|\nabla \log \mu(\mathbf{z}) - \log p_t(\mathbf{z})\|_2^2$, therefore this deterministic CNF is related to (overdamped) Langevin diffusion, see Appendix A.3 for the derivation of convergence rate and detailed discussion of their relation.

This optimal flow can also be considered as a special instance of Wasserstein gradient flow (Ambrosio et al., 2005) with KL divergence as the energy functional. Previous works (Finlay et al., 2020; Yang and Karniadakis, 2020; Onken et al., 2021) apply the optimal transport theory to regularize flow dynamics from Euclidean space perspective, while Wasserstein gradient flow instead regularizes it in probability measure space. We refer readers to (Ambrosio et al., 2005) for accessible introduction. In some cases, the flow can be solved indirectly by introducing a potential, $V(\mathbf{z}(t), t) = p_t(\mathbf{z}(t))/\mu(\mathbf{z}(t))$, and it has a partial differential equation (PDE):

$$\frac{\partial V(\mathbf{z}, t)}{\partial t} = \Delta V(\mathbf{z}, t) + 2\nabla \log \mu(\mathbf{z}) \cdot \nabla V(\mathbf{z}, t) + \nabla \log V(\mathbf{z}, t) \cdot \nabla V(\mathbf{z}, t), \tag{7}$$

with the initial condition $V(\mathbf{z}(0), 0) = \frac{p_0(\mathbf{z}(0))}{\mu(\mathbf{z}(0))} = \frac{p(\mathbf{x})}{\mu(\mathbf{x})}$. See Appendix A.4 for its derivation. However, solving this PDE for $p_t(\mathbf{z}(t))$ is non-trivial as the closed form solution is typically unknown. JKO integration is commonly used in literature (Mokrov et al., 2021; Fan et al., 2021), which approximates the dynamics of density p_t by its time discretization, but it requires to know the initial condition while $p(\mathbf{x})$ is generally unknown and needs to be modeled from samples. (Tabak and Vanden-Eijnden, 2010) proposes to approximate $p(\mathbf{x})$ by the spatial discretization of samples, which hardly can be scale up even for intermediate dimensions.

To tackle these difficulties and accelerate flows for faster convergence, we propose ascent regularization to learn parametric ACNFs, inspired by previous works (Yang and Karniadakis, 2020; Onken et al., 2021; Finlay et al., 2020; Kelly et al., 2020; Ghosh et al., 2020) that enforce certain behaviors of flows via regularization in training. Ascent regularization penalizes the difference between the parametric dynamics and the steepest dynamics in eq.(6), by minimizing $\|\mathbf{f}_{\boldsymbol{\theta}} - \mathbf{f}^*\|_2^2$, which needs to evaluate score function $\nabla \log p_t(\mathbf{z}(t))$. We propose the instantaneous change of the score function:

Theorem 2 (Instantaneous Change of Score Function)**.** *Let $\mathbf{z}(t)$ be a finite continuous random variable with probability density $p_t(\mathbf{z}(t))$ at time t. Let $\frac{\mathrm{d}\mathbf{z}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{z}(t), t)$ be a differential equation describing a continuous-in-time transformation of $\mathbf{z}(t)$. Assuming that $\mathbf{f}$ is uniformly Lipschitz continuous in $\mathbf{z}$ and t, the infinitesimal change in the gradient of log-density at t is*

$$\frac{\mathrm{d}\nabla \log p_t(\mathbf{z}(t))}{\mathrm{d}t} = -\nabla \log p_t(\mathbf{z}(t)) \frac{\partial \mathbf{f}(\mathbf{z}(t), t)}{\partial \mathbf{z}(t)} - \nabla \left(\boldsymbol{\nabla} \cdot \mathbf{f}(\mathbf{z}(t), t)\right). \tag{8}$$

Proof. See Appendix A.5 for detailed derivation. $\nabla \log p(\mathbf{z}(t), t)$ follows a linear matrix differential equation, where the linear coefficient matrix is Jacobian and the bias term is the gradient of divergence. To be noted, an alternative proof is derived in concurrent work (Lu et al., 2022, theorem D.1).

We demonstrate ACNF for two learning cases: maximum likelihood learning for data modeling and density estimation in Section 4; minimizing reverse KL divergence for learning annealed samplers for unbiased sampling in Section 5.

Algorithm 1 Maximum likelihood learning of ACNF with ascent regularization

Require: Data samples $\mathbf{X} = \{\mathbf{x}_j\}_{j=1,\dots,N}$, parameteric dynamics of flow $\mathbf{f}_{\boldsymbol{\theta}}$, length of flow T, ascent regularization coefficient λ, base distribution μ
Initialize $\boldsymbol{\theta}$
while $\boldsymbol{\theta}$ is not converged **do**
Sample a mini-batch of data $\mathbf{x}^i \sim \mathbf{X}$
Integrate augmented states $[\mathbf{z}^i(t), \log \tilde{p}_t(\mathbf{x}^i)]$ forward with initial value $[\mathbf{x}^i, \mathbf{0}]$ from 0 to T
Integrate augmented states $[\mathbf{z}^i(t), \nabla \log p_t(\mathbf{z}^i(t))]$ backwards with initial value $[\mathbf{z}^i(T), \nabla \log \mu(\mathbf{z}^i(T))]$ from T to 0
Compute loss function $\mathcal{L}$ in eq.(9) and $\nabla_{\boldsymbol{\theta}}\mathcal{L}$ by adjoint sensitivity method
Update $\boldsymbol{\theta}$ by gradient descent algorithm
end while

Algorithm 2 Training ACNF as annealed sampler for unbiased sampling with ascent regularization

Require: target distribution $\pi = \gamma/Z$, parameteric dynamics of flow $\mathbf{f}_{\boldsymbol{\theta}}$, length of flow T, number of samples N, ascent regularization coefficient λ, base distribution μ
Initialize $\boldsymbol{\theta}$
while $\boldsymbol{\theta}$ is not converged **do**
Sample $\mathbf{z}_0^i \sim p_0 = \mu$
Evaluate $\log \mu(\mathbf{z}_0^i)$ and $\nabla \log \mu(\mathbf{z}_0^i)$
Integrate augmented states $[\mathbf{z}^i(t), \log p_t(\mathbf{z}^i(t)), \nabla \log p_t(\mathbf{z}^i(t))]$ with initial value $[\mathbf{z}_0^i, \log \mu(\mathbf{z}_0^i), \nabla \log \mu(\mathbf{z}_0^i)]$ from 0 to T
Evaluate $\log w(\mathbf{z}^i(T)) = \log \gamma(\mathbf{z}^i(T)) - \log p_T(\mathbf{z}^i(T))$
Compute loss function $\mathcal{L}$ in eq.(10) and $\nabla_{\boldsymbol{\theta}}\mathcal{L}$ by adjoint sensitivity method
Update $\boldsymbol{\theta}$ by gradient descent algorithm
end while

4 Maximum Likelihood Learning of ACNF for Density Estimation via Ascent Regularization

For maximizing likelihood learning of $\tilde{p}_T$ to fit data, the total objective with ascent regularization is:

$$\min_{\mathbf{f}} \mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(-\log \tilde{p}_T(\mathbf{x}^i;\boldsymbol{\theta}) + \lambda \int_0^T \| \left(\nabla \log p_t(\mathbf{z}^i(t);\boldsymbol{\theta}) - \nabla \log \mu(\mathbf{z}^i(t))\right) + \mathbf{f}(\mathbf{z}^i(t), t; \boldsymbol{\theta})\|_2^2 \mathrm{d}t\right), \tag{9}$$

where λ is the ascent regularization coefficient to control the trade-off between maximizing likelihood and regularization on the ascent behavior of the learned dynamics. When $\lambda = 0$, ACNF degrades to CNF. The first term in eq.(9) is obtained by integrating eq.(5) over $[0,\ T]$, simultaneously with $\mathbf{z}(t)$, while the ascent regularization can be integrated backwards with augmented initial $[\mathbf{z}(T), \nabla \log p_T(\mathbf{z}(T))]$, with $\nabla \log p_T(\mathbf{z}(T)) \approx \nabla \log \mu(\mathbf{z}(T))$. We summarize the pseudo-code for maximum likelihood learning of ACNFs in Algorithm 1. We show the interpretation of ascent regularization as score matching in Section A.6 in Appendix, thus Algorithm 1 can be implemented in more efficient ways like (Lu et al., 2022; Song et al., 2021) for some cases.

5 Learning ACNF as Annealed Sampler for Unbiased Sampling

Except modeling data samples and performing density estimation, NF as a sampler shows to be more sample efficient in Annealed Importance Sampling (AIS) (Neal, 2001) when comparing to classic MCMC methods. A typical AIS and its extension use a sequence of annealed targets $\{\pi_k\}_{k=0:K}$ that bridges an easy-to-sample and tractable distribution $\pi_0 = \mu$ to the target $\pi_K := \pi = \gamma(\cdot)/Z$. SNF (Wu et al., 2020) and AFT (Arbel et al., 2021) propose to fit K discrete NFs and each NF approximates the transport map between π_{k-1} and π_k. However, the sampling convergence rate largely depends on the pre-defined annealed targets. Besides, more steps are needed to decrease the variance of the estimator, which comes at an additional computational cost (Doucet et al., 2022).

As ACNF can also define a flow from a base distribution to a target distribution, it can learn a continuous annealed target flow replacing the pre-define ones and later generates samples. Different to (Grosse et al., 2013), the annealed targets by ACNF do not require the specific form of distributions. As the distribution by ACNF is enforced to converge faster to the target distribution, ACNF sampler potentially generates better samples than CNF or linear annealed scheduling especially at limited steps of sampling, thus estimates, e.g. on logarithm of normalization constant $\log Z$, are more accurate.

Different to maximum likelihood in Section 4, training ACNF for an annealed sampler is to minimize the reverse KL divergence $\mathrm{KL}(p_T(\mathbf{z}(T))\|\pi(\mathbf{z}(T)))$. It can be evaluated up to a constant by

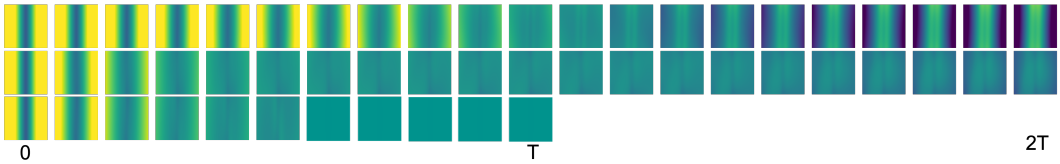

Figure 4: Comparison on log potential field along the flow by trained vanilla CNF and ACNF with $\lambda = 1$ and the numerical PDE solutions of eq.(7) for 2-modal Gaussian mixture at $t \in [0, 2T]$. Color indicates the value of field: turquoise is 0 and the lighter the color is the larger the value is.

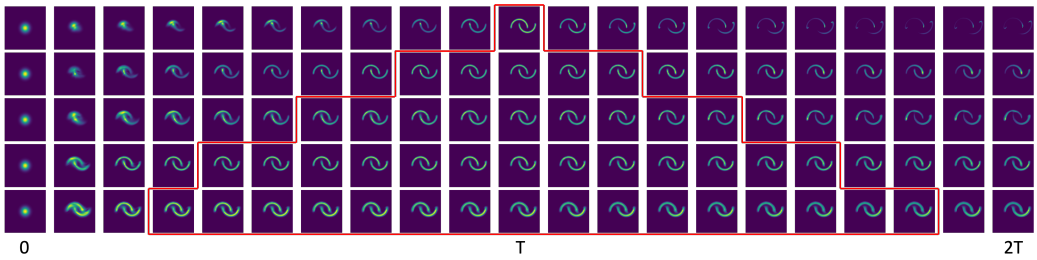

Figure 5: Comparison on density evaluation of trained vanilla CNF and ACNF with $\lambda = 0.0001, 0.0005, 0.001, 0.005$ on 2-moon distribution along integral $t \in [0, 2T]$.

the logarithm of importance weights, $\log w(\mathbf{z}(T)) = \log \gamma(\mathbf{z}(T)) - \log p_T(\mathbf{z}(T))$. With ascent regularization, the total objective becomes:

$$\min_{\mathbf{f}} \mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left(-\log w(\mathbf{z}^i(T); \boldsymbol{\theta}) + \lambda \int_0^T \| \left(\nabla \log p_t(\mathbf{z}^i(t); \boldsymbol{\theta}) - \nabla \log \mu(\mathbf{z}^i(t)) \right) + \mathbf{f}(\mathbf{z}^i(t), t; \boldsymbol{\theta}) \|_2^2 \mathrm{d}t \right), \tag{10}$$

where $\mathbf{f}(\mathbf{z}(t), t; \boldsymbol{\theta})$ is the annealed generation dynamics. Unlike the previous section, as the base distribution is known here, $\log p_t(\mathbf{z}(t))$ and $\nabla \log p_t(\mathbf{z})$ are integrated simultaneously with $\mathbf{z}(t) = \mathbf{z}(0) + \int_0^t \mathbf{f}(z(\tau), \tau) \mathrm{d}\tau$ with a sample $\mathbf{z}(0) \sim \mu$. We summarize pseudo-code for learning ACNF annealed sampler in Algorithm 2. Once ACNF sampler is learned, it can generate unbiased samples: generate one-shot samples from ACNF with flow length t according to computation budget; correct samples by resampling according to importance weights like (Müller et al., 2019) or by Markov Chain Monte Carlo methods with Metropolis-Hastings correction.

6 Experiments

6.1 Density Estimation on Toy 2D Distributions

Before we deploy ACNF for modeling complex distributions, we first examine it on a 2-modal Gaussian mixture and use a standard Gaussian as the base distribution in 2D. Figure 4 shows that the potential field of learned ACNF is very similar to the PDE solutions in eq.(7) while CNF potential converges much slower than ACNF and then diverges after T. See Appendix A.7 for experiment details and comparison on more λ, T and other regularization methods.

We then train vanilla CNF, RNODE (Finlay et al., 2020) and ACNFs to model data from various 2D toy distributions and visualize the density estimation along flows. Figure 5 shows the densities at $t \in [0, 2T], T = 10$ by learned CNF and ACNFs with various regularization coefficients for 2-moon distribution. The densities that are close to the target distribution are highlighted inside the red border. We show that even slight regularization makes the learned flows to 1) converge much faster towards the target; 2) maintain the best estimations for long time after T. The quantitative evaluation on the log-likelihood estimates indicates the same conclusion, seen from the left of Figure 6. More analysis on different T and experiment setups are given in Appendix A.8.

One may suspect that more complex dynamics explain the faster ascent of likelihood estimates. To validate the actual improvements by ACNF, we evaluate the number of function evaluations (NFEs) by counting the times when an adaptive step size solver calls the dynamics function in the integration, with and without log-likelihood for all models. The marks in the left of Figure 6 show NFEs along flow while the middle one plots log-likelihood estimates versus NFEs. ACNFs clearly demonstrate

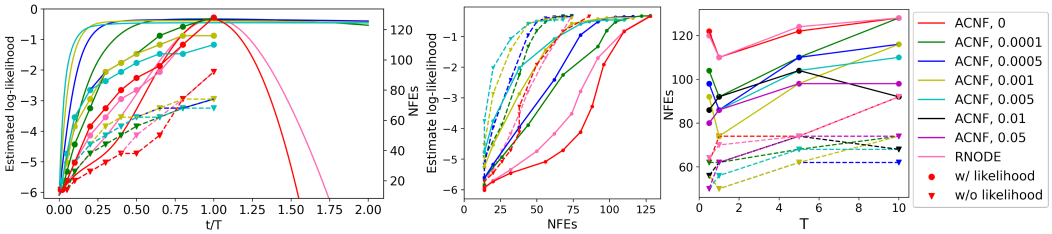

Figure 6: Left: comparison on estimated log-likelihoods of models trained under different regularization λ as Figure 5. Middle: log-likelihood vs NFE as the left figure. Right: comparison on NFEs evaluated at $t/T = 1$ of vanilla CNF, RNODE and ACNF trained with various flow length T and λ.

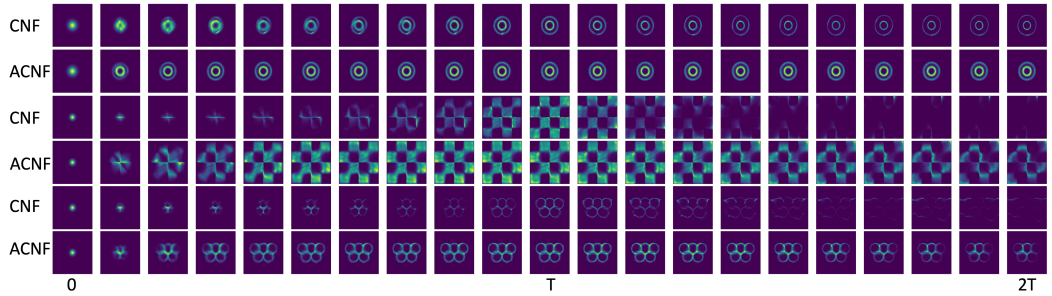

Figure 7: Comparison on density estimations of trained ACNF and vanilla CNF models on various two-dimensional toy distributions along flows with increasing $t \in [0, 2T]$

that they learn even less complex dynamics than CNF and RNODE, and log-likelihood gain per NFE of ACNFs are much higher than the two baselines especially at early stage. Among different ascent regularization, a larger coefficient leads to more rapid gain on the log-likelihoods initially, however, too large regularization over-constrains models to reach a good maximum. A moderate regularization benefits on the balance of maximum likelihood and faster convergence. Furthermore, we report NFEs at $t/T = 1$ for CNF, RNODE and ACNFs trained with various $T = 0.5, 1, 5, 10$ and $\lambda = 0.0001, 0.005, 0.001, 0.005, 0.01, 0.05$ on the right of Figure 6. ACNFs have generally lower NFEs than CNF and RNODE and most models report lowest NFEs at $T = 1$. It indicates that optimizing T by TO-FLOW (Du et al., 2022) and STEER (Ghosh et al., 2020) can decrease computational cost, however, the gain is marginal as NFEs do not grow linearly to integration length. More importantly, even an appropriate T cannot prevent slow convergence and density deterioration of CNF as shown in Figure 14 in Appendix A.8. Figure 7 shows density evaluations on more multi-modal distributions. Learned ACNFs show faster convergence than CNFs for all distributions and give a higher maximum density estimation on the challenging task, e.g. Olympics distribution.

6.2 Density Estimation on Real Datasets

We demonstrate density estimations on real-world benchmarks datasets including POWER, GAS, HEPMASS, MINIBOONE from the UCI machine learning data repository and BSDS300 natural image patches. Like FFJORD, all tabular datasets and BSDS300 are pre-processed as in (Papamakarios et al., 2017). Table 1 reports the averaged NLLs on test data for FFJORD, RNODE and ACNF trained with different λ. The detailed description of experiments and models refers to Appendix A.9. Although FFJORD with multi-step flows increases the flexibility of flows, it tends to have a worse performance than the base distribution initially and then improves NLL mainly at the late stage of flows. A larger ascent regularization of ACNFs contributes to more rapid initial increases on NLL that these flows transform the base distribution faster towards the data distribution. When training on HEPMASS and BSDS300, a too large regularization coefficient impedes model to converge.

6.3 ACNF as a faster annealing sampling proposal for unbiased sampling

Following Algorithm 2, we train CNF and ACNFs with regularization coefficients $\lambda = 0.0001, 0.01, 0.01$ for learning annealed targets. We evaluate the normalization constants of a Gaussian mixture with 8 components whose means are fixed evenly in space and standard deviation as 0.3 and the base distribution chooses a Gaussian $\mathcal{N}(\mathbf{0}, 3^2\mathbf{I})$ to have adequate support for the target distribu-

Table 1: Averaged negative log-likelihoods (NLLs) on test data for density estimation.

Datasets	Model	$0.1T$	$0.25T$	$0.5T$	$0.75T$	T	$t > T$‡
POWER	FFJORD†§	7.47	5.97	4.63	2.55	−0.42	5.02
	RNODE	7.80	6.11	4.80	2.64	−0.46	4.59
	ACNF, $1e^{-4}$	7.03	5.02	2.63	0.81	−0.40	0.12
	ACNF, $1e^{-3}$	6.26	4.15	1.99	**0.48**	−0.44	0.02
	ACNF, $1e^{-2}$	**5.43**	**3.47**	**1.87**	**0.48**	**−0.37**	**−0.26**
GAS	FFJORD†§	11.76	241.13	145.75	132.12	−8.60	279.97
	RNODE	12.03	102.77	78.20	45.38	**−8.79**	90.23
	ACNF, $1e^{-4}$	7.25	3.31	−1.28	−5.5	−8.56	−6.88
	ACNF, $1e^{-3}$	5.58	1.65	−1.73	−5.61	−8.45	−7.63
	ACNF, $1e^{-2}$	**3.57**	**−0.14**	**−3.73**	**−6.42**	−8.57	**−8.33**
HEPMASS	FFJORD†§	35.51	40.19	82.08	106.81	14.90	42.09
	RNODE	35.02	40.59	68.91	96.47	15.34	39.09
	ACNF, $1e^{-4}$	27.75	24.88	20.63	17.17	14.95	**15.36**
	ACNF, $1e^{-3}$	**25.24**	**22.78**	**18.81**	**15.94**	**14.88**	15.52
MINIBOONE	FFJORD§	58.54	53.98	42.12	24.55	10.50	16.25
	RNODE	57.98	53.65	41.79	24.12	10.77	15.86
	ACNF, $1e^{-4}$	54.15	43.76	29.29	19.00	10.63	13.95
	ACNF, $1e^{-3}$	52.64	42.87	28.92	18.72	**10.40**	12.57
	ACNF, $1e^{-2}$	**51.52**	**40.52**	**26.53**	**17.16**	10.95	**12.14**
BSDS300	FFJORD†§	41.72	30.11	0.91	−85.92	−156.60	−77.20
	RNODE	42.38	33.29	1.74	−84.03	**−156.71**	−89.35
	ACNF, $1e^{-4}$	**35.24**	24.10	−14.75	−100.35	−156.0	**−125.67**
	ACNF, $1e^{-3}$	37.45	**21.53**	**−17.84**	**−105.11**	−156.5	−120.89

† FFJORD uses multi-step flow models for some datasets, so the total length of flow is no longer training configuration of T but T times the number of flow steps. T listed here refers the total length of flow. ‡ The flow length t after T is set slightly different among datasets due to the multi-step FFJORD: $1.2T$ for POWER and GAS, $1.1T$ for HEPMASS, and $1.25T$ for MINIBOONE and BSDS300, but it is always the *same* among different models. §FFJORDs are trained to match the performance as originally reported.

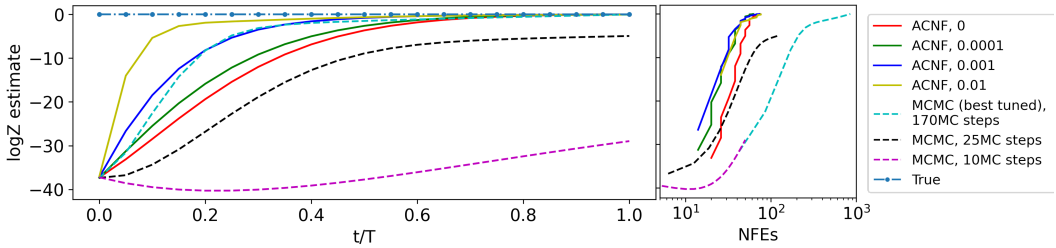

Figure 8: Left: comparison on estimated $\log Z$ by different methods along the flow over 5 different runs. Right: estimated $\log Z$ vs NFEs. See Figure 18 for generated sample comparisons.

tion. Figure 8 compares the estimates of $\log Z$ along flows and reports estimates versus NFEs. Except CNF and ACNFs, we also evaluate the linear annealed target $\log \gamma_k(\cdot) = \beta_k \log \gamma(\cdot) + (1-\beta_k) \log \pi_0$, where scheduling is $\beta_k = k/K = t_k/T$ and $K = 20$, using $\{170, 25, 10\}$-step Metropolis sampler between each intermediate target. As ACNFs converge faster towards the target distribution than CNF, the one-shot samples from ACNFs are less biased than those from CNF especially at the early stage of the flows. Besides, ACNFs are more computationally efficient in terms of accuracy gain per NFEs. ACNFs with coefficients $\lambda = 0.01, 0.001$ shows less biased estimates earlier than the best tuned linear annealed target. Besides, the linear annealed target requires at least 1 order more computations than CNF and ACNFs for comparable accuracy due to slow mixing of Metropolis sampler, and is sensitive to the number of MC steps. Therefore, ACNFs prevent slow convergence of sampling by the suboptimal annealed target and have more accurate estimates with less computation. Figure 18 in Appendix A.10 shows generated samples by all methods in Figure 8. Adding MC steps with learned ACNFs can further accelerate sample convergence to target and increase the expressiveness of ACNF.

6.4 Variational Inference with ACNFs

In addition to density estimation and unbiased sampling, CNFs provides more flexible variational approximations to improve variational inference (Rezende and Mohamed, 2015). We follow the

Table 2: Averaged negative ELBO on MNIST datasets under different length of flows t.

Model	$0.1T$	$0.25T$	$0.5T$	$0.75T$	T	$1.2T$
VAE-FFJORD	85.90	85.07	83.96	83.26	82.88(82.82†)	89.74
VAE-ACNF, $1e^{-4}$	85.62	84.60	83.45	83.06	**82.74**	85.67
VAE-ACNF, $1e^{-3}$	**84.70**	**83.95**	**83.22**	**82.53**	82.80	**84.37**

† originally reported in FFJORD

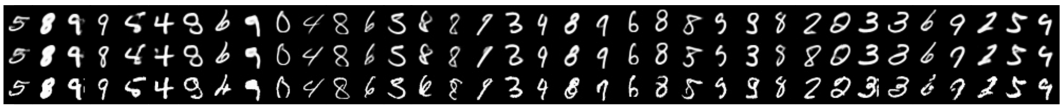

Figure 9: Reconstructions from VAE-ACNF and VAE and original data for some challenging samples.

experiment setup as (Grathwohl et al., 2018), that uses an encoder/decoder with a CNF architecture, of which the encoder gives the base latent posterior approximation for CNF and the decoder transforms latent inference at the end of the flow to the observation dimension. To train a VAE-ACNF model, the log weight in eq.(10) is replaced by an ELBO estimate like (Kingma and Welling, 2014).

We compare VAE-ACNF to VAE-FFJORD and vanilla VAE without flow on MNIST data. To make a fair comparison, we fix the learned encoder-decoder when training all three models. A detailed description of model architecture and experimental setup can be found in Appendix A.11. The averaged negative ELBO on test data of vanilla VAE is 86.50 and that of VAE-FFJORD and VAE-ACNFs with two regularization $\lambda = 1e^{-4}, 1e^{-3}$ along the flows are reported in Table 2. VAE-ACNFs show faster gains on ELBO at early stage of the flows, compared to VAE-FFJORD, and a larger coefficient accelerates the convergence of approximation to true posterior. VAE-ACNFs circumvent the ELBO deterioration which happens to VAE-FFJORD, thanks to the self-stabilization behavior of ACNF. We also show some reconstruction examples from VAE-ACNF in Figure 9 and Figure 19 in Appendix A.11. These reconstructions tend to correct some defects from original images, add details to strengthen identities while remaining sharp as the original images.

7 Scopes and Limitations

While we have demonstrated that ascent regularization is effective to learn ACNF that converges faster to a target distribution, there are still a number of limitations, which we would like to address in the future. First, more efficient implementations on score function evaluation by e.g. estimators or model design or via score matching learning (Song et al., 2021) can accelerate the model training for high-dimensional problems. Second, Hypernet (Ha et al., 2016) is found very effective to illustrate faster convergence behavior of ACNF as time exerts a large impact on the dynamics, however, it is slower to train than other simpler network architectures. A better architecture design for dynamics can improve the training speed while maintaining the faster convergence behavior of flows. Third, although the proposed ACNF and ascent regularization are discussed under the framework of CNF, the concept can be easily extended and explored for score-based models and other stochastic flows. Finally, the proposed ACNF and ascent regularization can be extended for a sequence of distributions, e.g. inference of sequential data, and computationally efficient ACNF may benefit online inference.

8 Conclusion

We have proposed ACNFs, a new class of CNFs, that define flows with monotonic convergence toward a target distribution. We derive the dynamics for the steepest ACNF and propose a practical implementation to learn parametric ACNFs via ascent regularization. We demonstrate ACNF in three use cases: modeling data and performing density estimation, learning an annealed sampler for unbiased sampling, and learning variational approximation for variational inference. The learned ACNFs illustrate three beneficial behaviors: 1) faster convergence to the target distribution with less computation; 2) self-stabilization to mitigate performance deterioration; 3) insensitivity to flow training length T. Experiments on both toy distributions and real-world datasets demonstrate the effectiveness of ascent regularization on learning ACNFs for various purposes.

Ethics Statement As this work mainly concerns to propose a flow-based model and practical implementation for learning, it does not involve human subjects, practices to data set releases, or security and privacy issue. At this stage of study, we do not foresee the effects of potential system failures due to weaknesses in the proposed methods.

Reproducibility Statement All proposition (proposition 1) and theorems (theorem 1 and 2) proposed in this paper are proved with details in Appendix A.1, A.2 and A.5 as well as other minor derivations mentioned in main body of the paper. The pseudo-code for both learning cases are provided in Algorithm 1 and Algorithm 2. The datasets, models, experiment setups for each demonstration are described in details in Appendix A.8 $\sim$ A.11. Furthermore, we attach some source codes in supplementary material for further checkup.

REFERENCES

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6572–6583, 2018.

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. *arXiv preprint arXiv:1605.08803*, 2016.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, pages 2722–2730. PMLR, 2019.

Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2018.

Ricky TQ Chen and David K Duvenaud. Neural networks with cheap differential operators. *Advances in Neural Information Processing Systems*, 32, 2019.

Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. In *Advances in Neural Information Processing Systems*, pages 3140–3150, 2019.

Stefano Massaroli, Michael Poli, Jinkyoo Park, Atsushi Yamashita, and Hajime Asama. Dissecting neural odes. *Advances in Neural Information Processing Systems*, 33:3952–3963, 2020.

Hao Wu, Jonas Köhler, and Frank Noé. Stochastic normalizing flows. *arXiv preprint arXiv:2002.06707*, 2020.

Liam Hodgkinson, Chris van der Heide, Fred Roosta, and Michael W Mahoney. Stochastic normalizing flows. *arXiv preprint arXiv:2002.09547*, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.

Qinsheng Zhang and Yongxin Chen. Diffusion normalizing flow. *Advances in Neural Information Processing Systems*, 34, 2021.

Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam M Oberman. How to train your neural ode: the world of jacobian and kinetic regularization. In *International Conference on Machine Learning*, 2020.

Derek Onken, S Wu Fung, Xingjian Li, and Lars Ruthotto. Ot-flow: Fast and accurate continuous normalizing flows via optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1–18, 2021.

Liu Yang and George Em Karniadakis. Potential flow generator with l2 optimal transport regularity for generative models. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Jacob Kelly, Jesse Bettencourt, Matthew J Johnson, and David K Duvenaud. Learning differential equations that are easy to solve. *Advances in Neural Information Processing Systems*, 33:4370–4380, 2020.

Arnab Ghosh, Harkirat Behl, Emilien Dupont, Philip Torr, and Vinay Namboodiri. Steer: Simple temporal regularization for neural ode. *Advances in Neural Information Processing Systems*, 33: 14831–14843, 2020.

Shian Du, Yihong Luo, Wei Chen, Jian Xu, and Delu Zeng. To-flow: Efficient continuous normalizing flows with temporal optimization adjoint with moving speed. *arXiv preprint arXiv:2203.10335*, 2022.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005.

Petr Mokrov, Alexander Korotin, Lingxiao Li, Aude Genevay, Justin M Solomon, and Evgeny Burnaev. Large-scale wasserstein gradient flows. *Advances in Neural Information Processing Systems*, 34, 2021.

Jiaojiao Fan, Amirhossein Taghvaei, and Yongxin Chen. Variational wasserstein gradient flow. *arXiv preprint arXiv:2112.02424*, 2021.

Esteban G Tabak and Eric Vanden-Eijnden. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.

Cheng Lu, Kaiwen Zheng, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Maximum likelihood training for score-based diffusion odes by high order denoising score matching. In *International Conference on Machine Learning*, pages 14429–14460. PMLR, 2022.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34:1415–1428, 2021.

Radford M Neal. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.

Michael Arbel, Alex Matthews, and Arnaud Doucet. Annealed flow transport monte carlo. In *International Conference on Machine Learning*, pages 318–330. PMLR, 2021.

Arnaud Doucet, Will Grathwohl, Alexander GDG Matthews, and Heiko Strathmann. Score-based diffusion meets annealed importance sampling. *arXiv preprint arXiv:2208.07698*, 2022.

Roger B Grosse, Chris J Maddison, and Russ R Salakhutdinov. Annealing between distributions by averaging moments. *Advances in Neural Information Processing Systems*, 26, 2013.

Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM Transactions on Graphics (TOG)*, 38(5):1–19, 2019.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016.

Qiang Liu. Stein variational gradient descent as gradient flow. *Advances in neural information processing systems*, 30, 2017.

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.