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. Although 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) proposes 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 samples, introduced diffusion comes with a cost: 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 compared to tractable CNF methods, which can be critical for some use cases such as online inference.

(Finlay et al., 2020; Onken et al., 2021; Kelly et al., 2020; 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 $T$ to be updated by gradient descent. To understand the effectiveness of these methods, we train multiple Neural ODE models with different length $T_n$ for a 2-moon distribution and Figure 2 shows the estimated log-likelihoods along all learned flows. Although optimizing $T$ dynamically performs model selection during training and encourages
In this work, we present a new family of CNFs, ascent continuous normalizing flows (ACNFs), to address the aforementioned questions. ACNF concerns a flow that makes a base distribution monotonically converge to a target distribution, and the dynamics of the steepest ACNF is derived. Although 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 $T_n$. 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 $z(t) = \Phi_t(x)$ on the initial value $x$, i.e. $z(0) = x$, the change of variable theorem reveals the relation between the transformed distribution $p_t(z(t))$ and $p(x)$:

$$p_t(z(t)) = |\text{det}(J_{\Phi_t}(x))| p(x),$$

(1)

where $J_{\Phi_t}$ is the Jacobian matrix of the transformation $\Phi$. As transformation $\Phi$, normalizes $x$ towards some base distribution, $p_t(z(t))$ is commonly referred as normalized distribution, starting from $p(x)$.

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

$$\frac{dz(t)}{dt} = \frac{d\Phi_t(x)}{dt} = f(z(t), t).$$

(2)

The instantaneous change of variable theorem (Chen et al., 2018, theorem 1) gives the total derivative of $\log p_t(z(t))$:

$$\frac{d\log p_t(z(t))}{dt} = -\nabla \cdot f(z(t), t).$$

(3)

Commonly, we assume the normalized distribution $p_T(z(T))$ at $T$ to be the base distribution $\mu$, e.g. Gaussian. $\log p_t(z(t))$ at $t$ can be integrated using solution $z(T)$:

$$z(t) = x + \int_0^t f(z(\tau), \tau)d\tau, \quad \log p_t(z(t)) = \log \mu(z(T)) - \int_t^T \nabla \cdot f(z(\tau), \tau)d\tau.$$
We define ascent continuous normalizing flows (ACNF) that decrease when only positive.

When only one direction measurement enables the maximum likelihood learning to fit the whole trajectory \( z(t), t \in [0, T] \). To avoid the extra integration on \([t, T]\), we present another density \( \tilde{p}_t(x) \) via the change of variable theorem:

\[
\tilde{p}_t(x) = \left| \det \left( J_{\Phi_t} (x) \right) \right| \mu(\Phi_t(x)),
\]

which initiates at \( \tilde{p}_0(x) = \mu(\Phi_0(x)) = \mu(x) \) and reveals the density estimation at \( x \) at \( t \). To distinguish this density from \( p_t \) in eq. (5), \( \tilde{p}_t(x) \) is commonly referred to as time- \( t \) estimated log-likelihood. See Figure 3 for the summary of transformations on variables and densities on both normalization and generation directions.

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

\[
\frac{d\log \tilde{p}_t(x)}{dt} = \nabla \cdot f(z(t), t) + \nabla \log \mu(z(t)) \cdot f(z(t), t).
\]

Proof. See Appendix A.1 for detailed derivation and its relation to the instantaneous change of variable theorem. It enables us to evaluate density estimations of \( x \) at any \( t \) in one single integration:

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

Combining eq. (4) and eq. (5), the estimated likelihood of time- \( t \) flow is related to the normalized distribution \( p_t(z(t)) \):

\[
\tilde{p}_t(x) = \frac{\mu(\Phi_t(x))}{p_t(\Phi_t(x))} = \frac{\mu(z(t))}{p_t(z(t))},
\]

which shows that as \( p_t \rightarrow \mu, \tilde{p}_t(x) \rightarrow p(x) \). Using KL divergence to measure the distance of distributions, the change of variable theorem shows the duality of KL divergence:

\[
\text{KL}(p(x) || \tilde{p}_t(x)) = \text{const} - \int p(x) \log \tilde{p}_t(x) dx = \text{KL}(p_t(z(t)) || \mu(z(t)))).
\]

This dual direction measurement enables the maximum likelihood learning to fit \( \tilde{p}_t(x) \) for samples from \( p(x) \) by minimizing the reverse KL divergence. Furthermore, we can consider KL divergence as a function of time \( t \) and define its time derivative as the convergence rate of log-likelihood estimate to true data distribution or normalized distribution to base distribution on both directions.

3 Ascent Continuous Normalizing Flows

We define ascent continuous normalizing flows (ACNF) that decrease KL divergence of \( p_t(z(t)) || \mu(z(t)) \) or equivalently increase the expectation of log-likelihood monotonically:

\[
\frac{\partial}{\partial t} \int p(x) \log \tilde{p}_t(x) dx \geq 0; \quad \text{or} \quad \frac{\partial}{\partial t} \text{KL}(p_t(z(t)) || \mu(z(t))) \leq 0.
\]

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

Theorem 1 (Dynamics for Steepest Ascent Continuous Normalizing Flows). Let \( z(t) \) be a finite continuous random variable and the solution of a differential equation \( \frac{dz(t)}{dt} = f(z(t), t) \) with initial
value \( z(0) = x \). Its probability \( p_t(z(t)) \) subjects to the continuity equation \( \partial_t p_t + \nabla \cdot (p_t f) = 0 \). The dynamics of the steepest flow for decreasing KL which is generally unavailable and needs to be modeled from samples, or by approximating it w.r.t. Theorem 2 (Mokrov et al., 2021; Fan et al., 2021), however, it requires to know the initial distributions. This flow is also a special instance of Wasserstein gradient flow that the energy functional of the training, we propose ascent regularization to learn parametric ACNFs which penalizes the difference.

\[
\begin{align*}
\nabla^* (z(t), t) &= \nabla \log \mu (z(t)) - \frac{\nabla p_t (z(t))}{p_t (z(t))} = \nabla \log \mu (z(t)) - \nabla \log p_t (z(t)).
\end{align*}
\]

**Proof.** See Appendix A.2 for detailed derivation. The convergence rate of the steepest flow can also be proved to be negative Fisher divergence, \(-F(p_t || \mu) = -\mathbb{E}_{p_t} \| \nabla \log \mu (z) - \log p_t (z) \|^2_2 \), therefore this deterministic CNF is related to overdamped Langevin dynamics, see Appendix A.3 for the derivation of the convergence rate and detailed discussion of their relation.

The steepest dynamics is defined by the difference between two gradients: \( \nabla \mu \) and \( \nabla \log p_t \) w.r.t. \( z(t) \). 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 \( z(t) \), therefore no history needs to be saved; 3) when \( t = 0 \), the flow is initiated by the gradient difference between \( \nabla \log \mu (x) \) and \( \log p(x) \), and it gradually slows down when \( p_t (z(t)) \) converges to \( \mu (z(t)) \).

This flow is also a special instance of Wasserstein gradient flow that the energy functional of the flow is KL divergence. Several previous works (Finlay et al. 2020; Yang and Karniadakis 2020; Onken et al. 2021) apply optimal transport theory to regularize flow dynamics from Euclidean space perspective, while Wasserstein gradient flow instead regularizes it in a probability measure space. We refer readers to (Ambrosio et al. 2005) for accessible introduction. In some cases, this Wasserstein gradient flow can be solved by introducing an auxiliary variable called potential, \( V(z(t), t) \), defined by ratio \( p_t (z(t))/\mu (z(t)) \), and the potential has a partial differential equation (PDE):

\[
\frac{\partial V(z, t)}{\partial t} = \Delta V(z, t) + 2 \nabla \log \mu (z) \cdot \nabla V(z, t) + \nabla \log V(z(t)) \cdot \nabla V(z, t),
\]

with the initial condition \( V(z(0), 0) = \frac{p_t (z(0))}{\mu (z(0))} = \frac{p(x)}{\mu (x)} \). See Appendix A.4 for its derivation. However, solving this PDE is non-trivial since a closed form solution is typically unknown. A common approach is to use JKO integration to approximate the dynamics of density \( p_t \) or potential (Mokrov et al. 2021; Fan et al. 2021), however, it requires to know the initial distributions \( p(x) \), which is generally unavailable and needs to be modeled from samples, or by approximating it via spatial discretization of samples (Tabak and Vanden-Eijnden 2010), which can hardly scale up well for high dimensional problems. To tackle these difficulties, we will propose a practical implementation to learn a parametric ACNF via ascent regularization in the next section.

Before we are able to compute the ascent regularization, we need to look into the score function, \( \nabla \log p_t (z(t)) \), in eq. (6). Like the change of variable theorem, we propose the instantaneous change of the score function, \( \nabla \log p_t (z(t)) \) as:

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

\[
\frac{d\nabla \log p_t (z(t))}{dt} = -\nabla \log p_t (z(t)) \frac{\partial f(z(t), t)}{\partial z(t)} - \nabla \cdot (\nabla f(z(t), t)).
\]

**Proof.** See Appendix A.5 for detailed derivation.

We introduce ascent regularization in two important learning cases: maximum likelihood learning for data modeling and density estimation in Section 4, minimizing reverse KL divergence for learning annealed sampler for unbiased sampling in Section 5.

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

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 encourage certain behaviors of flows via regularization in training, we propose ascent regularization to learn parametric ACNFs which penalizes the difference
As ACNF can also define a flow from a base distribution to a target distribution, it can learn a
where ACNF with ascent regularization
Algorithm 1 Maximum likelihood learning of ACNF with ascent regularization

Require: Data samples \( X = \{x_i\}_{i=1}^{N} \),
parametric dynamics of flow \( f_\theta \), length of flow \( T \), ascent regularization coefficient \( \lambda \), base distribution \( \mu \).

Initialize \( \theta \)
while \( \theta \) is not converged do
Sample a mini-batch of data \( x^i \sim X \)
Integrate augmented states [\( z^i(t), \log p_t(z^i(t)) \)] forward with initial value \( x^i, 0 \) from 0 to \( T \)
Integrate augmented states [\( z^i(T), \nabla \log p_T(z^i(T)) \)] backwards with initial value \( z^i(T), \nabla \log \mu(z^i(T)) \) from \( T \) to 0
Compute loss function \( L \) in eq.(9) and \( \nabla_\theta L \) by adjoint sensitivity method
Update \( \theta \) by gradient descent algorithm
end while

between the parametric dynamics \( f_\theta \) and the steepest dynamics as derived in eq.(9). Therefore, the
total objective for maximum likelihood learning on data becomes:

\[
\min_{\theta} L = \frac{1}{N} \sum_{i=1}^{N} \left( -\log \tilde{p}_T(x^i; \theta) + \lambda \int_0^T \left( \nabla \log p_t(z^i(t); \theta) - \nabla \log \mu(z^i(t)) \right) + f(z^i(t), t; \theta) \right) dt,
\]

(9)

where \( \lambda \) is an 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 \( z(t) \). The ascent regularization, however, is not easy due to the gradient \( \nabla \log p_t(z(t)) \). As mentioned earlier, we assume normalized distribution at \( T \) is close to the base distribution, i.e. \( p_T(z(T)) \approx \mu(z(T)) \), thus \( \nabla \log p_t(z(t)) \) can be integrated backwards from initial \( \nabla \log \mu(z(T)) \) which, in practical implementation, is augmented with states \( z(t) \), having a initial value \( z(T) \) from forward integration. We summarize pseudo-code for maximum likelihood learning of ACNFs in Algorithm 1.

5 LEARNING ACNF AS ANNEALED SAMPLER FOR UNBIASED SAMPLING

Except modeling data samples and performing density estimation, NF as a sampler has been found
advantageous for Annealed Importance Sampling (AIS) \((\text{Neal}, 2001)\) in terms of sample efficiency
compared to classic MCMC methods. A typical AIS or its extension relies on 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
\( \pi_K := \pi = \gamma / Z \). SNF \((\text{Wu et al., 2020})\) and AFT \((\text{Arbel et al., 2021})\) propose to fit \( K \) discrete NFs
that each approximates the transport map between \( \pi_{k-1} \) and \( \pi_k \). However, the sampling convergence
rate largely depends on the pre-defined annealed target. Besides, a larger number of steps \( K \) is needed
to decrease the variance of the estimator, but it comes at an additional cost \((\text{Doucet et al., 2022})\).

As ACNF can also define a flow from a base distribution to a target distribution, it can learn a
continuous annealed target and later easily generate samples. Different to \((\text{Grosse et al., 2013})\), the
annealed target by ACNF does not limit in the form of distribution. Since ACNF is enforced to converge
faster to the target distribution, it potentially generates better samples than vanilla CNF or
commonly used linear annealed scheduling especially at limited computations, thus the estimate on,
e.g. logarithm of normalization constant \( \log Z \), becomes more accurate.

Different to maximum likelihood learning, training ACNF for annealed sampling is to minimize the
reverse KL divergence \( KL(p_T(z(T)) || \pi(z(T))) \) and eq.(9) directly defines the dynamics for the
steepest descent of \( KL(p_t(z(t)) || \pi(z(t))) \). The objective can be equivalently estimated up to a
constant by the logarithm of importance weights, \( \log w(z^i(T)) = \log \gamma(z^i(T)) - \log p_T(z^i(T)) \),

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

Require: target distribution \( \pi = \gamma / Z \),
parametric dynamics of flow \( f_\theta \), length of flow \( T \), number of samples \( N \), ascent regularization coefficient \( \lambda \), base distribution \( \mu \).

Initialize \( \theta \)
while \( \theta \) is not converged do
Sample \( z^i_0 \sim p_0 = \mu \)
Evaluate \( \log \mu(z^i_0) \) and \( \nabla \log \mu(z^i_0) \)
Integrate augmented states [\( z^i(T), \log p_T(z^i(T)) \), \( \nabla \log p_T(z^i(T)) \)] with initial value \( z^i_0, \log p_t(z^i_0), \nabla \log \mu(z^i_0) \) from 0 to \( T \)
Evaluate \( \log w(z^i(T)) = \log \gamma(z^i(T)) - \log p_T(z^i(T)) \)
Compute loss function \( L \) in eq.(10) and \( \nabla_\theta L \)
by adjoint sensitivity method
Update \( \theta \) by gradient descent algorithm
end while
where \( z'(T) \) is samples \( z'_i \sim \mu \) after applying the flow and \( \log p_T(z'(T)) = \log \mu(z'_0) - \int_0^T \nabla \cdot f(z(t), t) dt \). Therefore, with ascent regularization, the total objective becomes:

\[
\min_{f} \mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left( -\log w(z'(T); \theta) + \lambda \int_0^T \| \nabla \log p_t(z'(t); \theta) - \nabla \log \mu(z'(t)) \|_2^2 dt \right),
\]

(10)

where \( f_\theta \) here is the annealed generation dynamics. Different from previous section, \( z'(t) \), \( \log p_t(z'(t)) \) and \( \nabla \log p_t(z'(t)) \) can be integrated simultaneously. We summarize pseudo-code for learning ACNF annealed sampler in Algorithm 2. Once ACNF is learned, it can generate unbiased samples: generate one-shot samples from ACNF with integration length \( T \) according to computation budget; correct samples by resampling via importance weights like (Müller et al., 2019) or Markov Chain Monte Carlo methods like Metropolis-Hastings. It is also possible to extend the one-shot sampler to SNF (Wu et al., 2020) or AFT (Arbel et al., 2021) with the correction in between the discrete steps of flows to alleviate the topological constraint of diffeomorphism.

6 EXPERIMENTS

6.1 DENSITY ESTIMATION ON TOY 2D DISTRIBUTIONS

Before we deploy ACNF for modeling complex distributions, we first train ACNF to model data from a 2-modal Gaussian mixture with a standard Gaussian as the base distribution in 2D. Figure 4 shows that the potential fields of learned ACNF is very similar to the PDE solutions while CNF potential converges much slower than ACNF and the PDE solution, and then diverges. See Appendix A.6 for experiment details and comparison on different regularization coefficients \( \lambda \) and flow lengths \( T \).

We train vanilla CNF, RNODE (Finlay et al., 2020) and ACNFs to model data from various 2D toy distributions and visualize the tractable density evaluation of learned flows. Figure 5 shows the densities at \( t \in [0, 2T] \) by CNF and ACNFs learned with \( T = 10 \) and different regularization coefficients for a 2-moon distribution. The densities that are close to the target distribution are highlighted inside the red border. We show that even with slight regularization, the learned flows 1) converge densities much faster towards the target; 2) maintain the best density estimations for longer length after \( T \). The quantitative evaluation is shown by the log-likelihood estimates on the left of Figure 6. More analysis on different \( T \) and experiment setups are given in Appendix A.7.

One may suspect that ACNF learns complex dynamics that contributes to the faster ascent of log-likelihoods. To examine the actual improvements by ACNF, we plot the number of function evaluations (NFEs) on the integration with and without log-likelihood augmentation for all models on the left of Figure 6 and log-likelihood estimate versus NFEs in the middle. It is clearly shown that
ascent regularization encourages the learning of even less complex dynamics than CNF and RNODE as log-likelihood gain per NFE of ACNFs are much higher than the two baselines. Comparing different ascent regularization, a larger coefficient strengthens more rapid gain on the log-likelihood at the initial stage, however, too large regularization constrains models to have a lower maximum. A moderate regularization benefits on the balance of a good maximum of likelihood estimate and faster convergence of flows. Furthermore, we report NFEs evaluation 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. Generally, learned ACNFs have lower NFEs than that of CNF and RNODE. Most models report their lowest NFEs at $T = 1$. It indicates that optimizing $T$ as proposed by TO-FLOW (Du et al., 2022) and STEER (Ghosh et al., 2020) can help to decrease the computational cost, however, the gain is not that much as NFEs are not linear to integration length and the strategy is not as effective as ascent regularization. Besides, neither method cannot prevent the deterioration of density estimation of CNF at $t > T$. Figure 7 show more density evaluations on learning multi-modal distributions. Learned ACNFs show faster convergence than CNFs for all distributions and give 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 and ACNF trained with three different ascent regularization coefficients. The detailed description of experiments and models refers to Appendix A.8. Although FFJORD with multi-step flows increases the flexibility of flows, it tends to have a worse performance than the base distribution and ACNFs initially and improves NLL mainly at the late stage of flows. The larger ascent regularization coefficients of ACNF contributes to more rapid initial increases on NLL which shows that these flows transform the base distribution faster towards the data distribution. When training on HEPMASS and BSDS300, a too large regularization coefficient hinders model convergences.
Table 1: Averaged negative log-likelihoods (NLLs) on test data for density estimation.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Model</th>
<th>0.1T</th>
<th>0.25T</th>
<th>0.5T</th>
<th>0.75T</th>
<th>T</th>
<th>t &gt; T⁺</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWER</td>
<td>FFJORD†§</td>
<td>7.47</td>
<td>5.97</td>
<td>4.63</td>
<td>2.55</td>
<td>-0.42</td>
<td>5.02</td>
</tr>
<tr>
<td></td>
<td>RNODE</td>
<td>7.80</td>
<td>6.11</td>
<td>4.80</td>
<td>2.64</td>
<td>-0.46</td>
<td>4.59</td>
</tr>
<tr>
<td></td>
<td>ACNF, 1e⁻⁴</td>
<td>7.03</td>
<td>5.02</td>
<td>2.63</td>
<td>0.81</td>
<td>-0.40</td>
<td>0.12</td>
</tr>
<tr>
<td></td>
<td>ACNF, 1e⁻³</td>
<td>6.26</td>
<td>4.15</td>
<td>1.99</td>
<td><strong>0.48</strong></td>
<td>-0.44</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>ACNF, 1e⁻²</td>
<td><strong>5.43</strong></td>
<td><strong>3.47</strong></td>
<td>1.87</td>
<td><strong>0.48</strong></td>
<td>-0.37</td>
<td><strong>-0.26</strong></td>
</tr>
<tr>
<td>GAS</td>
<td>FFJORD†§</td>
<td>11.76</td>
<td>241.13</td>
<td>145.75</td>
<td>132.12</td>
<td>-8.60</td>
<td>279.97</td>
</tr>
<tr>
<td></td>
<td>RNODE</td>
<td>12.03</td>
<td>102.77</td>
<td>78.20</td>
<td>45.38</td>
<td>-8.79</td>
<td>90.23</td>
</tr>
<tr>
<td></td>
<td>ACNF, 1e⁻⁴</td>
<td>7.25</td>
<td>3.31</td>
<td>-1.28</td>
<td>-5.5</td>
<td>-8.56</td>
<td>-6.88</td>
</tr>
<tr>
<td></td>
<td>ACNF, 1e⁻³</td>
<td>5.58</td>
<td>1.65</td>
<td>-1.73</td>
<td>-5.61</td>
<td>-8.45</td>
<td>-7.63</td>
</tr>
<tr>
<td></td>
<td>ACNF, 1e⁻²</td>
<td><strong>3.57</strong></td>
<td>-0.14</td>
<td>-3.73</td>
<td>-6.42</td>
<td>-8.57</td>
<td><strong>-8.33</strong></td>
</tr>
<tr>
<td>HEPMASS</td>
<td>FFJORD†§</td>
<td>35.51</td>
<td>40.19</td>
<td>82.08</td>
<td>106.81</td>
<td>14.90</td>
<td>42.09</td>
</tr>
<tr>
<td></td>
<td>RNODE</td>
<td>35.02</td>
<td>40.59</td>
<td>68.91</td>
<td>96.47</td>
<td>15.34</td>
<td>39.09</td>
</tr>
<tr>
<td></td>
<td>ACNF, 1e⁻⁴</td>
<td>27.75</td>
<td>24.88</td>
<td>20.63</td>
<td>17.17</td>
<td>14.95</td>
<td><strong>15.36</strong></td>
</tr>
<tr>
<td></td>
<td>ACNF, 1e⁻³</td>
<td><strong>25.24</strong></td>
<td><strong>22.78</strong></td>
<td>18.81</td>
<td><strong>15.94</strong></td>
<td>14.88</td>
<td>15.52</td>
</tr>
<tr>
<td>MINIBOONE</td>
<td>FFJORD†§</td>
<td>58.34</td>
<td>33.98</td>
<td>42.12</td>
<td>24.55</td>
<td>10.90</td>
<td>16.25</td>
</tr>
<tr>
<td></td>
<td>RNODE</td>
<td>57.98</td>
<td>53.65</td>
<td>41.79</td>
<td>24.12</td>
<td>10.77</td>
<td>15.86</td>
</tr>
<tr>
<td></td>
<td>ACNF, 1e⁻⁴</td>
<td>54.15</td>
<td>43.76</td>
<td>29.29</td>
<td>19.00</td>
<td>10.63</td>
<td>13.95</td>
</tr>
<tr>
<td></td>
<td>ACNF, 1e⁻³</td>
<td>52.64</td>
<td>42.87</td>
<td>28.92</td>
<td>18.72</td>
<td>10.40</td>
<td>12.57</td>
</tr>
<tr>
<td></td>
<td>ACNF, 1e⁻²</td>
<td><strong>51.52</strong></td>
<td><strong>40.52</strong></td>
<td>26.53</td>
<td><strong>17.16</strong></td>
<td>10.95</td>
<td><strong>12.14</strong></td>
</tr>
<tr>
<td>BSDS300</td>
<td>FFJORD†§</td>
<td>41.72</td>
<td>30.11</td>
<td>21.36</td>
<td>-85.92</td>
<td>-136.69</td>
<td>-72.21</td>
</tr>
<tr>
<td></td>
<td>RNODE</td>
<td>42.38</td>
<td>33.29</td>
<td>1.74</td>
<td>-84.03</td>
<td>-156.71</td>
<td>-89.35</td>
</tr>
<tr>
<td></td>
<td>ACNF, 1e⁻⁴</td>
<td><strong>35.24</strong></td>
<td>24.10</td>
<td>-14.75</td>
<td>-100.35</td>
<td>-156.0</td>
<td><strong>-125.67</strong></td>
</tr>
<tr>
<td></td>
<td>ACNF, 1e⁻³</td>
<td><strong>37.45</strong></td>
<td><strong>21.53</strong></td>
<td><strong>-17.84</strong></td>
<td><strong>-105.11</strong></td>
<td>-156.5</td>
<td><strong>-120.89</strong></td>
</tr>
</tbody>
</table>

† 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 to 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, 1.17T 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 7 for generated sample comparisons.

6.3 ACNF as a Faster Annealing Sampling Proposal for Unbiased Sampling

Following Algorithm 2 we train CNF and ACNFs with three regularization coefficients λ = 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 is chosen as a Gaussian $\mathcal{N}(0, 3^2I)$ to have adequate support for the target distribution. Figure 8 compares the estimates of log Z on different flow lengths and reports NFES versus estimates. 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 regularized ACNF converges faster towards the target distribution than CNF, the one-shot samples generated from ACNF are less biased than those from CNF especially at early stage of the flows and ACNFs are more computationally efficient in terms of accuracy gain per NFES as shown. Compared to the best tuned linear annealed target, ACNFs with coefficients $\lambda = 0.01, 0.001$ have more accurate estimates. Besides, due to the slow mixing in Metropolis sampler, the linear annealed target requires at least 1 order more computations than CNF and ACNFs for comparable accuracy, and is sensitive to the number of MC steps. Therefore, ACNF can prevent slow sample convergence from a suboptimal annealed target and make estimates more accurate with less computation. Figure 7 in Appendix A.9
Table 2: Averaged negative ELBO on MNIST datasets under different length of flows $t$.

<table>
<thead>
<tr>
<th>Model</th>
<th>0.1$T$</th>
<th>0.25$T$</th>
<th>0.5$T$</th>
<th>0.75$T$</th>
<th>$T$</th>
<th>1.2$T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE-FFJORD</td>
<td>85.90</td>
<td>85.07</td>
<td>83.96</td>
<td>83.26</td>
<td>82.88(82.82)</td>
<td>89.74</td>
</tr>
<tr>
<td>VAE-ACNF, $1e^{-4}$</td>
<td>85.62</td>
<td>84.60</td>
<td>83.45</td>
<td>83.06</td>
<td><strong>82.74</strong></td>
<td>85.67</td>
</tr>
<tr>
<td>VAE-ACNF, $1e^{-3}$</td>
<td><strong>84.70</strong></td>
<td><strong>83.95</strong></td>
<td><strong>83.22</strong></td>
<td><strong>82.53</strong></td>
<td>82.80</td>
<td><strong>84.37</strong></td>
</tr>
</tbody>
</table>

† originally reported in FFJORD

shows the corresponding generated samples by all methods. Adding MC steps in ACNFs flows can further accelerate sample convergence to target and increase the expressiveness of ACNF.

6.4 VARIATIONAL INFERENCE WITH ACNFs

In addition to density estimations and unbiased sampling, CNFs provides more flexible variational approximations to improve variational inference (Rezende and Mohamed, 2015). We follow the 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 VAE-ACNF model, the log weights in eq.(10) is replaced by ELBO estimate as (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.10. The averaged negative ELBO of vanilla VAE is 86.50. Table 2 reports the averaged negative ELBO on test data for VAE-FFJORD and VAE-ACNFs with two regularization $\lambda = 1e^{-4}, 1e^{-3}$ evaluated at various flow lengths. VAE-ACNFs show the rapid 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 also circumvent the ELBO deterioration of VAE-FFJORD thanks to the self-stabilization behavior of ACNF. We also show some reconstruction examples from VAE-ACNF in Figure 18 in Appendix A.10. These reconstructions tend to correct some defects from original images, add details to avoid confusion on classifications and remain sharp as the original images.

7 SCOPES AND LIMITATIONS

While we have demonstrated that ascent regularization is effective to learn a flow that converges faster to a target distribution, our proposed method has a number of limitations, which we would like to address in future work. First, during training, the integration of the score function needs to evaluate the Jacobian matrix and the gradient of divergence, which requires more efficient implementations, e.g. more computationally efficient estimators or dynamics model design, to scale up for high-dimensional problems. Second, Hypernet (Ha et al., 2016) is found effective to illustrate faster convergence behavior of ACNF as time in the input exerts a large impact on the dynamics, however, it is slower to train than other simpler network architecture. Better architecture design for dynamics model may substantially improve the training speed while maintaining faster convergence behavior of flows. Finally, although the experiments demonstrate in learning static distributions, proposed ACNF and ascent regularization can be extended for a sequence of distributions and applied on the inference of sequential data, e.g. computationally efficient ACNF may benefit for online inference problems.

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 parametereric 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.7–A.10. Furthermore, we attach source codes in supplementary material for further checkup.
REFERENCES


