Semi-Autoregressive Energy Flows:
Exploring Likelihood-Free Training of Normalizing Flows

Phillip Si [1 2]   **Zeyi Chen** [3]   **Subham Sekhar Sahoo** [1]   **Yair Schiff** [4]   **Volodymyr Kuleshov** [1 4]

Abstract

Training normalizing flow generative models can be challenging due to the need to calculate computationally expensive determinants of Jacobians. This paper studies the likelihood-free training of flows and proposes the energy objective, an alternative sample-based loss based on proper scoring rules. The energy objective is determinant-free and supports flexible model architectures that are not easily compatible with maximum likelihood training, including semi-autoregressive energy flows, a novel model family that interpolates between fully autoregressive and non-autoregressive models. Energy flows feature competitive sample quality, posterior inference, and generation speed relative to likelihood-based flows; this performance is decorrelated from the quality of log-likelihood estimates, which are generally very poor. Our findings question the use of maximum likelihood as an objective or a metric, and contribute to a scientific study of its role in generative modeling.

1. Introduction

Normalizing flows form one of the major families of probabilistic and generative models (Rezende & Mohamed, 2015; Kingma et al., 2016; Papamakarios et al., 2019). They feature tractable inference and maximum likelihood learning and have applications in areas, such as image generation (Kingma & Dhariwal, 2018), anomaly detection (Nalisnick et al., 2019), and density estimation (Papamakarios et al., 2017). However, training flows requires calculating computationally expensive determinants of Jacobians; this

constrains the range of architectures that can be used to parameterize flow models.

This paper questions the use of maximum likelihood for training normalizing flows and shows the existence of alternative objectives that do not require computing determinants or performing adversarial training (Grover et al., 2018). These objectives yield highly performant models; however, their performance is entirely decorrelated from competitive log-likelihood scores, which are very poor. Our findings contribute to the scientific understanding of generative model objectives and further question the common practice of training and evaluating models via the likelihood, hinting at the viability of alternative methods.

Specifically, we introduce the *energy objective*, a sample-based multidimensional extension of proper scoring rules that does not require computing model densities (Gneiting & Raftery, 2007a). The energy objective extends the recent autoregressive quantile flow framework of Si et al. (2022) to flexible non-autoregressive flow architectures. We complement this objective with efficient estimators based on random projections and a theoretical analysis that draws connections to divergence minimization, and that highlights benefits over maximum likelihood.

The energy objective enables training model architectures that are more flexible than ones trained using maximum likelihood (e.g., densely connected networks). These models feature exact posterior inference (which is useful in applications such as semi-supervised learning and latent space manipulation), as well as exact likelihood estimation (which helps us study log-likelihood as an objective and a metric). In particular, we propose semi-autoregressive flows, an architecture that interpolates between fully autoregressive and non-autoregressive models. From a practical perspective, energy flows achieve improved sample quality, posterior inference, and generation speed across a number of tasks and compared to equivalent model architectures trained using maximum likelihood. Table 1 compares our approach to existing methods.

Contributions. In summary, this work (1) presents new results that question the use of maximum likelihood for training flows and proposes an alternative approach based

---

[1]Computer and Information Science, Cornell University, Ithaca, NY, USA [2]Machine Learning Department, Carnegie-Mellon University, Pittsburgh, PA, USA [3]Tsinghua University, Beijing, China [4]Department of Computer Science, Cornell Tech, NYC, NY, USA. Correspondence to: Phillip Si <ps789@cs.cornell.edu>, Volodymyr Kuleshov <kuleshov@cornell.edu>.

on proper scoring rules and two-sample tests that extends quantile flows (Si et al., 2022) to multiple dimensions. We (2) introduce specific two-sample objectives, such as the energy loss, and derive efficient slice-based estimators. We also (3) provide a theoretical analysis for the proposed objectives as they are consistent estimators and feature unbiased gradients. Finally, we (4) introduce a semi-autoregressive architecture with high speed and sample quality on generation and posterior inference tasks. [1]

2. Background

Normalizing Flow Models Generative modeling involves specifying a probabilistic model $p(\mathbf{y}) \in \Delta(\mathbb{R}^d)$ over a high-dimensional $\mathbf{y} \in \mathbb{R}^d$ (Kingma & Welling, 2014; Goodfellow et al., 2014). A normalizing flow is a generative model $p(\mathbf{y})$ defined via an invertible mapping $f : \mathbb{R}^d \to \mathbb{R}^d$ between a noise variable $\mathbf{z} \in \mathbb{R}^d$ sampled from a prior $\mathbf{z} \sim p(\mathbf{z})$ and the target variable $\mathbf{y}$ (Rezende & Mohamed, 2015; Papamakarios et al., 2019). We may obtain an analytical expression for the likelihood $p(\mathbf{y})$ via the change of variables formula $p(\mathbf{y}) = \left| \frac{\partial f(\mathbf{z})^{-1}}{\partial \mathbf{z}} \right| p(\mathbf{z})$, where $\left| \frac{\partial f(\mathbf{z})^{-1}}{\partial \mathbf{z}} \right|$ denotes the determinant of the inverse Jacobian of f. Computing this quantity is often expensive, hence we typically choose f to be in a class of models for which the determinant is tractable (Rezende & Mohamed, 2015), as in autoregressive models (Papamakarios et al., 2017).

Proper Scoring Rules Consider a score or a loss $\ell : \Delta(\mathbb{R}^d) \times \mathbb{R}^d \to \mathbb{R}_+$ over a probabilistic forecast $F \in \Delta(\mathbb{R}^d)$ and a sample $\mathbf{y} \in \mathbb{R}^d$. The loss ℓ is proper if the true distribution $G \in \arg\min_F \mathbb{E}_{\mathbf{y} \sim G} \ell(F, \mathbf{y})$ (Gneiting & Raftery, 2007a). A popular proper loss is the continuous ranked probability score (CRPS), defined for two cumulative distribution functions (CDFs) F and G as $\text{CRPS}(F, G) = \int (F(y) - G(y))^2 \, dy$. When we only have samples from G, we can generalize this score to obtain the following loss for a single sample y': $\text{CRPS}_s(F, y') = \int_y (F(y) - \mathbb{I}(y - y'))^2 \, dy$. where $\mathbb{I}$ denotes the Heaviside step function. The above CRPS can also be written as an expectation relative to the distribution F:

$$\text{CRPS}(F, y') = -\frac{1}{2}\mathbb{E}_F|Y - Y'| + \mathbb{E}_F|Y - y'|, \quad (1)$$

where Y, Y' are independent copies of a random variable distributed according to F. Recently, Si et al. (2022) proposed autoregressive quantile flows, which are trained using the CRPS and are determinant-free. We seek to extend the approach of Si et al. (2022) beyond autoregressive flows.

Two-Sample Tests and Integral Probability Metrics Two-sample tests compare distributions F, G based on

their respective sets of samples $\mathcal{D}_F = \{\mathbf{y}^{(i)}\}_{i=1}^m$ and $\mathcal{D}_G = \{\mathbf{x}^{(i)}\}_{i=1}^n$. Specifically, a two-sample test defines a statistic $T : \mathbb{R}^d \to \mathbb{R}$, and we determine whether $\mathcal{D}_F, \mathcal{D}_G$ originate from identical or different distributions F, G based on differences in T across $\mathcal{D}_F, \mathcal{D}_G$. Two-sample tests motivate objectives for generative models such as generative moment matching networks (GMMNets; (Dziugaite et al., 2015; Li et al., 2015)) and generative adversarial networks (GANs; (Goodfellow et al., 2014)). Two-sample tests are also an attractive training objective for flows because they are density-free and therefore do not require computing determinants (Grover et al., 2018).

More modern approaches include integral probability metrics (IPMs) (Müller, 1997), which take the form $\max_{T \in \mathcal{T}} \mathbb{E}_{y \sim F}[T(y)] - \mathbb{E}_{y \sim G}[T(y)]$, where $\mathcal{T}$ is a family of functions. A special case of IPMs is maximum mean discrepancy (MMD) (Gretton et al., 2008), in which $\mathcal{T} = \{T : ||T||_{\mathcal{H}} \leq 1\}$ is the set of functions with bounded norm in a reproducing kernel Hilbert space (RKHS) with norm $|| \cdot ||_{\mathcal{H}}$; the CRPS objective can be shown to be a form of MMD (Gretton et al., 2008).

3. Exploring Determinant-Free Training of Normalizing Flows

'We propose training normalizing flows using objectives inspired by two-sample tests, which do not require computing densities. This idea poses two sets of challenges: (1) most classical two-sample tests (e.g., Kolmogorov-Smirnov) are defined in one dimension and do not have simple multivariate extensions; (2) modern two-sample tests (e.g., IPMs) extend to high dimensions, but typically require solving a costly optimization problem. Here, we derive two-sample tests that form good learning objectives, and we use the theory of proper scoring rules to justify their validity.

3.1. Sample-Based Training of Normalizing Flows and the Energy Objective

We seek to extend autoregressive quantile flows (Si et al., 2022) to general architectures without the limitations of autoregressivity (e.g., slow sampling). Specifically, we leverage a generalization of the sample-based form of the CRPS objective (1) to a multi-dimensional version called the *energy score* by Székely (2003); Gneiting & Raftery (2007a):

$$\text{CRPS}_e(F, \mathbf{y}') = -\frac{1}{2}\mathbb{E}_F||\mathbf{Y} - \mathbf{Y}'||_2^\beta + \mathbb{E}_F||\mathbf{Y} - \mathbf{y}'||_2^\beta, \quad (2)$$

where $\beta \in (0, 2)$, $|| \cdot ||_2$ denotes the Euclidean norm, and $\mathbf{Y}, \mathbf{Y}' \in \mathbb{R}^d$ are independent copies of a vector-valued random variable distributed according to F. The rightmost term $\mathbb{E}_F||\mathbf{Y} - \mathbf{y}'||_2^\beta$ promotes samples $\mathbf{Y}$ from F that are

---

Table 1: Energy Flows and Semi-Autoregressive Energy Flows (SAEFs) are invertible generative models that feature expressive architectures, exact likelihood and posterior evaluation, and their training does not require computing log-determinants, in contrast to VAEs (Kingma & Welling, 2014), MAFs (Papamakarios et al., 2017), NAFs (Huang et al., 2018), AQFs (Si et al., 2022), GMMNets (Li et al., 2015), and CramerGANs (Bellemare et al., 2017).

Method	Likelihood / Posterior	Sampling	Representation	Objective
VAE	Approx.	Feedforward	Gaussian	ELBO
MAF	Exact	Autoregressive	Gaussian	Likelihood
NAF	Exact	N/A	Non-Gaussian	Likelihood
AQF	Exact	Autoregressive	Non-Gaussian	Quantile Loss
GMMNet	N/A	Feedforward	Non-Gaussian	MMD
CramerGAN	N/A	Feedforward	Non-Gaussian	Discr.+Energy
Energy Flow (Ours)	Exact	Feedforward	Non-Gaussian	Energy Loss
SAEF (Ours)	Exact	Autoregressive	Non-Gaussian	Energy Loss

close to the data point $\mathbf{y}'$; the leftmost term $\frac{1}{2}\mathbb{E}_F||\mathbf{Y}-\mathbf{Y}'||_{\mathbf{2}}^{\beta}$ encourages the model to produce diverse samples and not concentrate all probability mass on one $\mathbf{y}$.

The Kernelized Energy Objective As a generalized extension of the CRPS, the Kernelized Energy Objective extends the Euclidean norm to a kernel function:

$$\text{CRPS}_K(F,\mathbf{y}') = -\frac{1}{2}\mathbb{E}_F K(\mathbf{Y},\mathbf{Y}') + \mathbb{E}_F K(\mathbf{Y},\mathbf{y}'), \quad (3)$$

The kernelized energy loss can be shown to be a proper loss (Gneiting & Raftery, 2007a) and, thus, represents a valid training objective for a generative model. The flow objective consists of $\mathbb{E}_{\mathbf{y}'\sim\mathcal{D}}[\text{CRPS}_K(F,\mathbf{y}')]$ and reveals the connection to two-sample tests between F and $\mathcal{D}$.

Two-Sample Baselines In Appendix D.1, we define two classical statistical tests as baselines and illustrate examples of alternative methods that can be derived from our two-sample-based approach. In brief, **Hotelling's two-sample test** uses the statistic $H_2(\mathcal{D}_F,\mathcal{D}_G) = (\mathbf{m}_F - \mathbf{m}_G)^\top S^{-1}(\mathbf{m}_F-\mathbf{m}_G)$, where $\mathbf{m}_F, \mathbf{m}_G$ are sample means, S_F, S_G are sample variances and $S = (S_F + S_G)/2$. The **Fréchet distance** uses the objective $R(\mathcal{D}_F,\mathcal{D}_G) = ||\mathbf{m}_F - \mathbf{m}_G||_2^2 + \text{tr}(S_F + S_G - 2(S_F S_G)^{1/2})$, where we are using the same notation.

3.2. Theoretical Properties

Divergence Minimization When the variable $y \in \mathbb{R}$ is one-dimensional, the objective $\text{CRPS}(F,G)$ is precisely equivalent to the Cramér divergence $\ell_2^2(F,G) = \int_{-\infty}^{\infty}(F(y)-G(y))^2 dy$ between distributions F,G. Székely (2003) show that the one-dimensional version of the energy loss (2) is precisely equivalent to ℓ_2^2. The kernelized version is a valid divergence between distributions, which directly follows from the fact that it is a proper loss (Gneiting & Raftery, 2007a). The connection to divergence minimization lends additional support to using (2) and (3) as principled

objectives—if G is the data distribution, minimizing (2) or (3) over a space of models produces an F that is close to G.

Unbiased Gradient Estimation Our objectives have the property that given a sequence $\mathbf{Y}_n$ of n samples $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n$ from G, the gradient of the empirical distribution over these samples yields an unbiased estimate of the gradient of the expected loss: $\nabla_\theta\mathbb{E}_{\mathbf{Y}_n}\ell(F_\theta,\hat{G}_n) = \nabla_\theta\ell(F_\theta,G)$, where ℓ is one of our objective functions, $\hat{G}_n$ is the empirical distribution over $\mathbf{Y}_n$, and F_θ is a model with parameters θ that we are optimizing. The statement above follows directly from the fact that both the energy and the CRPS objectives are proper scoring rules (Bellemare et al., 2017).

Why Energy Objectives? Consider the set of objectives $\ell_p^p(F,G) = \int_{-\infty}^{\infty}(F(y)-G(y))^p dy$ for $p \geq 1$ over $y \in \mathbb{R}$; these are also known as Wasserstein p-metrics (Kantorovich, 1960). The energy objective corresponds to ℓ_2^2, and it is *the only ℓ_p^p objective to support unbiased gradients* (Bellemare et al., 2017). In high dimensions, IPMs are general-purpose two-sample tests; popular IPMs include the Kantorovich metric (Kantorovich & Rubinstein, 1958), Fortet-Mourier metric (Fortet & Mourier, 1953), the Lipschitz (or Dudley) metric (Dudley, 1966), and the total variation distance. In general, IPMs are defined in terms of a potentially costly optimization problem; out of the aforementioned IPMs, only the energy objective has a known analytical (optimization-free) solution, and it also features a faster statistical convergence rate (Sriperumbudur et al., 2009).

Overall, we summarize the above facts as part of the following formal result:

Theorem 1. *The energy objectives (2) and (3) are consistent estimators for the data distribution and feature unbiased gradients.*

This follows from properties of proper scoring rules and MMD; see Appendix B for the full proof.

3.3. Scaling Sample-Based Flow Objectives Using Projections

The framework of IPMs provides a wide range of high-dimensional sample-based objectives (Müller, 1997). However, most of these objectives involve costly optimization problems, with the energy loss being a rare exception. At the same time, there exist many popular one-dimensional two-sample tests that have appealing statistical and computational properties and can yield training objectives.

We propose further improving our objective via *random projections*, specifically *slicing*, which projects data into one dimension (Kolouri et al., 2019; Song et al., 2019; Nguyen et al., 2020). Formally, we define a sampling probability $p(\mathbf{v})$ over one-dimensional vectors $\mathbf{v} \in \mathbb{R}^d$. We define a sliced version of a one-dimensional loss function $L_p(x, y) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ as

$$L_p(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{\mathbf{v} \sim p(\mathbf{v})} \left[L(\mathbf{v}^\top \mathbf{x}, \mathbf{v}^\top \mathbf{y}) \right]. \quad (4)$$

We approximate the expectation with Monte-Carlo samples.

Sliced Energy Objectives The sliced energy objective applies (1) to the projected data. In practice, we find that the number of slices needed for good performance is lower than the dimensionality of the data, resulting in a favorable computational profile. Furthermore, we can formally prove that the resulting objective has appealing statistical properties.

Theorem 2. *The sliced versions of the energy objectives (2) and (3) are consistent estimators for the data distribution and feature unbiased gradients.*

Intuitively, the first part of the theorem is true because the CRPS objective is related to the MMD. At the same time, for each $\mathbf{v}$ the objective remains a proper score; a weighted combination of proper scores is also a proper score, hence the second part holds. See Appendix B for the full proof. Recall also that in one dimension, the energy loss reduces to the CRPS, which is equivalent to the Wasserstein-2 distance. Wasserstein distances have more favorable convergence properties (Arjovsky et al., 2017) than maximum likelihood training, which lends further support to our choice of objective.

Sliced Two-Sample Baselines Slicing also allows us to use univariate two-sample tests as objectives. We describe several objectives in Appendix D.2. In brief, these include: **Kolmogorov-Smirnov**, a popular statistical test defined for two CDFs F and G as $\mathrm{KS}(F, G) = \sup_y |F(y) - G(y)|$; **Hotelling's** t^2 test $H_u(\mathcal{D}_F, \mathcal{D}_G) = \frac{(m_F - m_G)^2}{s^2}$, a sliced version of Hotelling's objective; the sliced version of the **Fréchet** objective $R_u(\mathcal{D}_F, \mathcal{D}_G) = (m_F - m_G)^2 + (s_F^2 - s_G^2)^2$.

4. Energy Flows

Next, we introduce *energy flows*, a class of models trained with our proposed determinant-free objectives. An energy flow is defined by an invertible mapping between $\mathbf{z}$ and $\mathbf{y}$ and is trained using the energy loss. As a result, energy flows improve over classical flow models by, among other things, supporting flexible architectures and by simultaneously providing fast training and sampling.

Previous work on normalizing flows involved constrained architectures with tractable determinants, such as autoregressive models. In contrast, our model supports flexible feedforward architectures, which we outline in Appendix E. In brief, our loss supports **dense invertible flows** (DIFs), sequences of fully connected layers constrained to be invertible, **invertible residual networks** (Behrmann et al., 2019), as well as **rectangular flows** (REFs), in which the dimensionality of $\mathbf{y}$ and $\mathbf{z}$ is not equal (Nielsen et al., 2020; Cunningham & Fiterau, 2021; Caterini et al., 2021).

4.1. Motivation for Energy Flows

Energy flows possess the following useful features: (1) exact posterior inference; (2) fast feed-forward generation; (3) a stable and effective training objective; (4) implicitly defined distributions (i.e., $p(x \mid z)$ is not assumed to be of any parametric form); (5) flexibility in terms of architecture for parameterizing the model. Closest in terms of features to energy flows are Generative Moment Matching Networks (GMMNets (Dziugaite et al., 2015; Li et al., 2015))—our work can be seen as introducing a principled way of doing posterior inference in GMMNets. Also related are VAEs; however they do not possess property (4), and in our experiments produce worse samples. Classical flows do not possess (5). GANs and autoregressive also differ in terms of training stability and generation speed, respectively.

It may appear that energy flows do not retain a key benefit of normalizing flows—being able to compare models via their log-likelihood. We argue that model comparison using log-likelihood may not be a good idea to being with; also, we see energy flows as a tool for the scientific study of the limitations of log-likelihood as a metric. Moreover, energy flows retain other aforementioned benefits over classical flows and other models: exact posterior inference, fast sampling via flexible architectures, and improved generation and latent space quality.

Grover et al. (2018) introduced adversarial training of flows and showed that it yields poor log-likelihoods; we propose alternative objectives that are more stable while retaining competitive sample quality. Our results contribute to the above line of work and further question the use of the likelihood for training flows. See Appendix G for more

details.

4.2. Semi-Autoregressive Flows

We also introduce semi-autoregressive flows (SAEF; pronounced "safe"), an architecture trained with the energy loss that combines the speed of feed-forward architectures with the sample quality of autoregressive models. The SAEF model divides d-dimensional data into B blocks and generates samples blockwise. As a result, sampling time is reduced by a factor of $O(d/B)$ relative to autoregressive models.

Formally, SAEFs define an invertible mapping between a latent variable $\mathbf{z}$ and an observed variable $\mathbf{y}$ and require choosing a partition of $\mathbf{y}, \mathbf{z}$ into B ordered blocks $(\mathbf{y}_b)_{b=1}^B$ and $(\mathbf{z}_b)_{b=1}^B$ (e.g., 4x4 blocks of pixels in an image). They induce a probabilistic model $p(\mathbf{y})$ over $\mathbf{y}$ that factorizes as $p(\mathbf{y}) = \prod_{b=1}^B p(\mathbf{y}_b|\mathbf{y}_{<b})$, where each $p(\mathbf{y}_b|\mathbf{y}_{<b})$ is defined via an invertible mapping

$$\mathbf{y}_b = \tau(\mathbf{z}_b; \mathbf{h}_b) \qquad \mathbf{h}_b = c_b(\mathbf{y}_{<b}), \qquad (5)$$

where $\tau(\mathbf{z}_b; \mathbf{h}_b)$ is an invertible transformer mapping the b-th latent block $\mathbf{z}_b$ to the b-th observed block $\mathbf{y}_b$, and c_b is the b-th conditioner, which outputs transformer parameters $\mathbf{h}_b$. Any invertible feed-forward energy flow can be used to parameterize τ—we provide specific examples below. The entire SAEF is trained via a sum of energy losses applied to each block $\mathbb{E}_{\mathbf{y}\sim\mathcal{D}} \left[\sum_{b=1}^B \ell(F_{\mathbf{y}_b}, \mathbf{y}_b) \right]$, where $\mathcal{D}$ is a training set, $\mathbf{y} \sim \mathcal{D}$ is a datapoint sampled from $\mathcal{D}$, $F_{\mathbf{y}_b}$ is the distribution over $\mathbf{y}_b$ induced by $\tau(\cdot, \mathbf{h}_b(\mathbf{y}_{<b}))$, and ℓ is one of our two-sample losses, such as (2) or (3).

When blocks are one-dimensional, this reduces to a standard autoregressive architecture that features high sampling quality but slow sampling speed. When blocks are full-dimensional, this reduces to a non-autoregressive energy flow with fast sampling but possibly worse quality. In our experiments, we show that SAEFs can trade-off between these two regimes and obtain the best of both worlds. Note also that SAEFs are hard to train using maximum likelihood, as they require specifying invertible non-autoregressive mappings τ between possibly high-dimensional blocks $\mathbf{y}_b, \mathbf{z}_b$—**the SAEF architecture is only trainable using the energy objective**. See Appendix F for pseudocode.

5. Experiments

We evaluate our framework on a range of UCI datasets (Dua & Graff, 2017), datasets of handwritten digits (Pedregosa et al., 2011; Deng, 2012), and real world images, such as CIFAR10 (Krizhevsky et al., 2009) and Celeb-A (Liu et al., 2015).

Baselines We benchmark our models against normalizing flows trained using either maximum likelihood or quantile loss (Si et al., 2022) and parameterized by either autoregressive or non-autoregressive architectures. Our autoregressive models are based on baselines from earlier work (Papamakarios et al., 2017; Si et al., 2022) and include Autoregressive Quantile Flows (AQF) and Masked Autoregressive Flows (MAF-LL). These models assume a parameterization $p(\mathbf{y}|\mathbf{z}) = \prod_{j=1}^d p(y_j|y_{<j}, z_j)$, where each $p(y_j|y_{<j})$ is a probability conditioned on the previous variables and z_j. In MAFs, the $p(y_j|y_{<j})$ are Gaussian; in AQFs they are parameterized by a flexible quantile flow (Si et al., 2022). We also compared to models trained with a Jacobian-free objective. Specifically, we train autoregressive MAF and AQF models with the quantile loss (Si et al., 2022) and denote these as MAF-QL and AQF-QL, respectively.

Our non-autoregressive baselines consist of variational auto-encoders (VAEs) trained using the evidence lower bound (ELBO) on the maximum likelihood and based on a fully-connected architecture (see below for details). In order to understand the benefits of our objective, we fit a VAE model with the same invertible architecture for the decoder as the one used by our energy flow models; we refer to the resulting method as a dense invertible flow trained using maximum log-likelihood (DIF-LL). We additionally compare against two state-of-the-art flow models, FFJORD (Grathwohl et al., 2018) and Invertible Resnets (Behrmann et al., 2019), using their respective open-source codebases.

Energy Flow Models We constructed flow models using dense invertible layers, referring to the resulting model as a Dense Invertible Flow trained with an energy loss (DIF-E). The DIF-E model consists of three feedforward invertible layers and Leaky ReLU activation functions and is trained using the kernelized energy loss (3) with a mixture of RBF kernels with bandwidth in $\{2, 5, 10, 20, 40, 80\}$. We also use the energy score to train non-invertible rectangular flows trained with the energy loss (REF-E). In particular, rectangular flows are parametrized by layers of size $[d/8, d/4, d/2, d]$ as compared to DIF-E which requires all layers to be of size d, for invertibility.

Metrics We evaluate the models in terms of log-likelihood (when available and appropriate) and using variants of the CRPS metric. For VAE-type models trained using the ELBO, we report the ELBO as a lower bound on the log-likelihood. We use two CRPS-style metrics which have the following structure: the first is a sample-based version as in (2). The second, marked as univariate CRPS (U-CRPS), is the sum of one-dimensional CRPS measured for each output dimension and estimates the quality of marginal distributions (Si et al., 2022). Both versions use the ℓ_1 norm. See Appendix A.3 for more details. In our image datasets,

we are also interested in a quantitative estimate of the quality of the generated samples and their similarity to the data distribution. We therefore define a metric called the D-loss, which is measured by the accuracy of a discriminative model in determining whether an image is generated (see Appendix A.2 for details).

5.1. Understanding Sample-Based Objectives

We start with experiments that analyze the properties of the sliced energy objectives and compare them to baseline two-sample objectives on the UCI and image datasets.

Energy Objective vs. Two-Sample Baselines   We claim that the energy objective is a particularly favorable training criterion for flows. We empirically establish this fact by comparing it against the other two-sample objectives, which we see as strong baselines. We train an invertible flow model on the Miniboone UCI dataset. Complexity is written where b denotes the batch size, d denotes the dimension, and n denotes the number of projections made.

Flows trained using the energy objective achieve the best performance in Table 2, outperforming the strong baselines. In subsequent experiments, we focus on the energy objective.

Improvements in Scalability From Slicing   Next, we seek to understand the scalability improvements from slicing. We train a projection based model on MNIST using projected energy loss. The model consists of 4 dense layers of size 784 with Leaky ReLU activation functions for the first three layers and a sigmoid activation function for the last. When we calculate the loss, we take n projections into a single dimension, which we denote in the top half of Table 3 as n for the projection parameters. We additionally conduct slicing experiments on the SAEFs, in particular the 7x7 block size variant. Slicing is conducted separately for each block in a similar manner to the fully-feedforward flow, so we are projecting a 49-dimensional block down to a single dimension. The number of projections per block is denoted in Table 3 as Block-n. In general, even for the block parameterization, CRPS is stable, even when using fewer projections.

We see in Table 3 that sliced objectives perform comparably to non-sliced objectives while having improved computational complexity.

On Likelihood vs. Non-Likelihood Based Losses   Our work questions the use of maximum likelihood for training flow models. In Table 4, we observe that a Glow model (Kingma & Dhariwal, 2018) trained with likelihood has low CRPS, and conversely poor likelihoods when trained with an energy loss, though samples have high quality for both
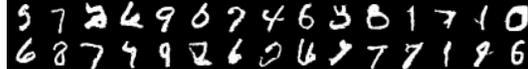
(a) FFJORD (64)
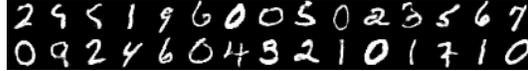
(b) SAEF-4

Figure 1: MNIST samples

setups (see Appendix H, Figure 4 for sample generations).

5.2. UCI Experiments: Use of CRPS as a Metric

In the previous sections, we primarily use CRPS as an evaluation metric for the performance of models trained with different losses. In this section, we compare CRPS to other evaluation metrics. Specifically, we implement energy flows on the Miniboone and Hepmass UCI datasets, which have been used previously as benchmarks by Papamakarios et al. (2017) and Si et al. (2022) We use an LSTM architecture (Hochreiter & Schmidhuber, 1997) for all autoregressive models. In Table 5, we report Frechet Distance, MMD, and CRPS and find that they are strongly correlated. See Appendix L for CRPS-based evaluation on an expanded set of UCI datasets.

We note that training and evaluating with the same metric is not uncommon: in fact, most generative models are trained and evaluated using the log-likelihood.

5.3. MNIST

On MNIST, we train autoregressive models with the PixelCNN architecture. The architecture has 3 residual blocks, and each masked convolutional layer has a kernel size of 7. Our SAEF models use the same general architecture, but adapted for the corresponding block sizes and trained using the block energy loss. The PixelCNN (Oord et al., 2016) maps from a base distribution (either a normal distribution, PixelMAF, or a uniform distribution, PixelAQF-QL) to the target distribution. SAEF models utilize the same architecture but with generation sectioned off into different blocks. In addition, we train a rectangular flow model using the ELBO approximation of the log-likelihood (REF-LL), resulting in a model equivalent to a VAE. Results are shown in Table 6. Samples for a state-of-the-art flow model (FFJORD) and SAEF-4 are presented in Figure 1 (see Appendix K for more models).

Results   Though the fully feedforward (DIF-E) and sliced (DIF-E-Proj) variants demonstrate decent performance with fast training and generation speed and good CRPS, we find that introducing a degree of autoregressive modeling, improves results. In particular, SAEF-4 greatly outperforms these non-autoregressive energy-based models in terms

Table 2: CRPS and Complexity for invertible flows trained on Miniboone UCI dataset with various two-sample objectives. KS stands for Kolmogorov-Smirnov test and FD stands for Fréchet Distance.

Metrics	KS	1D Hotelling	Hotelling	1D-FD	FD	1D-Energy	Energy
CRPS	1.53	0.57	0.717	0.558	0.559	0.545	0.548
Complexity	$n \log b$	n	d^3	n	d^3	bn	bd

Table 3: Slicing on MNIST.

n	400	200	100	50
U-CRPS	0.088	0.088	0.088	0.091
CRPS	0.191	0.191	0.192	0.195
Block-n	100	20	10	5
U-CRPS	0.084	0.085	0.084	0.084
CRPS	0.086	0.087	0.086	0.087

Table 4: Glow on MNIST.

Models	NLL	CRPS
Glow-LL	2.57	0.197
Glow-Energy	7.03	0.190

of FID. In addition, SAEF-4 provides an order of magnitude sampling time speedup compared to some fully autoregressive models.

Inversion and Interpolation. A key feature of the DIF-E model is exact posterior inference (despite not being trained with log-likelihood). To demonstrate this, we create intermediate representations between pairs of MNIST samples through which we can smoothly interpolate (Figure 2). Inverting the decoder model, by inverting the activation functions and weight matrices, allows us to create an encoder, similar to that of the VAE. Like the VAE, the energy flow can generate interpolated samples, with the added advantage of exact posterior inference.

Second, we examine the utility of latent space of DIF-E by training a logistic regression model to predict the class of each datapoint based on its latent representation **z**. We split

Table 5: Various Metrics on the UCI Datasets

Miniboone			
Method	Frechet Distance	MMD	CRPS
MAF-LL	3.97	1.392	0.561
MAF-QL	3.14	1.408	0.567
DIF-E	1.62	0.088	0.524
Hepmass			
Method	Frechet Distance	MMD	CRPS
MAF-LL	0.532	1.235	0.617
MAF-QL	0.501	1.179	0.614
DIF-E	0.274	0.072	0.58
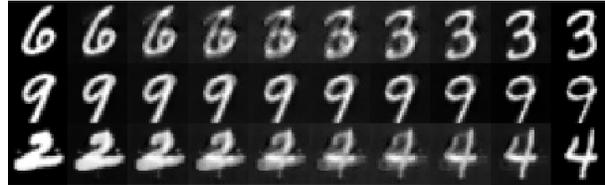
Figure 2: Interpolated MNIST samples with Energy Flow

the original 10,000 test samples into 8,000 training data and 2,000 evaluation data for the logistic model. Compared to a standard LL-trained Glow model's latent representations which produce 84.7% accuracy, our energy-trained Glow model yields improved accuracy of 88.3%.

5.4. CIFAR-10

We modify a PixelCNN model (Oord et al., 2016) to (a) use the energy objective and (b) use a semi-autoregressive architecture. In Table 7, we see that Energy-based training of a standard PixelCNN yields equal or slightly better sample quality. Further modifying the architecture yields improvements in sampling speeds at a cost of a relatively small reduction in image quality. Samples are depicted in Figure 3 (with additional samples in Appendix I).

5.5. Celeb-A

We also include an experiment on the Celeb-A dataset where we follow the same methodology as in our CIFAR-10 experiment. We can see in Table 8 that switching to the energy loss for the PixelCNN considerably improves FID. Even the 2x2 block version shows marked improvements in generation quality, with a much faster sampling time (75 seconds compared to 540 seconds for 10k samples) and improves over other baselines. For comparison, we also include Glow (Kingma & Dhariwal, 2018) and VAE (Kingma & Welling, 2013) baselines trained with LL, with FID values obtained from Xie et al. (2022).

6. Discussion and Related Work

Comparison to Other Generative Model Families We provide a complete comparison to other models in Table 1. Our approach contrasts against VAE style models (Kingma & Welling, 2014) by providing exact inference and likelihood evaluation. Unlike MAFs (Papamakarios

Table 6: MNIST Generation Experiments

Method	U-CRPS	CRPS	D-Loss	FID	MMD	Training (sec)	Sampling (sec)
PixelMAF-LL	.128	.279	1.00	100.55	0.296	35	195.38
PixelMAF-QL	.099	.215	.983	85.08	0.287	35	195.38
PixelAQF-QL	.119	.228	.986	61.27	0.232	29	195.38
REF-LL (VAE)	.090	.189	.903	44.55	0.140	8	0.25
DIF-LL (VAE)	.089	.190	.855	36.91	0.131	10	0.48
FFJORD (16)	.101	.208	.650	24.78	0.103	540	48.88
FFJORD (64)	.102	.209	.633	9.69	0.087	3100	155.69
iResNet	.100	.206	.642	41.47	0.111	840	2.43
Flow-GAN	.085	.187	0.608	43.67	0.068	15	0.40
GLOW	.090	.197	0.983	63.06	0.600	1400	190.63
REF-E	.085	.187	.778	41.04	0.052	3	0.21
DIF-E	**.084**	**.186**	.701	22.76	**0.051**	6	0.40
DIF-E-Proj	.085	.186	.819	22.55	0.056	3	0.40
SAEF-2	.085	.188	.675	9.86	0.167	32	31.22
SAEF-4	.085	.187	**.567**	**7.05**	0.081	12	8.19
SAEF-7	.085	.187	.608	14.91	0.088	6	2.17
SAEF-14	.085	.187	.650	19.57	0.068	5	0.93

Table 7: CIFAR Generation Experiments

Method	FID	Sampling (sec)
PixelCNN (Oord et al., 2016)	65.93	489
PixelCNN-Energy-1x	63.95	489
PixelCNN-Energy-2x	74.51	312
PixelCNN-Energy-4x	81.33	98

Table 8: Celeb-A Generation Experiments

Method	FID
VAE	38.76
Glow	23.32
PixelCNN	36.17
PixelCNN-Energy-1x	16.23
PixelCNN-Energy-4x	18.73

et al., 2017), our models are neither fully autoregressive in nature, nor do they make any Gaussianity assumptions. They instead use a fully neural parameterization of the output probabilities, which contributes to improved performance and modeling flexibility. Approaches like AQF (Si et al., 2022) and NAF (Huang et al., 2018) also provide neural approximators but use fully autoregressive architectures, resulting in slow sampling. Closely related feedforward models include GMMNets (Li et al., 2015; Dziugaite et al., 2015) and CramerGANs (Bellemare et al., 2017), but they do not offer likelihood evaluation and posterior inference. Our framework is also related to the Maximum Mean Discrepancy (MMD) (Gretton et al., 2008) and its

generalizations ((Li et al., 2015), (Dziugaite et al., 2015)). However, unlike models optimizing MMD (Li et al., 2015; Dziugaite et al., 2015), ours provide latent variable inference and exact density evaluation using the change of variables formula.

We also draw comparisons to self-normalizing flows (Keller et al., 2021) and flow matching (Lipman et al., 2022). Self-normalizing flows (Keller et al., 2021) achieve a similar goal to ours via two neural networks, parameterizing the forward and backward directions separately. Unlike our work, their method yields an approximate inverse (albeit one that works well in practice) and relies on a training objective that could be more susceptible to local optima, e.g., if the two neural networks are imperfect inverses of each other. Their method has the advantage of providing good scalability and modularity, as flow components can be composed to arbitrary depth and the training objective decomposes per step, while in our framework all components are trained together, which might not scale as well with depth. Flow matching (Lipman et al., 2022) is a concurrent work based on continuous normalizing flows. It can be seen as an extension of diffusion models to more general corruption processes. Thus, it is a determinant-free approach to flows, similar to stochastic differential equation-based formulations of diffusion models. It inherits the advantages of diffusion models, such as high-quality samples and good log-likelihoods; disadvantages include slow sampling speeds.

(a) PixelCNN-Energy-1x

(b) PixelCNN-Energy-2x

(c) PixelCNN-Energy-4x

(d) Gated PixelCNN

Figure 3: CIFAR samples

Comparison to Other Architectures Coupling layers are an invertible architecture that is an alternative to autoregressive flows. A coupling layer partitions the input into two blocks and keeps one block the same while defining the output of the second block as an invertible transformation conditioned on the first block. Our semi-autoregressive architecture sits between the coupling layers and autoregressive flows: it makes larger modifications to the input than a coupling layer, but smaller ones than an autoregressive layer. Crucially, our semi-autoregressive layer has significantly fewer restrictions on its parametric form than a classical coupling layer: the latter typically implements a scale-and-shift operation (e.g., in Glow (Kingma & Dhariwal, 2018), RealNVP (Dinh et al., 2017)), while our semi-autoregressive layer admits much more expressive transformations when it is trained with the energy loss.

The Likelihood as an Objective and Metric Our work explores the benefits and limitations of the likelihood as an objective and metric for generative models. We proposed energy flows, a model trained with an alternative sample-based objective, but whose likelihood is still tractable. We found that energy flows yield worse likelihoods while being faster and producing better samples (as measured by FID). This adds to an existing body of work on the decorrelation between likelihood and sample quality (Grover et al., 2018), and extends it to non-adversarial objectives. We visualized samples from a number of energy flows; our qualitative examination suggests that

mode collapse plays a role in poor likelihood scores at least in some settings (e.g., Figure 3). Thus, while our work provides a more stable likelihood-free objective than that of Grover et al. (2018), it also suggests that mode collapse is caused by the fact that an objective is sample-based, not the fact that it is adversarial. Other sample-based models may thus also suffer from mode collapse (Dziugaite et al., 2015; Li et al., 2015).

Our findings add further evidence to the notion that if one cares about sample quality, the likelihood is not the ideal objective or metric; however there may be a price to pay for high quality samples, such as diversity. The right tradeoff between quality and diversity is a choice that needs to be made by the user; our work adds tools for making this choice. More broadly, our work suggests the potential for further research into alternative training objectives for generative models; these can include regularization, even using likelihood as in Grover et al. (2018).

7. Conclusion

In this work, we proposed training normalizing flows using the energy objective, a proper scoring rule that does not require computing determinants during training. We then proposed semi-autoregressive energy flows, which feature fast sampling and high sample quality. Using an invertible flow architecture also allows us to retain exact posterior inference in energy flows. We see our work as questioning the use of likelihood for training normalizing flows, and a a further step in exploring determinant-free flows based on novel learning objectives and architectures.

Acknowledgements

References

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan, 2017. URL https://arxiv.org/abs/1701.07875.

Behrmann, J., Grathwohl, W., Chen, R. T. Q., Duvenaud, D., and Jacobsen, J.-H. Invertible residual networks, 2019.

Bellemare, M. G., Danihelka, I., Dabney, W., Mohamed, S., Lakshminarayanan, B., Hoyer, S., and Munos, R. The cramer distance as a solution to biased wasserstein gradients, 2017.

Caterini, A. L., Loaiza-Ganem, G., Pleiss, G., and Cunningham, J. P. Rectangular flows for manifold learning, 2021.

Chollet, F. et al. Keras, 2015. URL https://github.com/fchollet/keras.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Cunningham, E. and Fiterau, M. A change of variables method for rectangular matrix-vector products. In Banerjee, A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2755–2763. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/cunningham21a.html.

Deng, L. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp, 2017.

Dolatabadi, H. M., Erfani, S., and Leckie, C. Invertible generative modeling using linear rational splines, 2020.

Dua, D. and Graff, C. Uci machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Dudley, R. M. Weak convergence of probabilities on nonseparable metric spaces and empirical measures on Euclidean spaces. *Illinois Journal of Mathematics*, 10(1):109 – 126, 1966. doi: 10.1215/ijm/1256055206. URL https://doi.org/10.1215/ijm/1256055206.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows, 2019.

Dziugaite, G. K., Roy, D. M., and Ghahramani, Z. Training generative neural networks via maximum mean discrepancy optimization, 2015.

Fortet, R. and Mourier, E. Convergence de la répartition empirique vers la répartition théorique. *Annales scientifiques de l'École Normale Supérieure*, 3e série, 70(3):267–285, 1953. doi: 10.24033/asens.1013. URL http://www.numdam.org/articles/10.24033/asens.1013/.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007a.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007b.

Gneiting, T., Balabdaoui, F., and Raftery, A. E. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69 (2):243–268, 2007.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., and Duvenaud, D. Ffjord: Free-form continuous dynamics for scalable reversible generative models, 2018. URL https://arxiv.org/abs/1810.01367.

Gretton, A., Borgwardt, K., Rasch, M. J., Scholkopf, B., and Smola, A. J. A kernel method for the two-sample problem, 2008.

Grover, A., Dhar, M., and Ermon, S. Flow-gan: Combining maximum likelihood and adversarial learning in generative models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. doi: 10.1609/aaai.v32i1.11829. URL https://ojs.aaai.org/index.php/AAAI/article/view/11829.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. Neural autoregressive flows. In *International Conference on Machine Learning*, pp. 2078–2087. PMLR, 2018.

Kantorovich, L. and Rubinstein, G. S. On a space of totally additive functions. *Vestnik Leningrad. Univ*, 13:52–59, 1958.

Kantorovich, L. V. Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422, 1960.

Keller, T. A., Peters, J. W., Jaini, P., Hoogeboom, E., Forré, P., and Welling, M. Self normalizing flows. In *International Conference on Machine Learning*, pp. 5378–5387. PMLR, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes, 2013.

Kingma, D. P. and Welling, M. Stochastic gradient vb and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR*, 2014.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29:4743–4751, 2016.

Kolouri, S., Nadjahi, K., Simsekli, U., Badeau, R., and Rohde, G. K. Generalized sliced wasserstein distances. *CoRR*, abs/1902.00434, 2019. URL http://arxiv.org/abs/1902.00434.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Li, Y., Swersky, K., and Zemel, R. Generative moment matching networks, 2015.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Müller, A. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997. ISSN 00018678. URL http://www.jstor.org/stable/1428011.

Müller, T., McWilliams, B., Rousselle, F., Gross, M., and Novák, J. Neural importance sampling, 2019.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Hybrid models with deep and invertible features, 2019.

Nguyen, K., Ho, N., Pham, T., and Bui, H. Distributional sliced-wasserstein and applications to generative modeling, 2020. URL https://arxiv.org/abs/2002.07367.

Nielsen, D., Jaini, P., Hoogeboom, E., Winther, O., and Welling, M. Survae flows: Surjections to bridge the gap between vaes and flows. *Advances in Neural Information Processing Systems*, 33:12685–12696, 2020.

Oord, A. v. d., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., and Kavukcuoglu, K. Conditional image generation with pixelcnn decoders, 2016. URL https://arxiv.org/abs/1606.05328.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. *arXiv preprint arXiv:1705.07057*, 2017.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows, 2015.

Sejdinovic, D., Sriperumbudur, B., Gretton, A., and Fukumizu, K. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The Annals of Statistics*, pp. 2263–2291, 2013.

Si, P., Kuleshov, V., and Bishop, A. Autoregressive quantile flows for predictive uncertainty estimation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=z1-I6rOKv1S.

Song, Y., Garg, S., Shi, J., and Ermon, S. Sliced score matching: A scalable approach to density and score estimation, 2019. URL https://arxiv.org/abs/1905.07088.

Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Lanckriet, G. R. G., and Schölkopf, B. A note on integral probability metrics and ϕ-divergences. *CoRR*, abs/0901.2698, 2009. URL http://arxiv.org/abs/0901.2698.

Székely, G. J. E-statistics: The energy of statistical samples. *Bowling Green State University, Department of Mathematics and Statistics Technical Report*, 3(05):1–18, 2003.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio, 2016.

Wehenkel, A. and Louppe, G. Unconstrained monotonic neural networks, 2021.

Xie, J., Zhu, Y., Li, J., and Li, P. A tale of two flows: Cooperative learning of langevin flow and normalizing flow toward energy-based model, 2022.

A. Additional Experimental Details

A.1. Architecture Details

Abbreviations and Losses   In Table 9, we provide a list of abbreviations, where the upper half consists of the abbreviation for the model names, and the bottom half consists of the abbreviations for losses.

Throughout the text we use loss abbreviations appended to model abbreviations to indicate which the objective used for was training a particular model (e.g., PixelMAF-LL corresponds to a PixelMAF model trained with log likelihood). For the remaining baseline models which do not follow this convention (FFJORD, GLOW, and iResNet, in Table 6) we note that they are trained with log-likelihood, and note that Flow-GAN uses an adversarial loss combined with a log likelihood term.

Table 9: Abbreviations for models (upper) and losses (lower) used throughout the text.

Abbreviation	Full Name
Models	
AQF	Autoregressive Quantile Flow
DIF	Dense Invertible Flow
MAF	Masked Autoregressive Flow
REF	Rectangular Flow
SAEF	Semi-Autoregressive Energy Flow
iResNet	Invertible Residual Network
Pixel	PixelCNN
Losses	
E	Energy Loss
LL	Log Likelihood
QL	Quantile Loss

Slicing   For our slicing experiments with the various multidimensional and single-dimension losses, the invertible feedforward model consists of 4 layers of size 43 (to match Miniboone dimension), and each objective is trained for 200 epochs with a learning rate of 1e-3. Each sample is projected onto 200 random normal vectors, and the resulting two-sample loss is summed across the 200 projections.

UCI Experiments   For the MAF and AQF models, the LSTM architectures are composed of two LSTM layers with hidden size equal to the dimensionality of the data. All models were trained with a batch size of 200 and learning rate of $1e^{-3}$ using the ADAM optimizer (Kingma & Ba, 2014) for 200 epochs for the smaller datasets (Miniboone, Hepmass) and 20 epochs for the larger ones (Gas, Power, BSDS 300).

Image Generation Experiments   For the PixelCNN architecture, we used a receptive field of 7 for digits and a receptive field of 15 for MNIST. We chose to switch the autoregressive architectures here from LSTM-based models used in the UCI experiments, which are more sequential, to the convolutional PixelCNN architecture to account for spatial location of features. The PixelMAF-LL and PixelMAF-QL autoregressively transform pixels from the image into a distribution of the next pixel, except that PixelMAF-LL uses log likelihood loss while PixelMAF-QL uses quantile loss. The PixelAQF-QL model takes in samples from a uniform distribution, and predicts the corresponding quantile loss for each pixel.

Our DIF models are parametrized with a layer size of 64 on the digits dataset, and 784 on MNIST, with invertible activation functions (Leaky ReLU activations up until the last layer and Sigmoid for the final activation). The images are flattened prior to being passed through the feedforward model. On MNIST, each model was trained for 300 epochs with a learning rate of $1e^{-3}$, while on digits, each model was trained for 2,000 epochs.

The additional baselines are constructed as follows: each model is adapted from the code from their original papers. FFJORD models use two blocks of stacked CNF layers composed of ODE Nets. We tested with both a hidden size of 64 and a hidden size of 16, and trained for a total of 100 epochs with a learning rate of $1e^{-3}$. The Invertible ResNet (Behrmann et al., 2019) has three scale blocks, each having 32 Invertible ResNet blocks, consisting of 32 filters with three convolution types with an

ELU activation function (Clevert et al., 2015). It used an AdaGrad optimizer with a batch size of 128 trained for 70 epochs at a learning rate of $3\mathrm{e}^{-3}$. Glow was trained with 3 levels and a depth of 1 on 8 GPUs for 250 epochs.

A.2. D-Loss

The D-Loss is derived from an ensemble of SVM discriminators (with RBF kernels having a bandwidth parameter γ of 0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000, and 10000) used to differentiate the real data versus the sampled data from the model. The D-Loss for a single discriminator model $D : \mathbb{R}^n \to \{0, 1\}$ is measured as the accuracy with which D can discern the difference between generated samples and true samples, which are labeled 0 and 1, respectively. Given validation set with m data points and labels $Y \in \{0, 1\}$, we have that

$$\text{D-Loss}_\gamma = \frac{1}{m} \sum_i^m D_\gamma(x^{(i)}) y^{(i)} + (1 - D_\gamma(x^{(i)}))(1 - y^{(i)})$$

The final D-Loss is measured by the validation accuracy achieved by the best SVM, which is gotten through a 80:20 train-validation split on the 300 real and 300 fake images: $\text{D-Loss} = \max_\gamma \text{D-Loss}_\gamma$

A.3. Defining CRPS and U-CRPS

For simulated samples $\mathbf{x_1}, ..., \mathbf{x_m}$ and true data $\mathbf{y}$, we define CRPS as

$$\text{CRPS}((\mathbf{x_1}, ..., \mathbf{x_m}), \mathbf{y}) = \frac{1}{m} \sum_{i=1}^m ||\mathbf{x_i} - \mathbf{y}||^2 - \frac{1}{2m^2} \sum_{i=1}^m \sum_{j=1}^m ||\mathbf{x_i} - \mathbf{x_j}||^2$$

and U-CRPS as

$$\text{U-CRPS}((x_1, ..., x_m), y) = \frac{1}{m} \sum_{i=1}^m |\mathbf{x_i} - \mathbf{y}| - \frac{1}{2m^2} \sum_{i=1}^m \sum_{j=1}^m |\mathbf{x_i} - \mathbf{x_j}|$$

B. Theoretical Analysis and Proofs

Multi-Variate Objectives   In this section, we make the assumption that the kernels K used to define our objectives are measurable and bounded by κ. Under these conditions, when using a kernelized objective with kernel K, each distribution F can be represented by a mean embedding μ_F in the reproducing kernel Hilbert space (RKHS) induced by K (Gretton et al., 2008). We also assume that there exists a unique mapping between μ_F and F for every F in the class of model distributions $\mathcal{F}$. Note that this can be satisfied for any Borel probability measure if the kernel K is chosen to be universal or characteristic (Gretton et al., 2008). Alternatively, we may satisfy this claim by choosing our $\mathcal{F}$ to be a restricted set of distributions for which the above claim is true.

Theorem. *The energy objectives (2) and (3) are consistent estimators for the data distribution and feature unbiased gradients.*

Proof. First, we seek to establish the consistency of the minimizer of our objectives as an estimator of the data distribution. Our argument uses the fact that the (kernelized) energy score is closely connected to the maximum mean discrepancy (MMD; (Gretton et al., 2008)). Observe that

$$\mathbf{E}_{y' \sim \mathcal{D}} \text{CRPS}_K(F, \mathbf{y}') = \frac{1}{2} \mathbb{E}_F K(\mathbf{Y}, \mathbf{Y}') - \mathbb{E}_{F, \mathcal{D}} K(\mathbf{Y}, \mathbf{y}')$$

$$= (\frac{1}{2} \mathbb{E}_F K(\mathbf{Y}, \mathbf{Y}') - \mathbb{E}_{F, \mathcal{D}} K(\mathbf{Y}, \mathbf{y}') - \frac{1}{2} \mathbb{E}_{y, y' \sim \mathcal{D}} K(\mathbf{y}, \mathbf{y}')) + \frac{1}{2} \mathbb{E}_{y, y' \sim \mathcal{D}} K(\mathbf{y}, \mathbf{y}')$$

$$= -\frac{1}{2} \text{MMD}_K^2(F, \mathcal{D}) + \text{const.}$$

Hence, by maximizing the CRPS over a set of possible models F, we are minimizing a monotonic transformation of the MMD. Since (2) is a special case of (3), with the distance kernel (Sejdinovic et al., 2013), the above claim also holds for the non-kernelized energy objective (2).

We would like to establish that minimizing objectives (2) and (3) over a data distribution $\mathcal{D}_n$ of n samples from the true data distribution $\mathcal{P}$ yields a model F_n that is similar to what we would obtain if we searched for the best F using the full data distribution $\mathcal{P}$; in other words:

$$\mathbb{E}[L(F_n, \mathcal{P})] \leq \inf_{F \in \mathcal{F}} L(F, \mathcal{P}) + o(n),$$

where $L(F, \mathcal{P})$ is a metric or pseudo-metric[2] that we will instantiate shortly, $\mathcal{F}$ is the hypothesis class for the model F, F_n is the empirical risk minimization solution (from our method) over a dataset $\mathcal{D}_n$, and the additive $o(n)$ term decays to zero as we increase n. Note that if the model is well-specified (i.e., $\mathcal{P} \in \mathcal{F}$), we have $\mathbb{E}[L(F_n, \mathcal{P})] = o(n)$, and we have a consistent estimator.

To establish this fact, we will derive a version of the above identity for a modified version of the MMD, under the assumption of this section. We will also argue that the kernelized energy estimate satisfies that identity.

First, let $L(F, \mathcal{P}) = \text{MMD}(F, \mathcal{P})$. By the properties of MMD and kernels, we know that $\text{MMD}(F, \mathcal{P}) = ||\mu_F - \mu_{\mathcal{P}}||_{\mathcal{H}}$, and MMD is a pseudo-metric. Note that we have by the triangle inequality

$$L(F_n, \mathcal{P}) \leq L(F_n, \mathcal{D}_n) + L(\mathcal{D}_n, \mathcal{P}),$$

where we overload notation and use $\mathcal{D}_n$ to also denote the empirical distribution. Note that because our objective is a monotonic transformation ($\frac{1}{2}\text{MMD}^2 + \text{const}$) of the MMD, the F_n minimizes the MMD within $\mathcal{F}$. Thus we can write for any $F \in \mathcal{F}$

$$L(F_n, \mathcal{P}) \leq L(F, \mathcal{D}_n) + L(\mathcal{D}_n, \mathcal{P})$$
$$\leq L(F, \mathcal{P}) + 2L(\mathcal{D}_n, \mathcal{P})$$

where we have used once more the triangle inequality in the last line. Taking expectations on both sides and using the fact that $F \in \mathcal{F}$ was arbitrary, we find that

$$\mathbb{E}L(F_n, \mathcal{P}) \leq \inf_{F \in \mathcal{F}} L(F, \mathcal{P}) + 2\mathbb{E}L(\mathcal{D}_n, \mathcal{P})$$
$$\leq \inf_{F \in \mathcal{F}} L(F, \mathcal{P}) + 2\sqrt{\mathbb{E}L(\mathcal{D}_n, \mathcal{P})^2}.$$

To establish our claim, we need to bound the last term. Let x_i denote the i.i.d. samples from $\mathcal{D}_n$, let ϕ denote the embedding induced by the kernel K in its RKHS $\mathcal{H}$, and note that we have

$$\mathbb{E}L(\mathcal{D}_n, \mathcal{P})^2 = \mathbb{E}||\frac{1}{n}\sum_{i=1}^{n}\phi(x_i) - \mathbb{E}\phi(x)||_{\mathcal{H}}$$
$$= \text{Var}\left(\frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\right)$$
$$= \frac{1}{n}\text{Var}(\phi(x_1))$$
$$\leq \frac{2}{n}\mathbb{E}||\phi(x_1)||_{\mathcal{H}}$$
$$\leq \frac{2\kappa}{n}$$

Thus, our main claim follows with

$$\mathbb{E}L(F_n, \mathcal{P}) \leq \inf_{F \in \mathcal{F}} L(F, \mathcal{P}) + 2\sqrt{\frac{2\kappa}{n}}.$$

Thus the estimated model F_n satisfies the above inequality and if the data distribution $\mathcal{P} \in \mathcal{F}$, our consistency claim holds.

---

[2]A metric $d(x, y)$ satisfies four properties: (1) symmetry, (2) the triangle inequality, (3) $d(x, x) = 0$, (4) and $d(x, y) = 0 \iff x = y$. A pseudo-metric satisfies only the first three properties. The MMD objective is a metric if its kernel is characteristic or universal (or more generally if there is a one-to-one mapping between μ_F and F); otherwise, it is a pseudo-metric.

We can establish that the gradients are unbiased by leveraging properties of proper scoring rules. Recall from the background section that a loss $L : \Delta(\mathbb{R}^d) \times \mathbb{R}^d \to \mathbb{R}$ is strictly proper (Gneiting & Raftery, 2007a) if $G = \arg\min_F \mathbb{E}_{\mathbf{y} \sim G} L(F, \mathbf{y})$. In the context of the CRPS objective L, we have by definition of a proper loss

$$L(F, G) = \mathbb{E}_{\mathbf{y}' \sim G} \mathrm{CRPS}_K(F, \mathbf{y}') = \mathbb{E}_{\mathbf{y}' \sim G} \mathrm{CRPS}_K(F, G_{\mathbf{y}'}),$$

where $G_{\mathbf{y}'}$ is the empirical distribution derived from $\mathbf{y}'$.

Let $\mathcal{P}$ denote the true data distribution, $\mathcal{D}_n$ a dataset of size n drawn from $\mathcal{P}$, and G_n the resulting empirical distribution. Then we have:

$$
\begin{aligned}
\nabla_\theta \mathbb{E}_{\mathcal{D}_n \sim \mathcal{P}} L(F_\theta, G_n)) &= \nabla_\theta \mathbb{E}_{\mathcal{D}_n \sim \mathcal{P}} \mathbb{E}_{\mathbf{y}' \sim \mathcal{D}_n} L(F_\theta, \mathbf{y}') \\
&= \nabla_\theta \mathbb{E}_{\mathbf{y}' \sim \mathcal{P}} L(F_\theta, \mathbf{y}') \\
&= \nabla_\theta L(F_\theta, \mathcal{P}),
\end{aligned}
$$

which is equivalent to the statement that we wanted to prove.

$\square$

Alternative Approaches to Showing Consistency   The fact that consistency holds also follows from properties of the MMD for general classes $\mathcal{F}$; for example as shown in Dzuigaite et al. (Dziugaite et al., 2015) (Theorem 1),

$$\mathbb{E}[\mathrm{MMD}^2(F_n, \mathcal{P})] \leq \inf_{F \in \mathcal{F}} \mathrm{MMD}^2(F, \mathcal{P}) + o(n),$$

if $\mathcal{F}$ satisfies a fat-shattering condition. The desired consistency claim with $L(F, G)$ being our kernelized energy objective CRPS_K then follows directly from our earlier derivation by applying an affine transformation on each side of the above equation.

Alternative Approaches to Showing that Gradients are Unbiased   Note that a special case of the unbiased gradient property for the non-kernalized objective (1) has been established using techniques discussed in Bellemare et al. (2017) (Proposition 3). This result also follows from our aforementioned connection to the MMD and Lemma 6 Gretton et al. (2008).

Sliced Objectives

Theorem. *The sliced versions of the energy objectives (2) and (3) are consistent estimators for the data distribution and feature unbiased gradients.*

Proof. We establish the first part of the claim by observing that the sliced version of the energy objective

$$\mathrm{CRPS}(F, \mathbf{y}') = \mathbb{E}_{w \sim p(w)} \left[\frac{1}{2} \mathbb{E}_F K(w^\top \mathbf{Y}, w^\top \mathbf{Y}') - \mathbb{E}_F K(w^\top \mathbf{Y}, w^\top \mathbf{y}') \right], \tag{6}$$

where K is a kernel in 1D, is an affine transformation of squared MMD. Then the first part of the claim follows by the argument in Theorem 1. To see this, first, define the function $K_w : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ as

$$K_w(x, y) = K(w^\top x, w^\top y),$$

where K is the kernel used as part of the sliced energy objective. It is easy to see that $K_w(x, y)$ is a kernel. Consider any dataset $S = \{x_i\}_{i=1}^k$; then the matrix M defined as $M_{ij} = K_w(x_i, x_j)$ will be semi-definite because the corresponding matrix M' defined as $M'_{ij} = K(w^\top x_i, w^\top x_j)$ is also positive definite, because it is the kernel matrix for the set $S = \{w^\top x_i\}_{i=1}^k$. Hence, by Mercer's theorem K_w is a kernel.

Next, define the function $\bar{K} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ as

$$\bar{K}(x, y) = \mathbb{E}_{w \sim p(w)} K_w(x, y).$$

This is also a kernel, because it is a sum of kernels. Next, note that

$$\text{CRPS}(F, \mathbf{y}') = \mathbb{E}_{w \sim p(w))} \left[\frac{1}{2} \mathbb{E}_F K(w^\top \mathbf{Y}, w^\top \mathbf{Y}') - \mathbb{E}_F K(w^\top \mathbf{Y}, w^\top \mathbf{y}') \right]$$

$$= \frac{1}{2} \mathbb{E}_F \bar{K}(\mathbf{Y}, \mathbf{Y}') - \mathbb{E}_{F,\mathcal{D}} \bar{K}(\mathbf{Y}, \mathbf{y}'),$$

which is an instance of the kernelized energy objective that uses a modified kernel. Note that this is both a proper score and a rescaled version of the squared MMD with a modified kernel. The two claims of this theorem follow directly from Theorem 1.

$\square$

C. Expanded Background on Proper Scoring Rules

C.1. Predictive Uncertainty in Machine Learning

Probabilistic machine learning models predict a probability distribution over the target variable—e.g., class membership probabilities or the parameters of an exponential family distribution. We seek to produce models with accurate probabilistic outputs that are useful for generation.

Notation. Supervised models predict a target $y \in \mathcal{Y}$ from an input $x \in \mathcal{X}$. , where x, y are realizations of random variables $X, Y \sim \mathbb{P}$, and $\mathbb{P}$ is the data distribution. We are given a model $H : \mathcal{X} \to \Delta_\mathcal{Y}$, which outputs a probability distribution $F(y) : \mathcal{Y} \to [0, 1]$ within the set $\Delta_\mathcal{Y}$ of distributions over $\mathcal{Y}$; the probability density function of F is f. We are also given a training set $\mathcal{D} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^n$ consisting of i.i.d. realizations of random variables $X, Y \sim \mathbb{P}$.

C.2. Proper Scoring Rules

Comparing point estimates from supervised learning models is straightforward: we can rely on metrics such as accuracy or mean squared error. Probabilities, on the other hand, are more complex and require specialized metrics.

In statistics, the standard tool for evaluating the quality of predictive forecasts is a proper scoring rule (Gneiting et al., 2007). This paper advocates for evaluating the quality of uncertainties using proper scoring rules (Gneiting & Raftery, 2007b).

Formally, let $L : \Delta_\mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ denote a loss between a probabilistic forecast $F \in \Delta_\mathcal{Y}$ and a realized outcome $y \in \mathcal{Y}$. Given a distribution $G \in \Delta_\mathcal{Y}$ over y, we use $L(F, G)$ to denote the expected loss $L(F, G) = \mathbb{E}_{y \sim G} L(F, y)$.

We say that L is a *proper loss* if it is minimized by G when G is the true distribution for y: $L(F, G) \geq L(G, G)$ for all F. One example is the log-likelihood $L(F, y) = -\log f(y)$, where f is the probability density or probability mass function of F. Another example is the check score for $\tau \in [0, 1]$:

$$\rho_\tau(F, y) = \begin{cases} \tau(y - F^{-1}(\tau)) & \text{if } y \geq f \\ (1 - \tau)(F^{-1}(\tau) - y) & \text{otherwise.} \end{cases} \tag{7}$$

See Table 10 for additional examples.

What are the qualities of a good probabilistic prediction, as measured by a proper scoring rule? It can be shown that every proper loss decomposes into a sum of the following terms (Gneiting et al., 2007):

$$\text{proper loss} = \text{calibration} \underbrace{- \text{sharpness}}_{\text{refinement term}} + \text{irreducible term} .$$

Thus, there are precisely two qualities that define an ideal forecast: calibration and sharpness.

D. Details on Additional Two-Sample Training Objectives

D.1. Gaussian Two-Sample Baseline Objective Over High-Dimensional Vectors

We use the following classical objectives as baselines for our work and to illustrate examples of alternative methods that can be derived from our two-sample-based approach; both tests make a Gaussian modeling assumption.

Table 10: Examples of three proper losses: the log-loss, the continuous ranked probability score (CRPS), and the quantile loss. A proper loss $L(F, G)$ between distributions F, G—assumed here to be cumulative distribution functions (CDFs)—decomposes into a calibration loss term $L_c(F, Q)$ (also known as reliability) plus a refinement term $L_r(Q)$ (which itself decomposes into a sharpness and an uncertainty term). Here, $Q(y)$ denotes the CDF of $\mathbb{P}(Y = y \mid F_X = F)$, and $q(y), f(y)$ are the probability density functions of Q and F, respectively.

Proper Loss	Loss $L(F, G)$	Calibration $L_c(F, Q)$	Refinement $L_r(Q)$
Logarithmic	$\mathbb{E}_{y \sim G} \log f(y)$	$\mathrm{KL}(q \| f)$	$H(q)$
CRPS	$\mathbb{E}_{y \sim G} (F(y) - G(y))^2$	$\int_{-\infty}^{\infty} (F(y) - Q(y))^2 \mathrm{dy}$	$\int_{-\infty}^{\infty} Q(y)(1 - Q(y)) dy$
Quantile	$\mathbb{E}_{y \sim G}^{\tau \in U[0,1]} \rho_\tau(F, y)$	$\int_0^1 \int_{Q^{-1}(\tau)}^{F^{-1}(\tau)} (Q(y) - \tau) dy d\tau$	$\mathbb{E}_{y \sim Q}^{\tau \in U[0,1]} \rho_\tau(Q, y)$

Hotelling's Two-Sample Test   Being closely related to Student's t-test, it uses the following statistic:

$$H_2(\mathcal{D}_F, \mathcal{D}_G) = (\mathbf{m}_F - \mathbf{m}_G)^\top S^{-1} (\mathbf{m}_F - \mathbf{m}_G), \tag{8}$$

where $\mathbf{m}_F = \frac{1}{m} \sum_{\mathbf{y}^{(i)} \in \mathcal{D}_F} \mathbf{y}^{(i)}$, $\mathbf{m}_G = \frac{1}{m} \sum_{\mathbf{y}^{(i)} \in \mathcal{D}_G} \mathbf{y}^{(i)}$ are the sample means, and the matrices $S_F = \frac{1}{m-1} \sum_{\mathbf{y}^{(i)} \in \mathcal{D}_F} (\mathbf{y}^{(i)} - \mathbf{m}_F)(\mathbf{y}^{(i)} - \mathbf{m})^T$, $S_G = \frac{1}{m-1} \sum_{\mathbf{y}^{(i)} \in \mathcal{D}_G} (\mathbf{y}^{(i)} - \mathbf{m}_G)(\mathbf{y}^{(i)} - \mathbf{m})^T$ are sample covariances, while $S = (S_F + S_G)/2$ is their average. This objective encourages the two samples to have similar means.

Fréchet Distance   This is another Gaussian-based distance that we use as an objective:

$$\begin{aligned} R(\mathcal{D}_F, \mathcal{D}_G) = &\|\mathbf{m}_F - \mathbf{m}_G\|_2^2 \\ &+ \mathrm{tr}(S_F + S_G - 2(S_F S_G)^{1/2}), \end{aligned} \tag{9}$$

where we are using the same notation as above. This objective is encouraging the model to produce data with similar means and variances. It is derived from the Fréchet distance between two Gaussians.

D.2. Sliced Two-Sample Baselines

Slicing also allows us to use univariate two-sample tests as objectives. We give examples below.

Kolmogorov-Smirnov   One of the most popular ways of comparing the similarity between two distributions is via the quantity

$$\mathrm{KS}(F, G) = \sup_y |F(y) - G(y)|, \tag{10}$$

the maximum distance between two CDFs F and G. Empirical CDFs can be used for samples. While this test corresponds to an IPM, it does not have a widely accepted extension to higher dimensions.

Hotelling's Univariate Objective   The sliced version of Hotelling's objective corresponds to using Hotelling's t^2 univariate test (which is just the squared version of Student's t-test) as an objective.

$$H_u(\mathcal{D}_F, \mathcal{D}_G) = \frac{(m_F - m_G)^2}{s^2}, \tag{11}$$

where m_F, m_D, and s^2 are respectively the sample mean of $\mathcal{D}_F$, the sample mean of $\mathcal{D}_G$, and the combined sample variance, defined as in the multivariate version. Note that this formula is much less computationally expensive than the multi-dimensional one, which requires performing a matrix inversion (in worst-case $O(d^3)$ time), while the sliced version takes only $O(d)$ time.

Fréchet Univariate Objective   Similarly, the sliced version of the Fréchet objective is written as:

$$R_u(\mathcal{D}_F, \mathcal{D}_G) = (m_F - m_G)^2 + (s_F^2 - s_G^2)^2, \tag{12}$$

which encourages the sample means m_F, m_G and the sample variances s_F, s_G to be the same. Again, this $O(d)$ formula is less computationally expensive than in higher-dimensions, where it requires performing multiple matrix multiplications and a matrix square root ($O(d^3)$).

E. Feed-Forward Architectures for Energy Flows.

Since our objective does not require computing Jacobians, we are able to use flexible classes of invertible models that are difficult to train using log-likelihood. Previous work on invertible mapping leveraged integration-based transformers (Wehenkel & Louppe, 2021), spline approximations (Müller et al., 2019; Durkan et al., 2019; Dolatabadi et al., 2020), piece-wise separable models, and others. Our main requirement on the model architecture is efficient sampling. We describe several feed-forward flow architectures that are compatible with our objective below.

Dense Invertible Layers The simplest architecture we consider consists of a sequence of small $\mathbb{R}^d \to \mathbb{R}^d$ dense layers with invertible non-linearities (such as tanh or Leaky ReLUs). We enforce the invertibility of the dense layers by adding a scaled identity component σI_d for small $\sigma > 0$; other options for inducing invertibility include positivity constraints on the weights (Huang et al., 2018). Although the tanh non-linearities are invertible, numerical values close to $\{-1, 1\}$ tend to introduce numerical instability during inversion. We address this issue via activity regularization (Chollet et al., 2015). With these two architectural choices, for modestly sized d's, we were able to compute both $\mathbf{z} \to \mathbf{y}$ and $\mathbf{y} \to \mathbf{z}$ mappings analytically and in a numerically stable way.

Invertible Residual Networks Recently, residual networks with spectral normalization have been proposed as a flexible invertible architecture (Behrmann et al., 2019). Although one of the two directions of the flow is not computable analytically, it may be approximated using a fixed-point iteration algorithm. Invertible residual networks are typically trained using maximum likelihood; computing the determinant of the Jacobian of each layer requires a sophisticated approximation based on Taylor series expansion. Interestingly, when training these flows using maximum likelihood, the "fast" direction needs to be $\mathbf{y} \to \mathbf{z}$ in order to enable fast training, but generation becomes non-analytic. In contrast, when training using an energy objective, the $\mathbf{z} \to \mathbf{y}$ direction is fast both for training and generation.

Rectangular Flow Architectures Recently, several authors explored rectangular flows, in which the dimensionality of $\mathbf{y}$ and $\mathbf{z}$ is not equal (Nielsen et al., 2020; Cunningham & Fiterau, 2021; Caterini et al., 2021). Training with maximum likelihood involves sophisticated extensions to the change of variables formula. However, these models can be trained without modification using an energy loss as long as one can sample from them efficiently. At the same time, we may retain their pseudo-invertibility to perform posterior inference.

F. Pseudocode for Semi-Autoregressive Flows

In this section, we provide the algorithms for training (Algorithm 1) and sampling from (Algorithm 2) SAEFs.

Algorithm 1 Semi-Autoregressive Energy Flow Training

1: **Input**: $\mathbf{x} \sim \mathcal{D}$
2: **for** iteration $1, 2, \ldots$ **do**
3:     Generate $\mathbf{z}$ from random normal $(0, 1)^d$.
4:     **for** $b = 1, 2, \ldots, Blocks$ **do**
5:         $\mathbf{h}_b = c_b(\mathbf{x}_{<b})$
6:         $\mathbf{y}_b = \tau(\mathbf{z}_b; \mathbf{h}_b)$
7:         $\mathbf{y}'_b = \tau(\mathbf{z}'_b; \mathbf{h}_b)$
8:         Compute Energy Loss($\mathbf{y}_b, \mathbf{y}'_b, \mathbf{x}_b$)
9:     **end for**
10:    Backpropogate sum of energy loss on τ
11: **end for**

Algorithm 2 Semi-Autoregressive Energy Flow Sampling

1: **for** $b = 1, 2, \ldots, Blocks$ **do**
2:      $\mathbf{h}_b = c_b(\mathbf{y}_{<b})$
3:      $\mathbf{y}_b = \tau(\mathbf{z}_b; \mathbf{h}_b)$
4: **end for**
5: Return $(\mathbf{y}_1, ..., \mathbf{y}_b)$
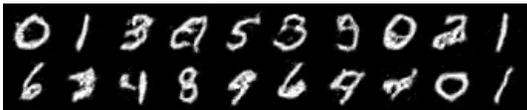
(a) Energy
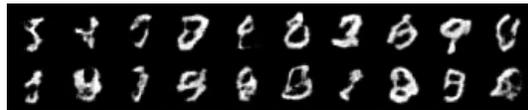
(b) LL

Figure 4: Glow samples when trained with Energy and LL loss

G. Additional Motivation for Energy Flows

From our experiments in Table 4, flows trained with the energy loss feature poor log-likelihoods. However, the FID and CRPS metrics in Table 6 are good: these models still generate very good images. We note that previous work has already shown that models trained with non-likelihood objectives tend to have very weak log-likelihoods, e.g., training normalizing flows with adversarial losses (Grover et al., 2018). Our work adds additional results to this line of work.

We believe that exact posterior inference is a particularly important feature of flows. Popular papers (RealNVP (Dinh et al., 2017), Glow (Kingma & Dhariwal, 2018)) use inferred latent $\mathbf{z}$ for interpolation, image manipulation, etc., which we have also done in our interpolation experiments. Though these are not low-dimensional, high-dimensional $\mathbf{z}$ are still useful. Many generative modeling families (GANs (Goodfellow et al., 2014), PixelCNN/WaveNet (van den Oord et al., 2016), GMMNets (Li et al., 2015)) do not have latent inference, and retaining this feature is an important advantage over many types of models.

H. Glow Samples on MNIST

Samples for a Glow model trained with either the energy or a log-likelihood objective on the MNIST dataset are presented in Figure 4. As discussed, in Section 5.1, relatively poor log-likelihood values do not translate to meaningful differences in generated sample quality.

I. Autoregressive Samples on CIFAR10

Additional autoregressive samples for CIFAR10 are added in Figure 5. The blended (PixelCNN-Energy-2x) samples feature the first half of the image generated by the fully autoregressive SAEF-1, and the second half of the image generated by SAEF-2, which allows for a 30% reduction in time compared to a fully-autoregressive model.

J. Stability Comparison of MMD and GAN loss.

We study how robust a neural network architecture is to the changes in hyperparameter when it is trained using a adversarial loss vs MMD loss. The adversarial loss is defined using a convolutional discriminator network which tries to determine whether the data is a true sample or comes from the generator. First, optimal hyperparameter settings for a generator network that features 4 fully connected layers of width 784 and Leaky ReLU activation were found: $lr = 2\mathrm{e}^{-5}$ for GAN loss and $lr = 0.0002$ for MMD. In Figure 6, we observe that removing 1 extra layer from the generator causes a significant deterioration in the sample quality for a model trained using the GAN loss. In contrast, when using MMD loss we do not see this deterioration. For both settings, we train using the optimal hyperparameters for the corresponding loss functions.

K. Additional MNIST Samples

MNIST generated samples for a broader set of comparison models are provided in Figure 7.

(a) PixelCNN-Energy-1x

(b) PixelCNN-Energy-2x

(c) PixelCNN-Energy-4x

(d) Gated PixelCNN

Figure 5: CIFAR samples
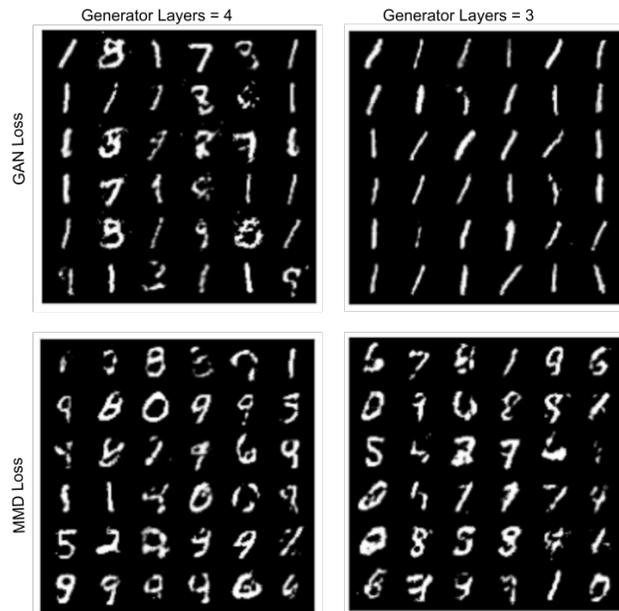
Figure 6: Stability comparision of the MMD and the GAN loss. A Generator network trained using MMD loss typically requires the same hyperparameters to train the model when subtle changes are made to the network's architecture. On the right column we see that the sample quality deteriorated for a generator that was trained using the GAN loss after removing layer and used the optimal hyperaparameter configuration for a 4 layered network.

(a) PixelMAF-LL

(b) PixelMAF-QL

(c) DIF-LL

(d) DIF-E

(e) FFJORD (64)

(f) SAEF-4
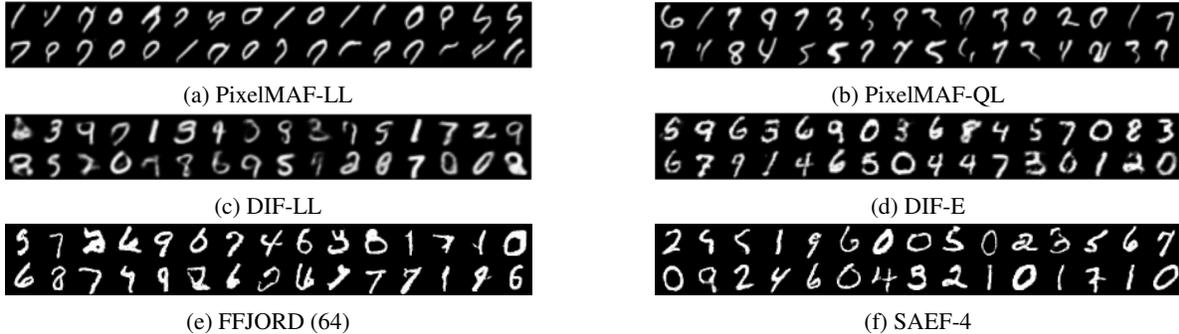
Figure 7: MNIST samples from six methods

L. Expanded Results on UCI Evaluated by CRPS

In Section 5.2 we presented results for two UCI datasets. Here, we include CRPS evaluation on an expanded set of UCI datasets: BSDS 300, Miniboone, Gas, Power, and Hepmass, which have been used previously as benchmarks by Papamakarios et al. (2017) and Si et al. (2022). The size of the datasets are noted in Table 11. The SAEFs use an appropriate block size b (listed as a column in the table) which divides the dimension of the data. Miniboone bas been padded to ensure an even divisibility of the dimension of the dataset by the dimensionality of the block. This would apply to any other dimensionality for which the appropriate block size would not divide the dimensionality of the data.

Table 11: Model performance on UCI datasets as measured by CRPS.

Dataset	d	MAF-LL	MAF-QL	AQF-QL	DIF-E	DIF-E Proj	b	SAEF
BSDS 300	63	.044	.036	**.033**	.039	.040	3	.037
Miniboone	43	.567	.561	.525	.524	.545	2	**.521**
Gas	8	.645	.565	**.513**	.548	.551	2	.530
Power	6	.542	.506	.502	.451	.454	2	**0.443**
Hepmass	21	.617	.614	**.523**	.589	.587	3	.559

Results. As shown in Table 11, energy flows perform comparably to the neural AQF-QL baseline. Both methods are trained using variants of the CRPS and obtain top performance across the five datasets. However, the DIF-E model is non-autoregressive, hence provides advantage in terms of sampling speed. Our experiment illustrate that non-autoregressive models can match the performance of autoregressive models trained with log-likelihood or versions of the CRPS objective. SAEFs, which use a variant of the same loss as DIF-E, further improve upon its results, giving slightly better CRPS scores across the board, while still being reasonable in terms of sampling speed.

M. Image Generation on Digits

We also test our methods on a small-scale generative modeling task: digit generation (Pedregosa et al., 2011), displaying results in Table 12. We use a PixelCNN architecture for the autoregressive models, which we denote PixelMAF-LL, PixelMAF-QL, and PixelAQF-QL. The PixelCNN maps from a $(\mathbb{N}(0,1))^d$ distribution (PixelMAF) and a $(\mathbb{U}(0,1))^d$ distribution (PixelAQF-QL) to the target distribution. SAEF models also utilize the same PixelCNN architecture, but with its generation sectioned off into different blocks, each of which is evaluated by our energy loss. We provide the ELBO loss as an upper bound on the NLL for VAE-type models, and training speed is given by seconds per epoch, while sampling speed is given by seconds per 1000 samples.

Results. The proposed DIF-E and SAEF models perform comparably on the U-CRPS and CRPS metrics to the AQF model, although the latter is more discriminable (has a better D-loss). On the other hand the samples generated by the DIF-E and SAEF outperforms those of any of the other autoregressive architectures, as well as the samples from non-autoregressive DIF-LL model, which is trained with maximum log likelihood.

Table 12: Digits Generation Experiments

Method	U-CRPS	CRPS	D-Loss	Training (sec)	Sampling (sec)
PixelMAF-LL	0.136	0.206	0.974	0.15	14.00
PixelMAF-QL	0.131	0.204	0.883	0.15	14.00
PixelAQF-QL	0.127	0.199	**0.681**	0.13	14.00
DIF-LL (VAE)	0.138	0.207	0.941	0.07	0.01
REF-E	0.127	0.201	0.823	0.12	0.06
DIF-E	**0.126**	**0.197**	0.807	0.14	0.07
DIF-E-Proj	0.127	0.199	0.815	0.06	0.08
SAEF-1	0.126	0.198	0.795	0.12	10.54
SAEF-2	0.126	0.199	0.754	0.07	2.73
SAEF-4	0.127	0.198	0.772	0.05	0.86