Featurized Density Ratio Estimation

Kristy Choi[1]   **Madeline Liao**[1]   **Stefano Ermon**[1]

[1]Computer Science Department, Stanford University

Abstract

Density ratio estimation serves as an important technique in the unsupervised machine learning toolbox. However, such ratios are difficult to estimate for complex, high-dimensional data, particularly when the densities of interest are sufficiently different. In our work, we propose to leverage an invertible generative model to map the two distributions into a common feature space prior to estimation. This featurization brings the densities closer together in latent space, sidestepping pathological scenarios where the learned density ratios in input space can be arbitrarily inaccurate. At the same time, the invertibility of our feature map guarantees that the ratios computed in feature space are equivalent to those in input space. Empirically, we demonstrate the efficacy of our approach in a variety of downstream tasks that require access to accurate density ratios such as mutual information estimation, targeted sampling in deep generative models, and classification with data augmentation.

1 INTRODUCTION

A central problem in unsupervised machine learning is that of density ratio estimation: given two sets of samples drawn from their respective data distributions, we desire an estimate of the ratio of their probability densities [Nguyen et al., 2007, Sugiyama et al., 2012b]. Computing this ratio gives us the ability to *compare and contrast* two distributions, and is of critical importance in settings such as out-of-distribution detection [Smola et al., 2009, Menon and Ong, 2016], mutual information estimation [Belghazi et al., 2018, Song and Ermon, 2019], importance weighting under covariate shift [Huang et al., 2006, Gretton et al., 2009, You et al., 2019], and hypothesis testing [Gretton et al., 2012]. Related areas of research which require access to accurate density ra-
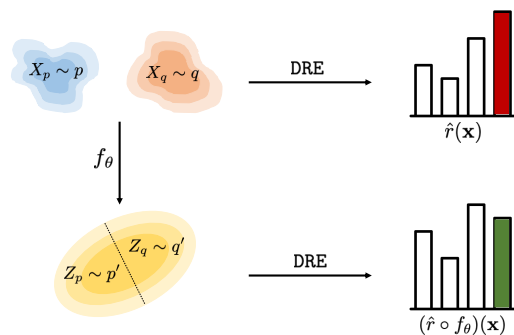

Figure 1: Flowchart for the featurized density ratio estimation framework. Direct density ratio estimation using a black-box algorithm DRE on samples leads to poor ratio estimates $\hat{r}(\mathbf{x})$ when p and q are sufficiently different. By training a normalizing flow f_θ on samples from both densities and encoding them to a shared feature space prior to estimation, we obtain more accurate ratios $(\hat{r} \circ f_\theta)(\mathbf{x})$.

tios, such as generative modeling [Gutmann and Hyvärinen, 2010, Goodfellow et al., 2014, Nowozin et al., 2016] and unsupervised representation learning [Thomas et al., 2021], have enjoyed tremendous success with the development of more sophisticated techniques for density ratio estimation.

Despite its successes, density ratio estimation is an extremely hard problem when the two distributions of interest are considerably different [Cortes et al., 2010, Yamada et al., 2013, Rhodes et al., 2020]. The fundamental challenge in reliably estimating density ratios in this scenario is precisely the access to only a finite number of samples. As the distance between the densities increases, we become less likely to observe samples that lie in low-density regions between the two distributions. Therefore, without an impractically large training set, our learning algorithm is highly likely to converge to a poor estimator of the true underlying ratio.

To address this challenge, we propose a general-purpose framework for improved density ratio estimation that brings the two distributions closer together in latent space. The

key component of our approach is an invertible generative model (normalizing flow), which is trained on a mixture of datasets drawn from the two distributions and used to map samples into a shared feature space prior to ratio estimation [Rezende and Mohamed, 2015]. Encoding the data via the normalizing flow transforms the observed samples from the two densities to lie within a unit Gaussian ball. We observe that this contraction helps mitigate pathological scenarios where the learned ratio estimates are wildly inaccurate. The invertibility of our feature map then guarantees that the ratios computed in feature space are equivalent to those in input space. We demonstrate the generality of our framework by pairing it with several existing density ratio estimation techniques, and explore various training procedures in estimation algorithms that require learning a probabilistic classifier. A flowchart of our featurized density ratio estimation algorithm can be found on Figure 1.

Empirically, we evaluate the efficacy of our approach on downstream tasks that require access to accurate density ratios. First, we demonstrate that applying our approach to existing density ratio estimation techniques on synthetic data leads to better performance on downstream domain adaptation and mutual information (MI) estimation. Next, we demonstrate the utility of our framework on a targeted generation task on MNIST [LeCun, 1998]. By leveraging the "featurized" density ratios for importance sampling from a trained generative model, we show that the resulting samples are closer to the target distribution of interest than the synthetic examples generated using input-space density ratios. Finally, we illustrate that our method can be used to improve upon naive data augmentation methods by reweighing synthetic samples, outperforming relevant baselines on multi-class classification on Omniglot [Lake et al., 2015].

The contributions of our work can be summarized as:

1. We introduce a general-purpose algorithm for estimating density ratios in feature space and show its applicability to a suite of existing ratio estimation techniques.

2. By leveraging the invertibility of our feature map, we prove that our featurized density ratio estimator inherits key properties such as unbiasedness and consistency from the original ratio estimation algorithm.

3. On downstream tasks that require access to accurate density ratios, we show that our approach outperforms relevant baselines that compute ratios in input space.

2 PRELIMINARIES

2.1 INVERTIBLE TRANSFORMATIONS VIA NORMALIZING FLOWS

Deep invertible generative models, or normalizing flows, are a family of likelihood-based models that describe the two-way transformation between a complex, continuous probability density and a simple one by the change of variables formula [Rezende and Mohamed, 2015, Papamakarios et al., 2019]. The flow is parameterized by a deep neural network $f_\theta : \mathcal{X} \to \mathcal{Z}$ with carefully designed architectures such that the overall transformation is composed of a series of bijective mappings with tractable inverses and Jacobian determinants [Dinh et al., 2016, Kingma et al., 2016, Papamakarios et al., 2017, Kingma and Dhariwal, 2018, Grathwohl et al., 2018, Ho et al., 2019]. As a result, the density of the random variable $X = f_\theta^{-1}(Z)$ can be evaluated *exactly*:

$$p(\mathbf{x}) = t(f_\theta(\mathbf{x})) \left| \det \frac{\partial f_\theta(\mathbf{x})}{\partial \mathbf{x}} \right|$$

where $f_\theta^{-1} : \mathcal{Z} \to \mathcal{X}$ denotes the inverse of the mapping f_θ, p denotes the probability density of the random variable X, and t denotes the probability density of Z. The base (prior) distribution $t(\mathbf{z})$ is typically chosen to be an isotropic Gaussian $\mathcal{N}(0, I)$ – the simplicity of evaluating this prior density, coupled with the tractability of f_θ^{-1} and its Jacobian, allows us to train the normalizing flow via maximum likelihood. The key property of normalizing flows that we exploit in our method is their *invertibility*: the dimensionality of $\mathcal{X}$ and $\mathcal{Z}$ are the same by design, and data points can be losslessly mapped between the two spaces. As we will demonstrate in Section 3.2, this will be critical for translating the density ratios obtained in latent space back to those in input space.

2.2 DENSITY RATIO ESTIMATION TECHNIQUES

Notation and Problem Setup. We denote the input variable as $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d$, and let $\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^d$ be a latent variable of the same dimensionality as the input. We use capital letters to denote random variables, e.g. $X_p \sim p$ and $X_q \sim q$. Then, $Z_p = f_\theta(X_p)$ and $Z_q = f_\theta(X_q)$ denote the random variables obtained by transforming X_p and X_q with f_θ. We note that since Z_p and Z_q are transformed by the same normalizing flow f_θ, we can form the mixture density $Z = \frac{1}{2}Z_p + \frac{1}{2}Z_q \sim t$, where $t(\mathbf{z}) \sim \mathcal{N}(0, I)$.

The learning setting we consider is as follows. Given two sets of observed samples $\mathcal{D}_p = \{\mathbf{x}_i^p\}_{i=1}^{n_p} \sim p(\mathbf{x})$ and $\mathcal{D}_q = \{\mathbf{x}_j^q\}_{j=1}^{n_q} \sim q(\mathbf{x})$, we wish to estimate the ratio of their underlying probability densities $r(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x})$. We focus on *direct* ratio estimation techniques, where we learn the density ratio estimator $\hat{r}$ rather than constructing explicit *density estimates* of $\hat{p}(\mathbf{x})$ and $\hat{q}(\mathbf{x})$ and computing their ratio [Sugiyama et al., 2012b]. The estimator $\hat{r}$ is obtained via an estimation algorithm DRE which takes as input two datasets and returns a function $\text{DRE}(\mathcal{D}_p, \mathcal{D}_q) = \hat{r} : \mathcal{X} \to \mathbb{R}$. Then, evaluating $\hat{r}$ at a particular point $\mathbf{x}$ gives us an estimate of the true density ratio $\hat{r}(\mathbf{x}) \approx r(\mathbf{x})$. In the following exposition, we provide background information on the suite of existing density ratio estimation algorithms.
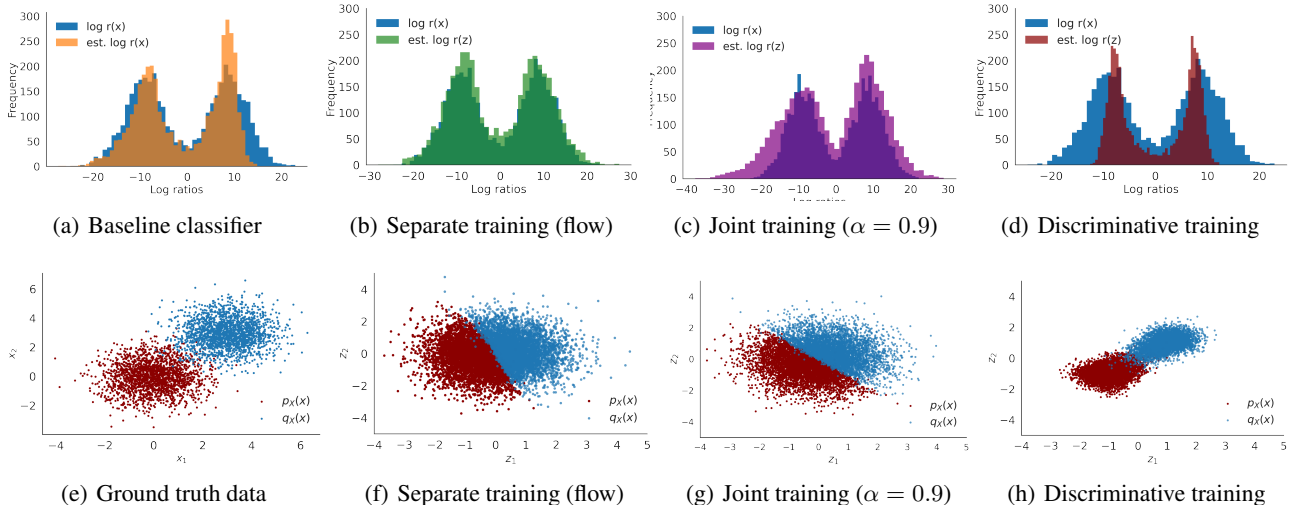
Figure 2: *Top row:* Motivating example on a synthetic 2-D Gaussian dataset, with learned density ratio estimates by method relative to the ground truth values for (a-d). *Bottom row:* Visualizations of the learned encodings for various training strategies for (f-h), with ground truth samples from $p(\mathbf{x})$ and $q(\mathbf{x})$ in (e). We note that using a pretrained flow as an invertible encoder as in (b) leads to the most accurate density ratio estimates.

Direct Ratio Estimation. From the wealth of alternative estimators for this task [Kanamori et al., 2009, Sugiyama et al., 2012a, Vapnik et al., 2013], we outline two classical methods which perform density ratio estimation that benefit from featurization as per our framework: (1) Kernel Mean Matching (KMM) [Huang et al., 2006, Gretton et al., 2009], which draws inspiration from moment matching techniques, and (2) the Kullback-Leibler Importance Estimation Procedure (KLIEP) [Nguyen et al., 2007, Sugiyama et al., 2008].

For KMM, density ratio estimates are obtained by projecting all inputs into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ induced by a characteristic kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Although several choices for the kernel are possible, Huang et al. use the Gaussian kernel $k(\mathbf{x}, \mathbf{x}') = \exp(||\mathbf{x} - \mathbf{x}'||^2)$ to arrive at the following objective:

$$\min_{\hat{r} \in \mathcal{H}} ||\mathbb{E}_{q(\mathbf{x})}[k(\mathbf{x}, \cdot)\hat{r}(\mathbf{x})] - \mathbb{E}_{p(\mathbf{x})}[k(\mathbf{x}, \cdot)]||^2_{\mathcal{H}}$$

where both expectations are approximated via Monte Carlo. Intuitively, KMM attempts to match the mean embedding of the two distributions (where the embedding is produced by the canonical feature map defined by k) in $\mathcal{H}$.

For KLIEP, the goal is to estimate density ratios $r(\mathbf{x}) = p(\mathbf{x})/q(\mathbf{x})$ such that the Kullback-Leibler (KL) divergence between $p(\mathbf{x})$ and $\hat{p}(\mathbf{x}) = \hat{r}(\mathbf{x})q(\mathbf{x})$ is minimized:

$$\min_{\hat{r}(\mathbf{x})} \mathbb{E}_{p(\mathbf{x})}\left[\log \frac{p(\mathbf{x})}{\hat{r}(\mathbf{x})q(\mathbf{x})}\right]$$

$$\text{s.t.} \int \hat{r}(\mathbf{x})q(\mathbf{x})d\mathbf{x} = 1$$

The solution to this constrained optimization problem can also be obtained in $\mathcal{H}$ by parameterizing $\hat{r}_\theta(\mathbf{x}) =$

$\sum_{i=1}^{n_p} \theta_i k(\mathbf{x}, \mathbf{x}_i^p)$ for $\mathbf{x}^p \sim p(\mathbf{x})$ and some kernel k, similar in spirit to KMM.

Probabilistic Classification. Another technique to obtain density ratio estimates is via probabilistic classification, in which a binary classifier $c_\phi : \mathcal{X} \to [0, 1]$ is trained to discriminate between samples from the two densities $p(\mathbf{x})$ and $q(\mathbf{x})$ [Friedman et al., 2001, Gutmann and Hyvärinen, 2010, Sugiyama et al., 2012b]. Concretely, suppose we construct a dataset such that all samples $\mathcal{D}_p$ are given the pseudolabel $y = 1$, and those from $\mathcal{D}_q$ are labeled as $y = 0$. Assuming that the two datasets are equal in size ($n_p = n_q$, though this can be relaxed with a scaling factor), we can use Bayes' rule to arrive at the following expression for the density ratio:

$$r(\mathbf{x}) = \frac{p(\mathbf{x})}{q(\mathbf{x})} = \frac{p(\mathbf{x}|y=1)}{q(\mathbf{x}|y=0)} = \frac{c_\phi^*(\mathbf{x})}{1 - c_\phi^*(\mathbf{x})}$$

where $c_\phi^*(\mathbf{x}) = P(y = 1|\mathbf{x})$ denotes the Bayes optimal classifier for this particular task.

3 FEATURIZED DENSITY RATIO ESTIMATION

3.1 MOTIVATING EXAMPLE AND INTUITION

Despite the suite of existing techniques for density ratio estimation, they are of limited use in settings where $p(\mathbf{x})$ and $q(\mathbf{x})$ are mismatched in support [Cortes et al., 2010, Yamada et al., 2013, You et al., 2019, Rhodes et al., 2020] We highlight an illustrative failure case in Figure 2 on a 2-dimensional toy dataset, where $p(\mathbf{x}) \sim \mathcal{N}([0, 0]^T, I)$ and

$q(\mathbf{x}) \sim \mathcal{N}([3,3]^T, I)$. As shown in Figure 2(e), the two random variables have regions of minimal overlap – when training a binary classifier c_ϕ to distinguish the two sets of samples, the log-ratio estimates $\log \hat{r}_\phi(\mathbf{x})$ learned by the classifier are noticeably inaccurate (Figure 2(a)).

To develop a solution, we consider a simple example to build intuition about the problem (with more details in Appendix B). Suppose we want to estimate the density ratios between two 1-dimensional Gaussians, $p \sim \mathcal{N}(m, 1)$ and $q \sim \mathcal{N}(-m, 1)$, with a finite number of samples $\mathcal{D}_p = \{x_i^p\}_{i=1}^n$ and $\mathcal{D}_q = \{x_i^q\}_{i=1}^n$ of size n from each. The analytic solution for $r(x) = p(x)/q(x) = \exp\left(-\frac{(x-m)^2-(x+m)^2}{2}\right) = \exp(2mx)$ for $x \in \mathbb{R}$, which grows exponentially with m. Without access to the parametric forms of p and q, we train a logistic regression model c_ϕ to discriminate between $\mathcal{D}_p$ and $\mathcal{D}_q$, where the maximum likelihood objective is:

$$\max_{w_0, w_1} \mathbb{E}_p[\log \sigma(w_0 + w_1 \cdot x)] + \mathbb{E}_q[\log \sigma(-w_0 - w_1 \cdot x)]$$

where $\sigma(z) = 1/(1 + \exp(-z))$. Although the logistic regression model is well-specified in this setting, and can achieve Bayes optimal risk in the limit of infinite data, we illustrate what can go wrong in the finite sample regime.

Suppose that $m > 0$ is large – there exists a large separation between p and q. Then, most samples $\mathcal{D}_p \sim p$ will take on positive values, and most samples $\mathcal{D}_q \sim q$ will be negative. In this situation, the model will be incentivized to push $w_1 \to \infty$ to maximize the objective. This will lead to wildly inaccurate density ratio estimates, as we know that the true values of $w_1 = 2m$ and $w_0 = 0$ are far from infinity (in fact, $r(x) = \exp(w_1 \cdot x)$). Thus we must see samples *between* p and q during training: concretely, samples from p such that $x_p \leq 0$ and samples from q such that $x_q \geq 0$. But with n samples from p, the probability that $\mathcal{D}_p$ contains all positive samples is $\prod_{i=1}^n P(X_p > 0) \geq (1 - n\exp(-m^2/2))$, which means that the number of samples required to avoid pathological solutions is exponential in m^2. This implies that density ratio estimation via probabilistic classification in input space is near impossible in such scenarios without extremely large amounts of training data.

3.2 METHODOLOGY

The motivating example in the previous section suggests that it is critical to bring p and q "closer together" to make the density ratio estimation problem tractable. Our solution is to do so in *latent space* by leveraging an invertible transformation. Concretely, we consider training an invertible deep generative model f_θ on a *mixture* of $p(\mathbf{x})$ and $q(\mathbf{x})$, such that $f_\theta(X_p)$ and $f_\theta(X_q)$ are mapped to a common feature space $\mathcal{Z}$. The result of utilizing f_θ as invertible feature map can be visualized in Figure 2(f): the flow compresses all data points to lie in different regions of a unit Gaussian ball. By mapping regions of low density in $\mathcal{X}$ into regions

of higher density in $\mathcal{Z}$, and training our probabilistic classifier c_ϕ on $f_\theta(\mathbf{x}) \in \mathcal{Z}$ rather than $\mathbf{x} \in \mathcal{X}$ directly, this contraction leads to learning more accurate density ratios as shown in Figure 2(b). We refer the reader to Appendix F for additional experimental details and results.

Our first observation is that as a direct consequence of the invertibility of f_θ, the density ratios obtained in feature space are equivalent to those obtained in input space. We formalize this statement in Lemma 1 below.

Lemma 1. *Let $X_p \sim p$ be a random variable with density p, and $X_q \sim q$ be a random variable with density q. Let f_θ be any invertible mapping. Let p', q' be the densities of $Z_p = f_\theta(X_p)$ and $Z_q = f_\theta(X_q)$ respectively. Then for any $\mathbf{x}$:*

$$\frac{p(\mathbf{x})}{q(\mathbf{x})} = \frac{p'(f_\theta(\mathbf{x}))}{q'(f_\theta(\mathbf{x}))}$$

Proof. We provide the proof in Appendix C.1. $\square$

This simple observation is quite powerful, as it lends us a general-purpose algorithm that may improve many existing ratio estimation techniques as a black-box wrapper. We provide the pseudocode for our training procedure in Algorithm 1. Given the two sets of samples, the ratio estimation method, and an invertible generative model family, we first train the normalizing flow on a mixture of the two datasets (Lines 2-3). We then use the trained flow to encode the samples into a common feature space and plug them into the base density ratio estimator algorithm DRE$(\cdot)$ to obtain our featurized density ratio estimator $\hat{r} \circ f_\theta^*$, which is implicitly composed with the trained normalizing flow (Line 6). This algorithm allows us to lightly modify existing approaches such as KMM and KLIEP as detailed in Appendix A, and we explore their featurized variants in our experiments.

Algorithm 1 Featurized Density Ratio Estimation

Input: Datasets $\mathcal{D}_p$ and $\mathcal{D}_q$, Density Ratio Estimation Algorithm DRE, Invertible Generative Model Family $\{f_\theta, \theta \in \Theta\}$

Output: Featurized Density Ratio Estimator $\hat{r} \circ f_\theta^*$

1: ▷ Phase 1: Train invertible generative model
2: Concatenate datasets $\mathcal{D} = \{\mathcal{D}_p, \mathcal{D}_q\}$
3: Train f_{θ^*} on $\mathcal{D}$ via maximum likelihood
4: ▷ Phase 2: Obtain density ratio estimator
5: $\hat{r} = \text{DRE}(f_{\theta^*}(\mathcal{D}_p), f_{\theta^*}(\mathcal{D}_q))$
6: **return** $\hat{r} \circ f_\theta^*$

3.3 TRAINING PROCEDURE

In practice, there are a variety of ways to implement the training procedure as outlined in Algorithm 1. The most

general is **separate training**, which leverages a pre-trained flow f_θ as an invertible encoder to map the inputs into a common feature space, prior to ratio estimation. This approach is capable of handling all parametric and nonparametric techniques which operate directly in input space.

In the probabilistic classification setting, where the density ratio estimation algorithm $\text{DRE}(\cdot)$ requires learning a binary classifier c_ϕ to distinguish between $\mathcal{D}_p$ and $\mathcal{D}_q$, we can *adapt* the normalizing flow f_θ to account for the known structure of c_ϕ. We call this procedure **joint training**. Both the normalizing flow f_θ and the discriminative classifier c_ϕ are trained jointly via the following objective:

$$\mathcal{L}_{\text{joint}}(\theta, \phi) = \alpha \mathcal{L}_{\text{sup}}(\theta, \phi) + (1 - \alpha)\mathcal{L}_{\text{flow}}(\theta) \quad (1)$$

where $\mathcal{L}_{\text{sup}}$ denotes the standard binary cross entropy (logistic) loss, $\mathcal{L}_{\text{flow}}$ denotes the maximum likelihood objective for the flow f_θ, and $\alpha \in [0, 1]$ is a hyperparameter which balances the importance of the two terms in the loss function. This approach is quite common in learning deep hybrid models [Kuleshov and Ermon, 2017, Nalisnick et al., 2019].

Finally, we explore **discriminative training**, where we modify the classifier c_ϕ's architecture to incorporate that of the flow f_θ to build an "invertible" classifier $c_{\phi,\theta} : \mathcal{X} \rightarrow [0, 1]$ that is trained solely via the logistic loss $\mathcal{L}_{\text{sup}}(\theta, \phi)$. This is inspired by the strong performance of invertible networks such as i-RevNet [Jacobsen et al., 2018], i-ResNet [Behrmann et al., 2019], and Mintnet [Song et al., 2019] on downstream classification tasks.

3.4 CHARACTERIZATION OF THE LEARNED FEATURE SPACE

At a first glance, Lemma 1 appears to suggest that *any* feature space induced by an invertible map should work well for density ratio estimation, as long as $f_\theta(X_p)$ and $f_\theta(X_q)$ are closer together than X_p and X_q. To gain further insight into the desirable characteristics of the learned feature space, we visualize the encodings of the various training strategies in Figure 2. For both a pretrained (Figure 2(f)) and jointly trained (Figure 2(g)) normalizing flow, the data points are mapped to lie closer together in different regions of the unit Gaussian ball. However, for the discriminatively trained classifier equipped with an invertible "encoder" (Figure 2(h)), the encoded examples more closely resemble the shape of the original inputs (Figure 2(e)). This observation, combined with the low quality density ratio estimates in Figure 2(d) relative to the other training methods (Figure 2(b-c)), suggests that maximum likelihood training of the normalizing flow f_θ *in addition to* shrinking the gap between the densities p and q is crucial for obtaining accurate density ratios in feature space. We hypothesize that mapping the observations into a unit Gaussian ball is an important property of our method, and we save an in-depth theoretical analysis of this phenomenon for future work.

4 THEORETICAL ANALYSIS

In this section, we provide theoretical justifications for several properties of the featurized density ratio estimator. As a consequence of Lemma 1, we find that our estimator inherits many of the desirable properties of the original estimator.

4.1 PROPERTIES OF THE ESTIMATOR

Unbiasedness. Unbiasedness is one of the most fundamental desiderata of a statistical estimator, as it guarantees that the estimated parameter is equivalent to the parameter's true value in expectation. In Corollary 1, we prove that unbiasedness of the featurized ratio estimator follows directly if the original estimator is also unbiased.

Corollary 1. *Let $\mathcal{D}_p$ be n i.i.d samples from density p, and $\mathcal{D}_q$ be n i.i.d samples from density q. Let $\hat{r}(\mathbf{x})$ obtained from $\hat{r} = DRE(\mathcal{D}_p, \mathcal{D}_q)$ be an unbiased estimator of $r(\mathbf{x}) = \frac{p(\mathbf{x})}{q(\mathbf{x})}$ and any p, q, and let f_θ denote any invertible mapping. Then, $(\hat{r}' \circ f_\theta)(\mathbf{x})$ obtained from $\hat{r}' = DRE(f_\theta(\mathcal{D}_p), f_\theta(\mathcal{D}_q))$ is also an unbiased estimator of $\frac{p(\mathbf{x})}{q(\mathbf{x})}$ for any p, q.*

Proof. We provide the proof in Appendix C.2. $\square$

Consistency. Consistency is another key property in a statistical estimator, as it guarantees that in the limit of infinite data used in the estimation procedure, the probability that the estimator becomes arbitrarily close to the true parameter converges to one. We prove in Corollary 2 that consistency of the featurized density ratio estimator also follows if the original density ratio estimator is consistent. This is desirable, as estimators such as the KLIEP and KMM (with universal kernels) are both consistent [Huang et al., 2006, Gretton et al., 2009, Sugiyama et al., 2012b].

Corollary 2. *Let $\mathcal{D}_p$ be n i.i.d samples from density p, and $\mathcal{D}_q$ be n i.i.d samples from density q. Let $\hat{r}(\mathbf{x})$ obtained from $\hat{r} = DRE(\mathcal{D}_p, \mathcal{D}_q)$ be a consistent estimator of $\frac{p(\mathbf{x})}{q(\mathbf{x})}$ for all $\mathbf{x} \in \mathcal{X}$ and for any p, q. Let f_θ be any invertible mapping. Then, $(\hat{r}' \circ f_\theta)(\mathbf{x})$ obtained from $\hat{r}' = DRE(f_\theta(\mathcal{D}_p), f_\theta(\mathcal{D}_q))$ is also a consistent estimator of $\frac{p(\mathbf{x})}{q(\mathbf{x})}$ for any p, q.*

Proof. We provide the proof in Appendix C.3. $\square$

5 EXPERIMENTAL RESULTS

In this section, we are interested in empirically investigating the following questions:

1. Are the density ratios learned in feature space indeed more accurate than those learned in input space?

2. Do estimates in feature space yield better performance on downstream tasks that rely on density ratios?

For conciseness, we report the average over several runs for all experiments and report complete results in Appendix F.

Datasets. We evaluate the efficacy of featurized density ratio estimation on both synthetic and real-world datasets. The synthetic experiments include toy examples on Gaussian mixtures of varying dimensionality (see Appendix F.2), as well as datasets from the UCI Machine Learning Repository [Dua and Graff, 2017]. For more challenging scenarios, we consider MNIST [LeCun, 1998] and Omniglot [Lake et al., 2015]. Additional details on the dataset construction for all experiments can be found in Appendix D.

Models. We train different classifiers depending on the difficulty of the classification task, but largely keep the same architecture (either an MLP or CNN) across different tasks. For the normalizing flow, we utilize the Masked Autoregressive Flow (MAF) for all datasets [Papamakarios et al., 2017]. We train the MAF separately on the mixture of the two datasets prior to density ratio estimation for all experiments with the exception of the MI estimation experiment in Section 5.2, where we explore various training strategies mentioned in Section 3.3. For additional details regarding architecture design and relevant hyperparameters, we refer the reader to Appendix E.

5.1 DOMAIN ADAPTATION

We first pair our method with two existing techniques, KMM and KLIEP, to assess whether estimating ratios in feature space improves performance on domain adaptation tasks with: 1) 2-D Gaussian mixtures and 2) the UCI Breast Cancer dataset. On the synthetic dataset, our method achieves a lower test error than both baseline logistic regression (without importance weighting) and reweighted logistic regression using density ratios estimated by KMM and KLIEP in input space. See Appendix F.2 for full results.

The UCI Breast Cancer dataset consists of $n = 699$ examples from 2 classes: benign ($y = 1$) and malignant ($y = -1$), where each sample is a vector of 9 features. We replicate the experimental setup of [Huang et al., 2006] to construct a source dataset with a heavily downsampled number of benign labels, while leaving the target dataset as is. After learning the importance weights via density ratio estimation on a mixture of the source and (unlabeled) target datasets, we train a support vector machine (SVM) with a Gaussian kernel of bandwidth $\sigma = 0.1$ and varying penalty hyperparameter values $C = \{0.1, 1, 10, 100\}$ with importance weighting on the source domain. The binary classifier is then tested on the target domain. As shown in Figure 3, when applied to KMM, for nearly all values of C, our method (`z-dre`) achieves the lowest test error on the target dataset compared to both a vanilla SVM (`baseline`)

and a reweighted SVM with density ratio estimates computed in input space (`x-dre`). Additionally, we note that our method achieves the absolute lowest test error across varying values of C. We report the average values of our KMM experiments over 30 runs in Figure 3.

All methods performed poorly overall for our KLIEP experiments. This result aligns with many past works with KLIEP importance-weighted classification; empirically, KLIEP only outperforms baseline unweighted classifiers on synthetic data, while on more complex datasets (e.g. UCI), KLIEP shows no significant improvements [Sugiyama et al., 2007, Tsuboi et al., 2008, Yamada and Sugiyama, 2009, Loog, 2012]. In order to confirm the consistency of this behavior, we performed an additional experiment with a slightly different dataset-biasing process in which data points that were further from the mean were selected less often, similarly to Huang et al. [2006]; we report more details on the biased subsampling process in Appendix D.2. We used two datasets: 1) the UCI Blood Transfusion dataset and 2) the UCI Wine Quality dataset and found that both reweighted classifiers performed similarly to the baseline. Notably, our `z-dre` method does not degrade the performance of KLIEP. Table 1 shows our results.
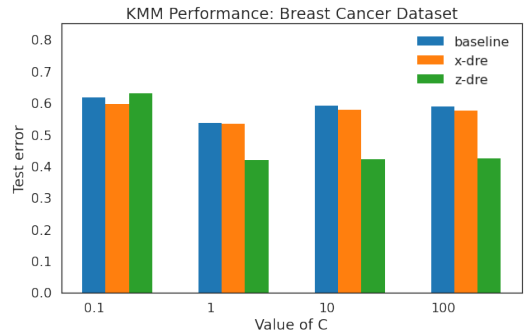

Figure 3: KMM test error on binary classification of the UCI Breast Cancer dataset using a SVM with varying C. Lower is better. Results are averaged over 30 runs.

5.2 MUTUAL INFORMATION ESTIMATION

Next, we test our approach on a mutual information (MI) estimation task between two correlated 20-dimensional Gaussian random variables, where the ground truth MI is tractable. MI estimation between two random variables X_p and X_q is a direct application of density ratio estimation, as the problem can be reduced to estimating average density ratios between their joint density and the product of their marginals. If we let v denote the joint density of X_p and X_q, we can see that: $I(X_p; X_q) = \mathbb{E}_{v(\mathbf{x}^p, \mathbf{x}^q)} \left[\log \frac{v(\mathbf{x}^p, \mathbf{x}^q)}{p(\mathbf{x}^p) q(\mathbf{x}^q)} \right]$. We adapt the experimental setup of [Belghazi et al., 2018, Poole et al., 2019, Song and Ermon, 2019] to use a correlation coefficient of $\rho = 0.9$.

Blood Transfusion	$C = 0.1$	$C = 1$	$C = 10$	$C = 100$
KLIEP with DRE in z-space (ours)	$0.235 \pm .0274$	$0.235 \pm .0274$	$0.234 \pm .0283$	$0.234 \pm .0282$
KLIEP with DRE in x-space	$0.235 \pm .0274$	$0.235 \pm .0274$	$0.234 \pm .0282$	$0.233 \pm .0284$
Unweighted SVM baseline	$0.235 \pm .0274$	$0.235 \pm .0274$	$0.234 \pm .0287$	$0.234 \pm .0285$
Wine Quality	$C = 0.1$	$C = 1$	$C = 10$	$C = 100$
KLIEP with DRE in z-space (ours)	$0.304 \pm .0120$	$0.260 \pm .00937$	$0.265 \pm .00817$	$0.290 \pm .00987$
KLIEP with DRE in x-space	$0.304 \pm .0123$	$0.262 \pm .0105$	$0.266 \pm .0113$	$0.290 \pm .0103$
Unweighted SVM baseline	$0.302 \pm .0274$	$0.257 \pm .0074$	$0.262 \pm .00863$	$0.289 \pm .0933$

Table 1: KLIEP test error of each method on binary classification for the UCI Blood Transfusion and Wine Quality datasets. Results are averaged over 30 runs. KLIEP reweighting in general does not offer significant improvement over the unweighted baseline–in particular, our method (z-space) doesn't degrade performance.

We further explore the effect of the various training strategies as outlined in Section 3.3. While we use a MAF as the normalizing flow for all configurations, we evaluate our approach against: (a) the baseline classifier (`baseline`); (b) the two-stage approach (`separate`), where the flow is trained first on a mixture of $\mathcal{D}_p$ and $\mathcal{D}_q$ before training the classifier on the encoded data points; (c) jointly training the flow and the classifier (`joint`); and (d) a purely discriminative approach where the classifier architecture has a flow component (`disc-only`). For joint training, we sweep over $\alpha = \{0.1, 0.5, 0.9\}$. As shown in Figure 4, the probabilistic classifier trained in feature space (after encoding the data using the normalizing flow) via our method outperforms relevant baselines. Interestingly, we find that for the joint training, higher values of α (which places a greater emphasis on the classification loss $\mathcal{L}_{\text{sup}}$ rather than $\mathcal{L}_{\text{flow}}$ as in Eq. 1) leads to more accurate MI estimates. For additional details on the data generation process and experimental setup, we refer the reader to Appendix E.
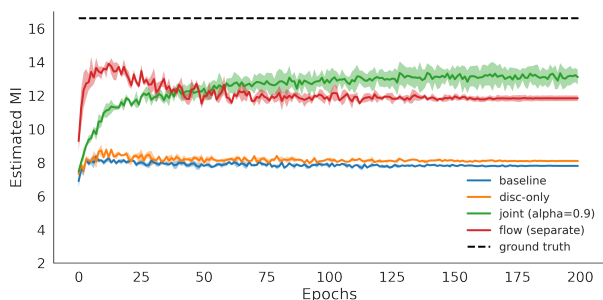

Figure 4: Estimated MI for various training strategies. The true MI for the corresponding value of $\rho = 0.9$ is 16.67. While separate training outperforms all baselines, joint training achieves competitive performance with larger α.

5.3 TARGETED GENERATION WITH MNIST

For this experiment, we evaluate the effectiveness of our learned density ratio estimates on a targeted generation task using the MNIST dataset. Our goal is to generate samples

according to a target distribution $q(\mathbf{x})$ in a data-efficient manner, given samples from both $p(\mathbf{x})$ and $q(\mathbf{x})$. We test two scenarios: (a) `diff-digits`: a subset of MNIST in which $p(\mathbf{x})$ is comprised of the digits labeled $\{1,2\}$, and $q(\mathbf{x})$ which is comprised of the digits labeled $\{0,7\}$; (b) `diff-background`: a setting in which $p(\mathbf{x})$ contains the original MNIST digits (black background, white digits); and $q(\mathbf{x})$ contains the same examples but with flipped colors (white background, black digits). The second scenario is trickier than the first, since there exists an obvious gap between the two distributions. We also explore the effect of the target dataset size $q(\mathbf{x})$ in learning accurate density ratios. Following the setup of [Choi et al., 2020], we sweep over various sizes of $q(\mathbf{x})$ relative to $p(\mathbf{x})$, which we call `perc`={0.1, 0.25, 0.5, 1.0} (where `perc` $= 0.5$ indicates that $\mathcal{D}_q$ is 50% the size of $\mathcal{D}_p$). After training a MAF on both $\mathcal{D}_p$ and $\mathcal{D}_q$ and obtaining density ratio estimates (importance weights), we sample from the trained MAF via sampling-importance-resampling (SIR) [Liu and Chen, 1998, Doucet et al., 2000] at generation time.

As shown in Table 2, we achieve greater success in the targeted generation task when performing SIR with importance weights learned in feature space. Averaged across `perc`={0.1, 0.25, 0.5, 1.0} with 1000 generated samples each, our method generates 19.1% more samples from $q(\mathbf{x})$ relative to the pretrained flow and 6.7% more samples than the baseline with importance weights learned in input space on the `diff-digits` task. Similarly for the `diff-background` task, our framework generates 18.8% more samples from $q(\mathbf{x})$ relative to the pretrained flow and 16.4% more samples than the baseline. For additional experimental details, as well as the generated samples, we refer the reader to Appendix D and F.

5.4 CLASSIFICATION WITH DATA AUGMENTATION ON OMNIGLOT

Finally, we follow the experimental setup of [Grover et al., 2019] by utilizing Data Augmentation Generative Adversarial Networks (DAGAN) [Antoniou et al., 2017] as a genera-

Different Digits	perc=0.1	perc=0.25	perc=0.5	perc=1.0
SIR with IW(z) (ours)	**0.447 ± 0.020**	0.518 ± 0.008	**0.777 ± 0.018**	**0.860 ± 0.004**
SIR with IW(x)	0.441 ± 0.002	**0.528 ± 0.004**	0.639 ± 0.007	0.754 ± 0.007
Regular sampling	0.406 ± 0.055	0.457 ± 0.07	0.596 ± 0.052	0.720 ± 0.035
Different Backgrounds	perc=0.1	perc=0.25	perc=0.5	perc=1.0
SIR with IW(z) (ours)	**0.186 ± 0.005**	**0.377 ± 0.001**	**0.580 ± 0.005**	**0.732 ± 0.008**
SIR with IW(x)	0.085 ± 0.003	0.202 ± 0.003	0.345 ± 0.013	0.528 ± 0.022
Regular sampling	0.084 ± 0.003	0.196 ± 0.003	0.304 ± 0.003	0.493 ± 0.016

Table 2: MNIST targeted generation results averaged over 3 runs. Columns show the fraction of generated samples with the target attribute (higher is better) across varying sizes of the target dataset. 1000 samples were generated for each setup.

tive model for importance-weighted data augmentation on the Omniglot dataset [Lake et al., 2015]. Since Omniglot is comprised of 1600+ classes with only 20 examples per class, the goal of this experiment is improve the performance of a downstream multi-class classifier by effectively leveraging additional samples generated by the DAGAN. To do so, we train a separate probabilistic classifier to distinguish between the true and the generated examples, yielding importance weights for each synthetic example that can be used for training the downstream classifier of interest.

We first train a MAF on a mixture of the training examples and generated samples, encode all the data using the flow, and obtain importance weights via the encodings. The importance weights are obtained by training a binary classifier on the featurized inputs. We experiment with different baselines: (a) training the classifier without any data augmentation (`Data-only`); (b) training the classifier on purely synthetic samples (`Synthetic-only`); (c) training the classifier with data-augmentation without any importance weighting (`Mixture-only`); (d) the data-augmented classifier with importance weights obtained from input space (`Mixture + IW(x)`); and (e) the data-augmented classifier with importance weights obtained from feature space (`Mixture + IW(z)`). As shown in Table 3, the importance weights learned in the feature space show a significant boost in overall downstream classification accuracy as compared to relevant baselines: our method improves 3.7% over the `Data-only` baseline, and 2.2% over the highest performing baseline. We refer the reader to Appendix F for additional experimental details and results.

6   RELATED WORK

Density Ratio Estimation in Feature Space.   Although density ratio estimation in machine learning has an extremely rich history [Friedman et al., 2001, Huang et al., 2006, Nguyen et al., 2007, Gutmann and Hyvärinen, 2010, Sugiyama et al., 2012b], there is considerably less work exploring the method's counterpart in feature space. [Rhodes et al., 2020], while tackling the same problem of density

ratio estimation between two different data distributions, adopts a different approach than our framework. In particular, they propose a divide-and-conquer solution by constructing intermediate distributions between the two densities $p(\mathbf{x})$ and $q(\mathbf{x})$, and requires the training of a multi-task logistic regression model rather than a single binary classifier. Their interpolation technique, which is also conducted in the latent space of a normalizing flow in one of their experiments, is complementary to our framework – investigating the combination of these two approaches would be interesting future work. Additionally, density ratio estimation (in the form of learning importance weights) has been popular in a variety of domain adaptation approaches such as [Bickel et al., 2007, Long et al., 2015, You et al., 2019] which leverage a feature extractor to project the inputs into a lower-dimensional manifold prior to estimation. Although our approach shares a similar idea, the invertibility of our feature map guarantees that the density ratios between input space and feature space are equivalent – this is not necessarily true if the inputs are lossily compressed.

Neural Hybrid Models.   Combining both generative and discriminative training approaches in neural networks has previously been explored in the literature [Maaløe et al., 2016, Gordon and Hernández-Lobato, 2017, Kuleshov and Ermon, 2017]. Our work bears most similarity to [Nalisnick et al., 2019], as we also require learning an invertible generative model and a discriminator. However, our method does not require that the normalizing flow be trained together with the probabilistic classifier, and can be used for more downstream applications beyond out-of-distribution detection and semi-supervised learning, as our goal is to accurately estimate density ratios. Additionally, our approach is related to conditional normalizing flows such as [Dinh et al., 2019] and [Winkler et al., 2019] which explicitly partition the latent space of the flow $p_Z(\mathbf{z})$ to map different components of the input into disjoint regions in the prior. Although we empirically verify that this is also the case for our method, it is more general precisely because the best partitioning is *learned* by the model.

Dataset	Data-only	Synthetic-only	Mixture-only	Mixture + IW(x)	Mixture + IW(z)
Accuracy	0.756 ± 0.001	0.557 ± 0.003	0.767 ± 0.003	0.765 ± 0.005	$\mathbf{0.784 \pm 0.007}$

Table 3: Downstream predictive accuracy on the Omniglot dataset. Standard errors are computed over 3 runs.

7 CONCLUSION

In this paper, we proposed a general-purpose framework for improved density ratio estimation in settings where the two underlying data distributions of interest are sufficiently different. The key component of our approach is a normalizing flow that is trained on a mixture of the data sources, which is then used to encode the data into a shared feature space prior to estimating density ratios. By leveraging the invertibility of the flow, we showed that the ratios of the densities in feature space are not only identical to those in input space, but are also easier to learn. Additionally, our method is applicable to a suite of existing density ratio estimation techniques. Empirically, we demonstrated the utility of our framework on various combinations of density ratio estimation techniques and downstream tasks that rely on accurate density ratios for good performance, such as domain adaptation, mutual information estimation, and targeted generation in deep generative models. We provide a reference implementation in PyTorch [Paszke et al., 2017], and the codebase for this work is open-sourced at `https://github.com/ermongroup/f-dre`.

One limitation of our method is the need to train a normalizing flow on a mixture of the two datasets if a pre-trained model is not available; this may be difficult if the generative model must be extremely high-capacity. For future work, it would be interesting to explore whether the necessity of strict invertibility of the flow can be relaxed, and to gain a deeper theoretical understanding of the role of maximum likelihood training in our framework.

Author Contributions

Kristy Choi and Madeline Liao had an equal contribution to this paper.

References

Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2019.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pages 531–540. PMLR, 2018.

Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning for differing training and test distributions. In *Proceedings of the 24th international conference on Machine learning*, pages 81–88, 2007.

Kristy Choi, Aditya Grover, Trisha Singh, Rui Shu, and Stefano Ermon. Fair generative modeling via weak supervision. In *International Conference on Machine Learning*, pages 1887–1898. PMLR, 2020.

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In *Nips*, volume 10, pages 442–450. Citeseer, 2010.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Laurent Dinh, Jascha Sohl-Dickstein, Razvan Pascanu, and Hugo Larochelle. A rad approach to deep mixture models. *arXiv preprint arXiv:1903.07714*, 2019.

Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville,

and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

Jonathan Gordon and José Miguel Hernández-Lobato. Bayesian semisupervised learning with deep generative models. *arXiv preprint arXiv:1706.09751*, 2017.

Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, and Bernhard Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Aditya Grover, Jiaming Song, Ashish Kapoor, Kenneth Tran, Alekh Agarwal, Eric J Horvitz, and Stefano Ermon. Bias correction of learned generative models using likelihood-free importance weighting. In *Advances in Neural Information Processing Systems*, pages 11058–11070, 2019.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

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

Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19:601–608, 2006.

Jörn-Henrik Jacobsen, Arnold Smeulders, and Edouard Oyallon. i-revnet: Deep invertible networks. *arXiv preprint arXiv:1802.07088*, 2018.

Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *The Journal of Machine Learning Research*, 10:1391–1445, 2009.

Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving variational inference with inverse autoregressive flow. *arXiv preprint arXiv:1606.04934*, 2016.

Volodymyr Kuleshov and Stefano Ermon. Deep hybrid models: Bridging discriminative and generative approaches. In *Proceedings of the Conference on Uncertainty in AI (UAI)*, 2017.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Jun S Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044, 1998.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

Marco Loog. Nearest neighbor-based importance weighting. In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, 2012. doi: 10.1109/MLSP.2012.6349714.

Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. In *International conference on machine learning*, pages 1445–1453. PMLR, 2016.

Aditya Menon and Cheng Soon Ong. Linking losses for density ratio and class-probability estimation. In *International Conference on Machine Learning*, pages 304–313. PMLR, 2016.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Hybrid models with deep and invertible features. *arXiv preprint arXiv:1902.02767*, 2019.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by penalized convex risk minimization. In *NIPS*, pages 1089–1096, 2007.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *arXiv preprint arXiv:1606.00709*, 2016.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *arXiv preprint arXiv:1705.07057*, 2017.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

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

Benjamin Rhodes, Kai Xu, and Michael U Gutmann. Telescoping density-ratio estimation. *arXiv preprint arXiv:2006.12204*, 2020.

Alex Smola, Le Song, and Choon Hui Teo. Relative novelty detection. In *Artificial Intelligence and Statistics*, pages 536–543. PMLR, 2009.

Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. *arXiv preprint arXiv:1910.06222*, 2019.

Yang Song, Chenlin Meng, and Stefano Ermon. Mintnet: Building invertible neural networks with masked convolutions. *arXiv preprint arXiv:1907.07945*, 2019.

Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul Von Buenau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS*, volume 7, pages 1433–1440. Citeseer, 2007.

Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical Mathematics*, 64(5):1009–1044, 2012a.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012b.

Owen Thomas, Ritabrata Dutta, Jukka Corander, Samuel Kaski, Michael U Gutmann, et al. Likelihood-free inference by ratio estimation. *Bayesian Analysis*, 2021.

Yuta Tsuboi, Hisashi Kashima, Shohei Hido, Steffen Bickel, and Masashi Sugiyama. *Direct Density Ratio Estimation for Large-scale Covariate Shift Adaptation*, pages 443–454. 2008. doi: 10.1137/1.9781611972788.40. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611972788.40.

Vladimir Vapnik, Igor Braga, and Rauf Izmailov. Constructive setting of the density ratio estimation problem and its rigorous solution. *arXiv preprint arXiv:1306.0407*, 2013.

Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

Makoto Yamada and Masashi Sugiyama. Direct importance estimation with gaussian mixture models. *IEICE Transactions*, 92-D:2159–2162, 10 2009. doi: 10.1587/transinf.E92.D.2159.

Makoto Yamada, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Masashi Sugiyama. Relative density-ratio estimation for robust distribution comparison. *Neural computation*, 25(5):1324–1370, 2013.

Kaichao You, Ximei Wang, Mingsheng Long, and Michael Jordan. Towards accurate model selection in deep unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 7124–7133. PMLR, 2019.