Training Deep Generative Models in Highly Incomplete Data Scenarios with Prior Regularization

Edgar A. Bernal
Rochester Data Science Consortium, University of Rochester
edgar.bernal@rochester.edu

Abstract

Deep generative frameworks including GANs and normalizing flow models have proven successful at filling in missing values in partially observed data samples by effectively learning –either explicitly or implicitly– complex, high-dimensional statistical distributions. In tasks where the data available for learning is only partially observed, however, their performance decays monotonically as a function of the data missingness rate. In high missing data rate regimes (e.g., 60% and above), it has been observed that state-of-the-art models tend to break down and produce unrealistic and/or semantically inaccurate data. We propose a novel framework to facilitate the learning of data distributions in high paucity scenarios that is inspired by traditional formulations of solutions to ill-posed problems. The proposed framework naturally stems from posing the process of learning from incomplete data as a joint optimization task of the parameters of the model being learned and the missing data values. The method involves enforcing a prior regularization term that seamlessly integrates with objectives used to train explicit and tractable deep generative frameworks such as deep normalizing flow models. We demonstrate via extensive experimental validation that the proposed framework outperforms competing techniques, particularly as the rate of data paucity approaches unity.

1. Introduction

Deep generative models (DGMs) have enjoyed success in tasks involving the estimation of statistical properties of data. Applications of DGMs involve generation of high-resolution and realistic synthetic data [2, 13, 35, 38, 39], exact [6, 21] and approximate [22, 40] likelihood estimation, clustering [53], representation learning [1, 54], and unsupervised anomaly detection [23]. Fundamentally, generative models perform explicit and/or implicit data density estimation [14]. Given the complexity of most signals of interest to the learning community (e.g., audio, language, imagery and video), reliably learning the statistical properties of a given population of data samples often requires immense amounts of training data. Recent work has empirically shown that, in order to continue pushing the state-of-the-art in high-fidelity synthetic data generation, scalable models able to ingest ever-growing data sources may be required [2].

Some of the data requirements imposed by current deep generative models may limit their applicability in real-life scenarios, where available data may not be plentiful, and additionally, may be noisy, or only partially observable. The nature of real-world data poses challenges to existing models, and mechanisms to overcome those challenges are needed in order to further the penetration of the technology. In this paper, we focus on enabling the learning of DMGs in scenarios of high data missingness rates (e.g., 60% of entries missing per data sample and above), where the missingness affects *both* the training and the test sets. We specifically focus on the task of image imputation, which consists in filling in missing or unobserved values without access to fully observed images during training. Previous work on data imputation leveraging various forms of DMGs has explicitly addressed image imputation [29, 30, 41, 55]. While the results are reasonable in low- and mid-data missingness regimes, empirical results indicate that, as large fractions of the data become unobserved, either the perceived quality of the recovered data suffers [41], the original semantic content in the image is lost [55] or both [29]. These undesired consequences are likely caused by the ill-posedness of the problem of attempting to estimate certain statistical properties from partially observed data, an issue which becomes more extreme as the rate of unobserved data approaches totality. Of note, most existing work fails to consider the semantic content preservation aspect of the task altogether, and focuses solely on measuring the performance of the algorithms based on the realism of the recovered data samples [29, 30, 55].

Inspired by these observations, we propose to constrain the complexity of the solution space where the reconstructed image lies via regularization techniques, a technique initially exploited in traditional ill-posed inverse

problem formulations [49] and more recently adapted to statistical learning scenarios [51]. The proposed regularization term enforces a prior distribution on the gradient map of the reconstructed images [24] in the form of a shallow, hand-engineered constraint, and stands in contrast with recent trends which rely on the high expressivity and capacity of deep models to effectively construct data-driven priors, but which break down in scenarios where data scarcity is an issue. We seamlessly couple the regularizing priors with explicit likelihood estimates of reconstructed samples yielded by normalizing flows in a novel framework we dub *PRFlow*, which stands for *P*rior-*R*egularized Normalizing *Flow*. The contributions of this paper are as follows:

- a framework combining traditional explicit and tractable deep generative models with shallow, hand-engineered priors in the form of regularization terms to constrain the complexity of the solution space in high data paucity regimes;
- a formal derivation of the framework stemming from the formulation of the learning task with incomplete data as a joint optimization task of the network parameters and missing data values;
- a comprehensive testing framework –including a new metric that captures the semantic consistency between the original and the recovered data samples– which evaluates all aspects of performance that are relevant when learning from partially observed data; and
- empirical validation of the effectiveness of the proposed framework on the imputation of three standard image datasets and benchmarking against current state-of-the-art imputation models under the proposed testing framework.

2. Related Work

Deep learning frameworks have proven successful at a wide range of applications such as speech recognition, image and video understanding, and game playing, but are often criticized for their data-hungry nature [33]. Some scholars go as far as to say that the future of deep learning depends on data efficiency, and have attempted to achieve it in various ways, for example, by leveraging common sense [48], mimicking human reasoning [12] or incorporating domain knowledge into the learning process [36]. The ability to learn from incomplete, partially observed and noisy data will be fundamental to advance the adoption of deep learning frameworks in real-life applications. In recent years, a body of research on deep frameworks that can learn from partially observed data has emerged. Initial work focused on extensions of generative models such as Variational Auto Encoders (VAEs) [22] and Generative Adversarial Networks (GANs) [13], including Partial VAEs [32], the Missing Data Importance-Weighted Autoencoder (MIWAE) [34], the Generative Adversarial Imputation Network (GAIN) [55] and the GAN for Missing Data (MISGAN) [29]. More recently, the state-of-the-art benchmark on learning from incomplete data has been pushed by bidirectional generative frameworks which leverage the ability to map back and forth between the data space and the latent space. Two such examples include the Monte-Carlo Flow model (MCFlow) [41] which relies on explicit normalizing flow models [5, 6, 21], and the Partial Bidirectional GAN (PBiGAN) [30] which extends the bidirectional GAN framework [7, 8].

While the results achieved by recent work are impressive in their own right, these methods share a common thread: they all break down, in one way or another, as the missingness rate in the data approaches unity. This phenomenon can be intuitively understood if we think of a generative model as a probability density estimator (either explicit or implicit) [14], which is, at its core, an ill-posed inverse problem [4, 43]. From this standpoint, the ill-posedness becomes more extreme as the rate of occurrence of unobserved data increases. Historically, regularization techniques [9, 49] have been widely used to precondition estimators and avoid undesired behaviors of solutions by restricting the feasible space [10, 51]. While regularization in deep learning is commonplace (e.g., weight decay and weight sharing [37], dropout [45], batch normalization [19]), it is usually implemented to constrain the plausible space of network parameters and avoid overfitting in discriminative scenarios. Models that implement regularization on the *output* space tend to be of the generative type. For instance, image priors have been leveraged to address the inherently ill-posed single-image super resolution problem [20, 28, 46, 52, 56]. The proposed framework can be seen as an attempt to incorporate domain knowledge in learning scenarios in order to guide, facilitate or expedite the learning [11, 17, 18].

3. Proposed Framework

Parallels between ill-posed inverse problems and learning tasks have been established in the literature [42, 51]. To informally illustrate how the degree of ill-posedness of a learning task from partial observation grows with the rate of data missingness, consider the task of image imputation. Let b denote the bit depth used to encode each pixel value (i.e., pixels can take on values g, where $0 \leq g \leq 2^b - 1$) and N the number of pixels of the images in question. The total number of possible images that can be represented with this scheme is $(2^b)^N$. For the sake of discussion, let us ignore the fact that natural images actually lie on a lower-dimensional manifold within that image space. Let $0 \leq p \leq 1$ denote the data missingness rate. This means that when we partially observe an image, we are only exposed to $(1-p) \cdot N$ of its pixel values. The task of image

imputation involves estimating the remaining $p \cdot N$ pixel values, which means that for every partially observed input image, there are $(2^b)^{pN}$ possible imputed solutions. It is apparent that the dimensionality of the feasible solution space grows exponentially as the missingness rate grows linearly. The practical implication of this observation is that, in order to maintain a certain level of reconstruction performance, the number of partially observed data samples needs to grow exponentially as the missingness rate grows linearly. This is an example of an ill-posed problem where the observed data itself is not sufficient to find unique solutions.

When an imputation task is tackled with a learning framework (i.e., a deep generative network), the inductive bias that arises from the choice of network inherently constraints the solution space. This restriction is not only convenient but also necessary for learning [3], as illustrated by recent work which shows that the structure of a network captures natural image statistics prior to any training [50]. We will demonstrate empirically that inductive bias alone is not sufficiently effective at restricting the solution space in cases where data is missing at high rates. Experimental results conclusively show that augmenting the constraining properties of the inductive bias with shallow priors implemented in the form of regularizers is a simple an effective strategy in boosting the performance of deep models in scenarios of high data paucity.

3.1. Framework Description

Normalizing flow models are explicit generative models which perform tractable density estimation of the observed data. The density estimate is constructed by learning a cascade of invertible transformations which perform a mapping between the data space and a latent space. A simple, continuous prior is assumed on the latent variables, for example a spherical Gaussian density. Exact log-likelihood computation is achieved using the change of variable formula [5, 6, 21]. In this work, we introduce a principled framework that leverages the explicit and tractable likelihood capabilities of normalizing flow models to impose structured constraints on the constructed probabilistic models.

Although the proposed framework is generic enough to support a wide range of prior constraints, this study leverages the Hyper-Laplacian prior [24], which has been proven effective at modelling the heavy-tailed nature of the distribution of gradients in natural scenes. This distribution takes on the form $p_p(z) \propto e^{-k|z|^\alpha}$ (or equivalently, $\log p_p(z) \propto -k|z|^\alpha$), where $0 < \alpha \leq 1$ determines the heaviness of the tails in the distribution, and z is the gradient map of image x, which can be obtained by convolving x with a family of kernels f_i. Subscript p is used to denote the nature of the distribution (i.e., to contrast with data-driven priors). We use the notation $z = x * f_i$ to denote the convolution between image x and kernel f_i. When multiple filters are used, it is common to assume independence of the different edge maps so that $\log p_p(x) \propto -\sum_{i=1}^{I} |x * f_i|^\alpha$, where I is the total number of filters.

In scenarios where training data is only partially observed, training a normalizing flow model can be formulated as a joint optimization task where two sets of parameters are learned concurrently, namely the missing entries in the data and the parameters of the normalizing flow model itself. Let x_{rec} denote the reconstructed samples and G_θ the normalizing flow network parameterized by θ. The objective of the learning task can be written as

$$(x_{rec}, \theta^*) = \arg\max_{x,\theta} \{p(x, \theta)\} \tag{1}$$

Note that, as per the above objective, missing data values are treated as parameters to optimize. Throughout the remainder of the paper, we will refer to these values interchangeably as data parameters or missing data values.

Estimating the joint density from Eq. 1 is difficult. One way to circumvent this obstacle is to alternately optimize over the conditional distributions of each of the parameters interest, in a manner similar to the way sampling-based optimization frameworks such as Gibbs Sampling and MCMC [47] operate. Following this principle, the joint optimization task can be broken down into two conditional optimization tasks of likelihood functions. On the one hand, learning the parameters θ of normalizing flow network G_θ can be achieved in the traditional manner, that is, by maximizing the log-likelihood of the observed data:

$$\theta^* = \arg\max_{\theta} \{p(\theta | x_{rec})\} \tag{2}$$

A set of parameters θ defines an invertible network G_θ that maps images to a tractable latent space and vice-versa. Specifically, in order to perform log-likelihood estimation, a data sample x_i is mapped to its latent representation y_i by passing it through G_θ, namely $y_i = G_\theta(x_i)$. Since the likelihood for y_i is known (e.g., from a normality assumption), $p(x_i)$ (i.e., the likelihood of x_i) can be computed exactly via the variable change rule. The ability to estimate the likelihood of a data sample enables the resolution of the second conditional optimization task, which aims at finding the optimal entries for the missing values in the partially observed data by maximizing the likelihood of the reconstructed sample conditioned on the current model parameters:

$$x_{rec} = \arg\max_{x} \{p(x | \theta^*)\} \tag{3}$$

where the search space is constrained to images x whose entries match the observed entries of x_{obs}. Solving the optimization task from Eq. 3 effectively fills in unobserved data values, that is, performs data imputation. Training the overall imputation model involves alternately solving Eqs. 2 and 3, which yields a sequence of parameter pairs $(x_{rec}^{(n)}, \theta^{*(n)})$.

Convergence is achieved when little change is observed in the updated parameters. The description of the framework around Eqs. 2 and 3 follows closely the formulation in [41], although in that work, the training of the model was not framed as a joint optimization task.

As stated, solving Eq. 2 involves training a traditional normalizing flow model with the current estimate of the data parameters, i.e., the current values of the imputed data. In contrast, the optimization task in Eq. 3 is a highly ill-posed problem when the data missing rate in x_{obs} is high. PRFlow leverages the key insight that regularization of the task with prior knowledge on the solution space leads to improved, more stable solutions to Eq. 3. In order to incorporate this prior knowledge, first observe that, as per the Bayes rule:

$$p(x|\theta^*) \propto p(\theta^*|x)p_p(x) \tag{4}$$

where $p_p(x)$ is the prior introduced at the beginning of Sec. 3.1, and it has been assumed that model parameters θ^* are fixed. This is the case since at this stage in the training alternation, the optimization is over the missing data entries with the goal of performing data imputation. Combining Eqs. 3 and 4 and applying log yields

$$x_{rec} = \arg\max_x \{\log p(\theta^*|x) + \lambda \log p_p(x)\} \tag{5}$$

where λ is a parameter that controls the desired degree of regularization. In summary, training PRFlow involves alternately optimizing the objectives in Eqs. 2 and 5. It is worthwhile noting that the objective from Eq. 2 and the first term in the objective from Eq. 5 involve optimizing the same likelihood function relative to two different sets of parameters, namely the model parameters and the missing data values, respectively.

3.2. Framework Implementation

PRFlow is largely based on the architecture introduced in [41], which includes a normalizing flow network G that enables likelihood estimation, and a network H performing a non-linear mapping in the latent flow space and fills in missing values in the partially observed data samples. As in [41], network G is an instantiation of RealNVP [6]. The mapping to the latent space via G is performed because likelihood computation is tractable in that space, and the imputation task is being formulated as the solution of a maximum likelihood conditional objective (as per Eq. 3). At a high level, the imputation process comprises receiving a partially observed sample x_{obs}, computing its latent representation $y_{obs} = G_\theta(x_{obs})$, mapping this latent representation to $y_{rec} = H_\phi(y_{obs})$ with maximum likelihood, and recovering the corresponding maximum likelihood data sample $x_{rec} = G_\theta^{-1}(y_{rec})$ which matches the observed entries of x_{obs}. This process is illustrated in Fig. 1.

As described in Sec. 3.1, learning this framework involves optimizing two different objectives: training network G_θ and H_ϕ involves optimizing the objectives from Eqs. 2 and 5 respectively, with the optimization being carried out in an alternating way until convergence is achieved. The objectives used to learn these networks, as described below, are denoted $\mathcal{J}(\theta)$ and $\mathcal{J}(\phi)$. In the context of the proposed framework, the data parameters are not optimized directly; instead, network H_ϕ is learned according to $\mathcal{J}(\phi)$, a proxy objective to that in Eq. 5. We now describe how the two networks are learned.

Learning the optimal parameters θ^* of normalizing flow network G_θ is achieved by maximizing the log-likelihood of the training data, or equivalently, minimizing the cost function:

$$\mathcal{J}(\theta) = -\sum_i \log p_\theta(x_i^{(n)}) \tag{6}$$

where the sum is computed across training data samples, and the superscript (n) denotes samples which have been imputed with the most recent (i.e., the n-th) imputation model. Throughout this optimization stage, the training data remains unchanged. At initialization, where no imputation model is available, shallow imputation techniques (e.g., nearest neighbor or bilinear interpolation) are used. Minimizing this loss corresponds to solving the optimization task from Eqs. 2.

Learning the optimal parameters ϕ^* of the imputation model, which operates in the latent space of the normalizing flow network, is achieved by minimizing a three-term loss. Updating parameters ϕ results in an updated imputer network H_ϕ, which is used to obtain an updated training set $x^{(n)}$. Throughout this stage, normalizing flow network G_θ remains fixed. The first element of the loss involves maximizing the likelihood of the reconstructed samples as per the likelihood estimate provided by the normalizing flow model, or equivalently, minimizing the cost function:

$$\begin{aligned}\mathcal{J}_1(\phi) &= -\sum_i \log p_\theta(x_i^{(n)}) \\ &= -\sum_i \log p_\theta \left[G_\theta^{-1} \circ H_\phi \circ G_\theta(x_i^{(n-1)})\right]\end{aligned} \tag{7}$$

where the $\circ$ operator denotes functional composition and the expression for $x_i^{(n)}$ has been expanded to emphasize its dependence on the parameters being optimized, namely ϕ. Minimizing this loss is equivalent to optimizing the first term of the objective from Eq. 5. As stated before (see last paragraph in Sec. 3.1), this loss is equivalent to the loss from Eq. 6; the difference lies in the set of parameters that are being modified to achieve the objective. This term encourages the imputer to output recovered samples that are more likely to occur.

The second element involves minimizing the discrepancy between the recovered data and the known entries of the observed data:

$$\mathcal{J}_2(\phi) = \sum_i \text{MSE}(x_{i,obs}, G_\theta^{-1} \circ H_\phi \circ G_\theta(x_i^{(n-1)})) \tag{8}$$

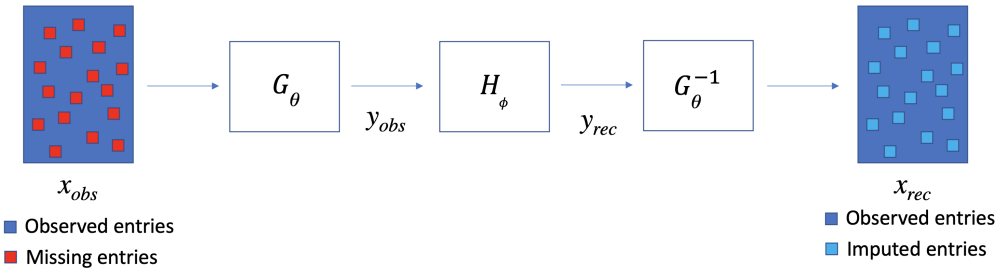

Figure 1: High-level view of the imputation process.

where the MSE is computed across the known entries of the observed data only. Note that these entries remain unchanged throughout both stages of the optimization process, thus no superscript is needed. This term encourages the imputer to output recovered samples that match the known entries at the observed positions.

The last term penalizes reconstructions that deviate from the expected behavior as dictated by the regularizing prior:

$$\begin{aligned}\mathcal{J}_3(\phi) &= -\sum_i \log p_p(x_i^{(n)}) = -\sum_i \sum_j |x_i^{(n)} * f_j|^\alpha \\ &= -\sum_i \sum_j |G_\theta^{-1} \circ H_\phi \circ G_\theta(x_i^{(n-1)}) * f_j|^\alpha\end{aligned} \tag{9}$$

where the summations indexed by i and j are performed across data samples and gradient kernels, respectively, and we have incorporated the expression for the prior introduced in Sec. 3.1. Minimizing this loss is equivalent to optimizing the second term in the objective from Eq. 5. In our implementation, and for the sake of computational efficiency and simplicity, we use two first-order derivative filters, namely $[1, 1]$ and $[1, 1]^\intercal$. Note that higher-order or learnable filters can be used instead, which would likely result in improved performance.

In summary, training PRFlow involves joint optimization of objectives $\{\mathcal{J}(\theta), \mathcal{J}_1(\phi), \mathcal{J}_2(\phi), \mathcal{J}_3(\phi)\}$ across θ and ϕ, where θ denotes the parameters of the normalizing flow network and ϕ denotes the imputer network parameters, i.e., the parameters that ultimately determine how the missing data values are filled in.

4. Experimental Results

Datasets and Procedure. The efficacy of PRFlow was evaluated on three different standard image datasets, MNIST [27], CIFAR-10 [25] and CelebA [31]. Four different rates of data missingness were tested, from 60% to 90% in steps of 10%. The training procedure follows the principles of recent work proposing models that support and rely purely on partially observed data during the learning phase [29, 30, 41, 55] by training with the dataset resulting from randomly dropping the corresponding percentage of pixels from the images in the standard training set from the respective dataset of interest according to a Bernoulli distribution. In MNIST, the training set comprises 60,000 28×28-pixel grayscale images, whereas in CIFAR it includes 50,000 32×32-pixel RGB images. Since no standard partition exists for CelebA, we use the first 100,000 images for training and the remaining for testing. We pre-process CelebA images by performing 108×108 pixel center cropping and resizing to 32×32 pixels. For testing, we adhere to the experimental principle drawn out in [41], where performance is measured on the standard test set of the relevant dataset after having randomly dropped the appropriate fraction of pixel values.

Metrics. We measure the performance of the algorithms relative to three different metrics, which we believe capture all relevant attributes of data recovered by an algorithm attempting to reconstruct partially observed data: (i) root mean squared error (RMSE), which measures differences between the reconstructed image and the ground truth at the pixel level; (ii) the Fréchet Inception Distance (FID), first proposed to measure the quality of data produced by generative models [16] and which captures population-level similarities; and (iii) the ratio of the classification accuracy of a classifier pre-trained on fully observed training data on the reconstructed data to the accuracy of the same classifier on the fully observed test set. This metric, which we denote the Semantic Consistency Criterion (SCC), aims at measuring the amount of semantic information preserved by the missing data recovery process. Formally, let acc_{imp} be the performance of the benchmark classifier on an imputed test set and acc_0 the performance of the same classifier on the original test set. Then SCC $= \min\{1, acc_{imp}/acc_0\}$, where the clipping is introduced to handle the unlikely case when $acc_{imp} > acc_0$. Normalization by the baseline classifier performance is done to minimize the impact of the choice of classifier. This overarching experimental framework contrasts with most previous work on generative modelling of incomplete data ([41] excepted), which doesn't

consider the preservation of semantic content as a metric of performance, and tends to make more emphasis on the realism of the recovered samples than on the pixel-level accuracy [29, 30]. In this work, we consider all three metrics to be equally important, and posit that one of the most salient strengths of the proposed method is that it minimizes the impact of the trade-off between the metrics relative to competing methods. Of note, RMSE is measured between the recovered values and the ground truth values at the unobserved pixel locations in the test set. This means that not only the pixels but also the full images used to measure the performance of the method are completely unseen by the framework during training, unlike approaches which measure performance on unobserved values within the training set [29, 30]. Similarly, FID is measured between the recovered test set and the ground truth test set, and SCC performance is measured on the recovered test set imagery.

Competing Methods. We benchmark the performance of PRFlow against three methods, namely MisGAN [29], PBiGAN [30] and MCFlow [41], which together comprise the state-of-the-art landscape in image imputation tasks across the considered metrics. We used the publicly available code for all three competing methods from their official repositories; we used the code as published for MNIST and made extensions to the code to enable support of CIFAR (no CIFAR versions were publicly available). We use LeNet [26], ResNet18 [15] to compute both SCC and FID on MNIST and CIFAR, respectively. Since CelebA has no classes, we use FaceNet [44] to compute FID only.

Experimental Setup. Throughout the experiments, we use $\alpha = 1/3$, a learning rate of 1×10^{-4}, and a batch size of 64. We train until little change is observed in the loss from Eq. 8, as opposed to competing methods which prescribe a set number of epochs to train. G_θ is a RealNVP [6] network with six affine coupling layers. We implement H_ϕ as a 3-hidden layer, fully connected network with 784 and 1024 neurons per layer for MNIST and CIFAR/CelebA, respectively, with input and output layers having the same number of neurons as the dimensionality of the images (i.e., $28 \times 28 = 784$ for MNIST, and $32 \times 32 \times 3 = 3072$ for CIFAR and CelebA). Although performance is somewhat robust to the choice of λ, we noticed it did affect convergence speed: too large a value would lead to oscillations and too small a value would lead to slow convergence. As a rule of thumb, we found that a value of λ that approximately equalizes the value of $\mathcal{J}_1(\phi)$ (Eq. 7) and the value of $\lambda\mathcal{J}_3(\phi)$ (Eq. 9) worked well.

Results. Table 1 includes the RMSE results for all competing methods across both datasets and considered data missingness rates. MCFlow and PRFlow perform similarly, while PBiGAN performs the worst, with the gaps in performance being significantly larger for CIFAR. These results are reasonable since neither MisGAN nor PBiGAN enforce an MSE loss explicitly. Table 2 includes the FID results laid out in a similar fashion. In this case, PRFlow again outperforms all competing methods, trailed closely by PBiGAN on MNIST, with performance being more even across the field on CIFAR and CelebA. These results highlight the efficacy of the regularizing prior at shaping the statistical behavior of the recovered imagery. Lastly, Table 3 includes SCC results. In the MNIST case, MCFlow, PBiGAN and PRFlow perform similarly, with MisGAN trailing by a somewhat significant margin, and with the margin increasing as the missing rate increases. In the CIFAR case, PRFlow outperforms the competition more handily.

Table 1: RMSE between recovered data and ground truth test set, unobserved pixels only (lower is better)

		Missing Rate			
Dataset	**Method**	0.6	0.7	0.8	0.9
MNIST	MisGAN	0.1329	0.1561	0.1958	0.2484
	PBiGAN	0.3155	0.3121	0.3045	0.2844
	MCFlow	0.1126	0.1300	0.1581	0.2080
	PRFlow	**0.1093**	**0.1243**	**0.1490**	**0.2059**
CIFAR	MisGAN	0.2568	0.2814	0.3081	0.3461
	PBiGAN	0.3380	0.3443	0.3623	0.4448
	MCFlow	0.0921	0.1059	0.1187	0.1460
	PRFlow	**0.0802**	**0.0919**	**0.1102**	**0.1299**
CelebA	MisGAN	0.2232	0.2273	0.2404	0.2777
	PBiGAN	0.2894	0.3356	0.3733	0.4230
	MCFlow	0.0793	0.0828	0.0927	0.1189
	PRFlow	**0.0738**	**0.0813**	**0.0924**	**0.1135**

Table 2: FID between recovered data and ground truth test sets (lower is better)

		Missing Rate			
Dataset	**Method**	0.6	0.7	0.8	0.9
MNIST	MisGAN	0.8300	1.5373	3.0956	7.9071
	PBiGAN	0.1356	0.3082	0.9927	4.2000
	MCFlow	0.7840	1.3382	3.0663	8.5047
	PRFlow	**0.0959**	**0.2888**	**0.8795**	**3.8759**
CIFAR	MisGAN	0.7299	0.8464	0.9136	0.9477
	PBiGAN	0.8743	0.9794	1.1229	1.1308
	MCFlow	0.4145	0.6564	0.8777	1.0808
	PRFlow	**0.2928**	**0.5111**	**0.6825**	**0.8437**
CelebA	MisGAN	0.3085	0.3486	0.4024	0.5693
	PBiGAN	0.7547	0.7861	0.8931	0.9415
	MCFlow	0.1225	0.1672	0.3333	0.7587
	PRFlow	**0.0887**	**0.1481**	**0.2359**	**0.5213**

Figs. 2 through 5 include sample reconstruction results which are intended to qualitatively showcase the performance of the competing methods. The results in Figs. 2 and 3 are arranged in groups of two rows of images, each group corresponding to reconstructions from the observed image (top row) and ground truth (bottom row) in the leftmost column of each image group. The remaining images in the top row of each group correspond to reconstructions by MisGAN, PBiGAN, MCFlow and PRFlow, respectively,

Table 3: SCC of recovered test set (higher is better)

Dataset	Method	Missing Rate			
		0.6	0.7	0.8	0.9
MNIST	MisGAN	0.9423	0.8763	0.6964	0.3489
	PBiGAN	0.9807	0.9619	0.9183	**0.7602**
	MCFlow	**0.9872**	**0.9705**	**0.9279**	0.7487
	PRFlow	0.9842	0.9693	0.9276	0.7471
CIFAR	MisGAN	0.4588	0.3828	0.3364	0.2737
	PBiGAN	0.3717	0.3020	0.2396	0.1757
	MCFlow	0.6606	0.5194	0.3893	0.3218
	PRFlow	**0.7225**	**0.5939**	**0.4719**	**0.3559**

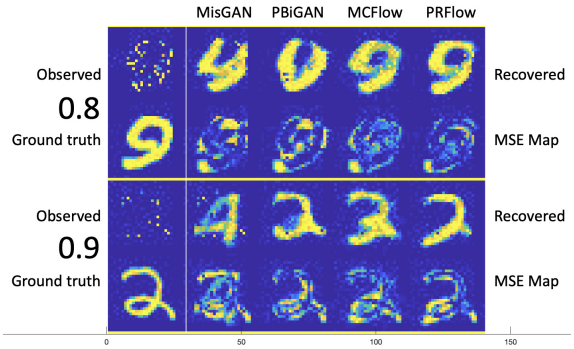

Figure 2: Sample results on MNIST for 80 and 90% missing rates (top to bottom image groups).

from left to right. The bottom row in each group includes the mean squared error maps between the reconstruction by each method and the ground truth. Fig. 2 includes results across different rates of missing data. It can be observed that, as the results from Table 1 indicate, GAN-based methods tend to produce higher MSE reconstructions. Further, the reconstructions produced by PRFlow showcase human-like handwriting across all levels, with strokes that are mostly continuous and largely uninterrupted. Lastly, the images recovered by PRFlow almost always resemble a readable digit, which is not the case with the competing methods, particularly for missing rates of 80% and above.

Fig. 3 focuses on the 90% missing data case and provides four additional examples. As before, all of the images restored by PRFlow resemble human-like handwritten digits. Failure to recover the original semantic content of the images happens mostly in cases where the original images themselves are ambiguous. Figs. 4 and 5 include reconstruction results on CIFAR-10 and CelebA. From left to right, the images include: ground truth, observed, and reconstructions by MNIST, PBiGAN, MCFlow and PRFlow. It can be seen that the Flow-based methods outperform the GAN-based methods, with PBiGAN lagging significantly behind. PRFlow has the overall edge in image quality with sharper edges, smoother backgrounds and more realistic reconstructions. Specifically, the edges of the plane and mountains against the sky are sharp in PRFlow recon-

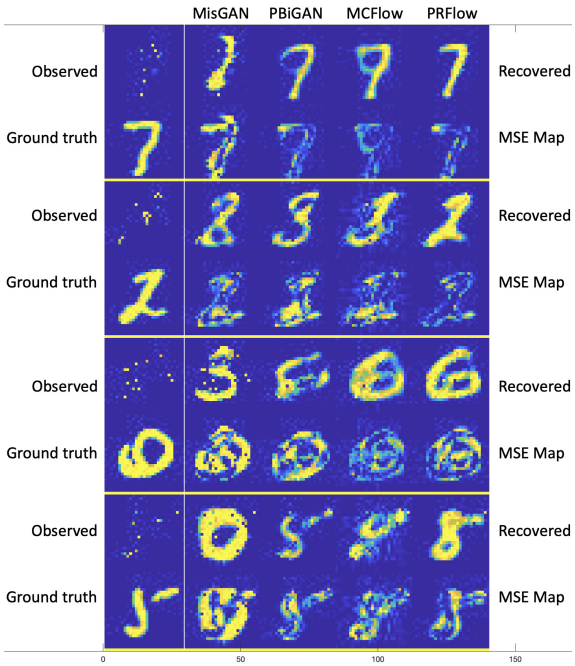

Figure 3: Sample results on MNIST for 90% missing data.

structions; the edges of sunglasses against skin are better defined; skin textures are more realistic and facial features (e.g., mouth, nose, hair strands) are rendered more naturally. While there are similarities between the MCFlow and PRFlow renditions, there are edge sharpness and texture differences (e.g., ringing and blockiness artifacts being more pronounced in the MCFlow images) that likely lead to the measurable gap in performance showcased in Tables 1-3. Lastly, the bottommost row in Fig. 5 illustrates a subtle but semantically significant reconstruction artifact where competing methods hallucinate a person with open eyes, while PRFlow accurately reconstructs a squinting face. We invite readers to attempt to fill in missing values themselves from the partially observed versions of the images. It can be a challenging task, in particular for high rates of missing data. We should note that humans have an advantage in that they know from experience what a number, an animal, or a face look like, whereas the algorithms competing herein were never exposed to a single fully observed image, and thus have to infer what the different objects look like by piecing together fractional observations from multiple images in the complete absence of labels.

5. Discussion

Traditionally, learning from incomplete or partially observed data has meant that trade-offs between various image quality aspects had to be incurred. Specifically, prior methods on image imputation suffered at one or more of the following image quality attributes: (i) realism, (ii) pixel-level quality, and (iii) semantic consistency between the re-

Figure 4: Sample results on CIFAR-10. From left to right: ground truth, observed, and MisGAN, PBiGAN, MCFlow and PRFlow reconstructions.

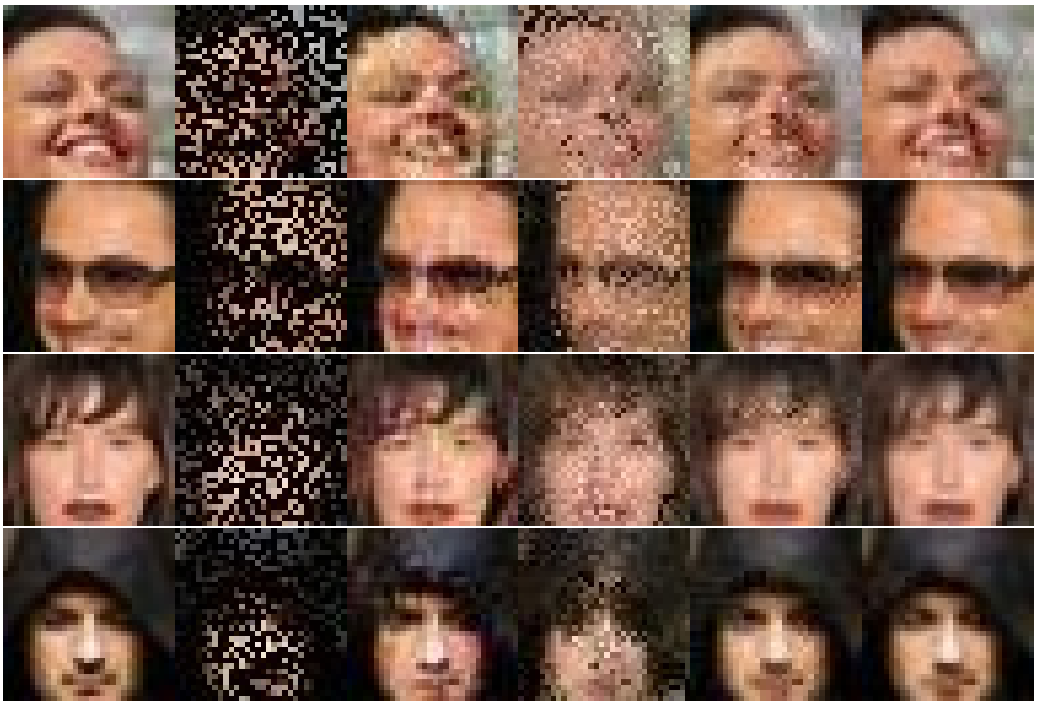

Figure 5: Sample results on CelebA. From left to right: ground truth, observed, and MisGAN, PBiGAN, MCFlow and PRFlow reconstructions.

covered and the partially observed image. These trade-offs became more significant as the degree of data paucity grew and approached unity. We hypothesize that this undesirable trend was due to the increasing level of ill-posedness of the recovery process and proposed a regularization approach that proved effective at addressing the three-pronged image quality trade-off. Extensive experimental results demonstrate that the proposed algorithm consistently matches or outperforms the performance of competing state-of-the-art approaches across all quality metrics in question. The seamless incorporation of domain knowledge in the form of a prior regularizer was made possible by the formulation of the learning task as a joint optimization objective.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 40–49, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 1

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 1

[3] Nadav Cohen and Amnon Shashua. Inductive bias of deep convolutional networks through pooling geometry. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 3

[4] A. K. Dey and F. H. Ruymgaart. Direct density estimation as an ill-posed inverse estimation problem. *Statistica Neerlandica*, 53(3):309–326, 1999. 2

[5] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *CoRR*, abs/1410.8516, 2014. 2, 3

[6] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 1, 2, 3, 4, 6

[7] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. 2

[8] Vincent Dumoulin, Mohamed Ishmael Diwan Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. 2017. 2

[9] H.W. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Mathematics and Its Applications. Springer Netherlands, 1996. 2

[10] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1), 2000. 2

[11] Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. *J. Mach. Learn. Res.*, 11:2001–2049, Aug. 2010. 2

[12] Dileep George, Wolfgang Lehrach, Ken Kansky, Miguel Lázaro-Gredilla, Christopher Laan, Bhaskara Marthi, Xinghua Lou, Zhaoshi Meng, Yi Liu, Huayan Wang, Alex Lavin, and D. Scott Phoenix. A generative vision model that trains with high data efficiency and breaks text-based captchas. *Science*, 358(6368), 2017. 2

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. 1, 2

[14] Ian J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2017. 1, 2

[15] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017. 5

[17] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2410–2420, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. 2

[18] Zhiting Hu, Zichao Yang, Russ R Salakhutdinov, Lianhui Qin, Xiaodan Liang, Haoye Dong, and Eric P Xing. Deep generative models with learnable knowledge constraints. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10501–10512. Curran Associates, Inc., 2018. 2

[19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR. 2

[20] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Image super-resolution using gradient profile prior. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2

[21] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10215–10224. Curran Associates, Inc., 2018. 1, 2, 3

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013. cite arxiv:1312.6114. 1, 2

[23] B. Kiran, Dilip Thomas, and Ranjith Parakkal. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging*, 4(2):36, Feb 2018. 1

[24] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1033–1041. Curran Associates, Inc., 2009. 2, 3

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 5

[26] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998. 6

[27] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. 5

[28] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, 2017. 2

[29] Steven Cheng-Xian Li, Bo Jiang, and Benjamin Marlin. MisGAN: learning from incomplete data with generative adversarial networks. In *International Conference on Learning Representations*, 2019. 1, 2, 5, 6

[30] Steven Cheng-Xian Li and Benjamin Marlin. Learning from irregularly-sampled time series: A missing data perspective. In *Proceedings of Machine Learning and Systems 2020*, pages 5756–5765. 2020. 1, 2, 5, 6

[31] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 5

[32] C. Ma, Wenbo Gong, José Miguel Hernández-Lobato, Noam Koenigstein, Sebastian Nowozin, and C. Zhang. Partial vae for hybrid recommender system. 2018. 2

[33] Gary Marcus. Deep learning: A critical appraisal. *CoRR*, abs/1801.00631, 2018. 2

[34] Pierre-Alexandre Mattei and Jes Frellsen. MIWAE: Deep generative modelling and imputation of incomplete data sets. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4413–4423, Long Beach, California, USA, 09–15 Jun 2019. PMLR. 2

[35] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. 1

[36] N. Muralidhar, M. R. Islam, M. Marwah, A. Karpatne, and N. Ramakrishnan. Incorporating prior domain knowledge into deep neural networks. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 36–45, 2018. 2

[37] Steven J. Nowlan and Geoffrey E. Hinton. Simplifying neural networks by soft weight-sharing. *Neural Comput.*, 4(4):473–493, July 1992. 2

[38] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio, 2016. cite arxiv:1609.03499. 1

[39] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 1

[40] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR. 1

[41] Trevor W. Richardson, Wencheng Wu, Lei Lin, Beilei Xu, and Edgar A. Bernal. MCFlow: Monte carlo flow models for data imputation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 4, 5, 6

[42] Lorenzo Rosasco, Andrea Caponnetto, Ernesto D. Vito, Francesca Odone, and Umberto D. Giovannini. Learning, regularization and ill-posed inverse problems. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1145–1152. MIT Press, 2005. 2

[43] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *Ann. Math. Statist.*, 27(3):832–837, 09 1956. 2

[44] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823. IEEE Computer Society, 2015. 6

[45] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, Jan. 2014. 2

[46] Y. Tai, S. Liu, M. S. Brown, and S. Lin. Super resolution using edge prior and single image detail synthesis. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2400–2407, 2010. 2

[47] M. Takahashi. Statistical inference in missing data by mcmc and non-mcmc multiple imputation algorithms: Assessing the effects of between-imputation iterations. *Data Science Journal*, 16(37):1–17, 2017. 3

[48] Niket Tandon, Aparna S. Varde, and Gerard de Melo. Commonsense knowledge in machine intelligence. *SIGMOD Rec.*, 46(4):49–52, Feb. 2018. 2

[49] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed problems*. W.H. Winston, 1977. 2

[50] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 3

[51] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998. 2

[52] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang. Deep networks for image super-resolution with sparse prior. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 370–378, 2015. 2

[53] Tao Yang, Georgios Arvanitidis, Dongmei Fu, Xiaogang Li, and Søren Hauberg. Geodesic clustering in deep generative models. *CoRR*, abs/1809.04747, 2018. 1

[54] Xitong Yang, Palghat Ramesh, Radha Chitta, Sriganesh Madhvanath, Edgar A Bernal, and Jiebo Luo. Deep mul-

timodal representation learning from temporal data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5447–5455, 2017. 1

[55] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: Missing data imputation using generative adversarial nets. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5689–5698, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 1, 2, 5

[56] Yuxin Zhang, Zuquan Zheng, and Roland Hu. Super resolution using segmentation-prior self-attention generative adversarial network, 2020. 2