However, this approach quickly becomes infeasible as it requires an exponential number of models with respect to the dimensionality of the input space. In this work, we propose several conditional transformations that handle arbitrary dimensionality in a principled manner. Our contributions are as follows. 1) We propose a novel extension of flow-based generative models to model the conditional distribution of arbitrary unobserved covariates in data imputation tasks. Our method is the first to develop invertible transformations that operate on an arbitrary set of covariates. 2) We strengthen a flow-based model by using a novel autoregressive conditional likelihood. 3) We propose a novel penalty to generate a single imputed "best guess" for models without an analytically available mean. 4) We run extensive empirical studies and show that achieves state-of-the-art performance for both missing feature imputation and image inpainting on benchmark real-world datasets.

1 Problem Formulation

Consider a real-valued distribution $p(x)$ over $\mathbb{R}^d$. We are interested in estimating the conditional distribution of *all* possible subsets of covariates $u \subseteq \{1, \dots, d\}$ conditioned on the remaining observed covariates $o = \{1, \dots, d\} \setminus u$. That is, we shall estimate $p(x_u \mid x_o)$ where $x_u \in \mathbb{R}^{|u|}$ and $x_o \in \mathbb{R}^{|o|}$, for all possible subsets u. For ease of notation, let $b \in \{0,1\}^d$ be a binary mask indicating which dimensions are observed. Furthermore, let $v[b]$ index dimensions v_i for which $b_i = 1$ for the bitmask b on a vector v. Thus, $x_o = x\,[b]$ denotes observed dimensions and $x_u = x\,[1-b]$ denotes unobserved dimensions. We also apply this index to matrices such that $W[b,b]$ indexes rows and then columns. Without loss of generality, conditionals may also be conditioned on the bitmask b, $p(x_u \mid x_o, b)$, and will be estimated with maximum log-likelihood estimation as described below. In addition, imputation tasks shall be accomplished by generating samples from the conditional distributions $p(x_u \mid x_o, b)$.

2 Background

builds on Transformation Autoregressive Networks (TANs) , a flow-based model that combines transformation of variables with autoregressive likelihoods. We expound on flow-based models and TANs below. The change of variable theorem (**??**) is the cornerstone of flow-based generative models, where q represents an invertible transformation that transforms covariates from input space $\mathcal{X}$ into a latent space $\mathcal{Z}$.

$$p_{\mathcal{X}}(x) = \left|\det \frac{dq}{dx}\right| p_{\mathcal{Z}}(q(x)) \tag{1}$$

In order to efficiently compute the determinant, the transformation q is often designed to have a diagonal or triangular Jacobian . Since this type of transformation is rather restrictive, the flow models are often composed of multiple transformations in a sequence to get a more flexible composite transformation, i.e. $q = q_m \circ q_{m-1} \circ \ldots \circ q_1$. Here, the covariates flow through a chain of transformations, substituting the last output variable as input for the next transformation. Typically, a flow-based model transforms the covariates to a latent space with a simple base distribution, like a standard Gaussian. However, TANs provides more flexibility by modeling the latent distribution with an autoregressive approach . This alters the earlier equation, in that $p_{\mathcal{Z}}(q(x))$ is now represented as the product of d conditional distributions.

$$p_{\mathcal{X}}(x) = \left|\det \frac{dq}{dx}\right| \prod_{i=1}^{d} p_{\mathcal{Z}}(z^i \mid z^{i-1}, ..., z^1) \tag{2}$$

Since flow models give the exact likelihood, they can be trained by directly optimizing the log likelihood. In addition, thanks to the invertibility of the transformations, one can draw samples by simply inverting the transformations over a set of samples from the latent space.

3 Methods

We now develop by constructing both conditional transformations of variables and autoregressive likelihoods that work with an arbitrary set of unobserved covariates. To deal with arbitrary dimensionality for conditioning covariates x_o, we define a zero imputing function $\phi(v; b)$ that imputes input vector v with zeros based on the specified binary mask b:

$$w = \phi(v; b), \quad w_i = \begin{cases} v_c, & b_i = 1 \\ 0, & b_i = 0 \end{cases}, \tag{3}$$

where $c = \sum_{j=1}^{i} b_j$ represents the cumulative sum over b. For example, $\phi(x_o; b)$ gives a d-dimensional vector with missing values imputed by zeros (See Fig. for an illustration.). Thus, we get a conditioning vector with fixed dimensionality. However, handling the arbitrary dimensionality of x_u requires further care, as discussed below. We first consider a conditional extension to the change of variable theorem:

$$p_{\mathcal{X}}(x_u \mid x_o, b) = \left|\det \frac{dq_{x_o,b}}{dx_u}\right| p_{\mathcal{Z}}(q_{x_o,b}(x_u) \mid x_o, b), \tag{4}$$

where $q_{x_o,b}$ is a transformation on the unobserved covariates x_u with respect to the observed covariates x_o and binary mask b as demonstrated in Fig.. However, the fact that x_u could have varying dimensionality and arbitrary missing dimensions makes it challenging to define $q_{x_o,b}$'s across different bitmasks b. One challenge comes from requiring the transformation to have adaptive outputs that can adapt to different dimensionality of x_u. Another challenge is that different missing patterns require the transformation to capture different dependencies. Since the missing pattern could be arbitrary, we require the transformation to learn a large range of possible dependencies. Aiming at solving those challenges, we propose several conditional transformations that leverage the conditioning information in x_o and b and can be adapted to x_u with arbitrary dimensionality. We describe them in detail below. For notation simplicity, we drop the subscripts $\mathcal{X}$ and $\mathcal{Z}$ in the following sections.

Affine Coupling Transformation Affine coupling is a commonly used flow transformation. Here, we derive a conditional extension to it. Just as in the unconditional counterparts , we divide the unobserved covariates x_u into two parts