
TzK Flow - Conditional Generative Model

Micha Livne (mlivne@cs.toronto.edu)

David Fleet (fleet@cs.toronto.edu)

Abstract

We introduce **TzK** (pronounced “task”), a conditional probability flow-based model that exploits attributes (*e.g.*, style, class membership, or other side information) in order to learn tight conditional prior around manifolds of the target observations. The model is trained via approximated ML, and offers efficient approximation of arbitrary data sample distributions (similar to GAN and flow-based ML), and stable training (similar to VAE and ML), while avoiding variational approximations. **TzK** exploits meta-data to facilitate a bottleneck, similar to autoencoders, thereby producing a low-dimensional representation. Unlike autoencoders, the bottleneck does not limit model expressiveness, similar to flow-based ML. Supervised, unsupervised, and semi-supervised learning are supported by replacing missing observations with samples from learned priors. We demonstrate **TzK** by training jointly on MNIST and Omniglot datasets with minimal preprocessing, and weak supervision, with results comparable to state-of-the-art.

1 Introduction

The quality of samples produced by latent variable generative models continues to improve [5, 10]. Among the different approaches, variational auto-encoders (VAE) use a variational approximation to the posterior distribution over latent states given observations [11, 12, 18]. One can also formulate auto-encoders with an adversarial loss, effectively replacing KL divergence in VAE with a discriminator [7, 13, 14]. More recently, flow-based models have produced striking results [10]. They learn invertible mappings between observations and latent states using maximum likelihood (ML) [4–6, 10], but invertibility entails computation of the determinant of the Jacobian, which is $\mathcal{O}(D^3)$ in the general case for D -dimensional data. By restricting model architecture so the Jacobian is triangular/diagonal, the computational cost becomes manageable (*e.g.*, by partitioning input dimensions [6, 12] or with autoregressive models [15, 16]). Continuous normalizing flows [2, 8] trade such memory-intensive flow-based models, with bounded run-time, for calculation-intensive models with bounded memory requirements (solving invertible ODE problem).

Probability flow-based models allows direct parameter estimation via maximum likelihood (ML). Inspired by such models, we introduce **TzK** (pronounced “task”), a conditional probability flow-based model that exploits attributes (*e.g.*, style, class membership, or other side information) in order to learn tight conditional priors around manifolds of the target observations. A **TzK** model assumes multiple independent attributes (“knowledge”), so new attributes can be added on-line to an existing model. Compared with models trained with variational inference (VI), **TzK** scales well to high-dimensional data, without limiting the expressiveness of the model (via a bottleneck), much like flow-based ML parameter estimation. **TzK** also presents a unified model for supervised and unsupervised training, by replacing missing observations with samples from current learned priors. Compared with GAN-based training [7, 9], **TzK** offers stable training, which benefits from optimal (converged) discriminators, without the ad hoc use of regularization in GAN [1]. We demonstrate **TzK** by training jointly on multiple datasets with minimal preprocessing, and weak supervision, while maintaining the ability to sample from each dataset independently.

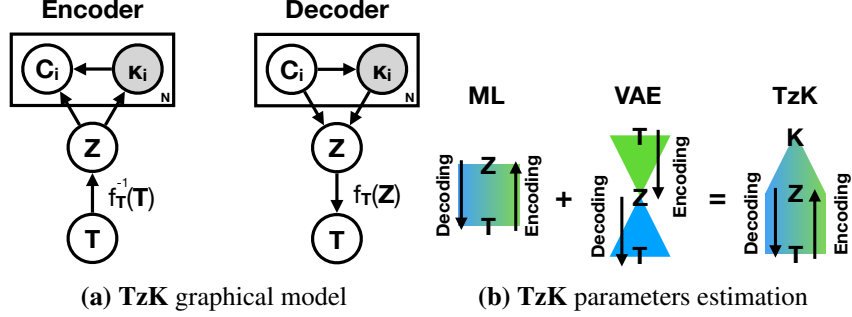


Figure 1: (a) **TzK** is formulated via a cycle consistent dual encoder/decoder representation of an abstract joint distribution over observations and knowledge types (*e.g.* class label). (b) Comparison with VAE and flow-based ML parameter estimation. Colors represents **encoding**, and **decoding**. **TzK** parameter estimation differs from VAE by placing the bottleneck as a conditional prior $p(T|K^i)$, without limiting the model expressiveness w.r.t. T . As a result, **TzK** can be optimized directly (similar to ML), and does not require the optimization of a lower bound over an approximated encoder. **TzK** differs from ML by incorporating bottlenecks (*i.e.*, conditional priors) to capture manifolds in \mathcal{T} , and thus can be used for semantic dimensionality reduction, like VAE. The use of flow $T = f_T(Z)$ allows parameter sharing between the various $p(T|K^i)$ and captures the similarity between the different knowledge types.

2 TzK Formulation

A **TzK** model comprises an observation $T \in \mathcal{T}$, a random variable with probability flow of a corresponding latent variable $Z \in \mathbb{R}^D$ [18]. Here, Z is mapped to T through a smooth invertible mapping $f_T : \mathbb{R}^D \rightarrow \mathbb{R}^D$ s.t. $T = f_T(Z)$. f_T transforms a given distribution $p(Z)$ (*e.g.*, Logistic or Normal) to the task distribution $p(T) = p(Z) \left| \det \frac{\partial f_T}{\partial Z} \right|^{-1}$. When f_T is not expressive enough, it is unlikely that the probability density of observed $f_T^{-1}(T)$ will match $p(Z)$. As a result, sufficiently expressive models tend to be memory-intensive, with slow training. However, in many cases we would like to sample from a specific manifold of T (*e.g.*, associated with a style attribute or class label). We therefore learn distributions over T conditioned on *knowledge* (or side-information), which also acts as a bottleneck. We define *knowledge* of type i as a joint discrete/continuous random variable $K^i = (\kappa^i \in \{0, 1\}, C^i \in \mathbb{R}^{C^i})$. κ^i is a latent binary variable, where $\kappa^i = 1$ indicates the presence of knowledge i for a corresponding observation T , and $p(\kappa^i = 1|\cdot) \in [0, \dots, 1]$ is a discriminator. C^i denotes knowledge characteristics, and is observed. K^i specifies a semantic prior over the observation space \mathcal{T} . We define all knowledge types to be independent (Fig. 1a). This design choice enables support in unknown number and type of knowledge, allows knowledge overlap (*i.e.*, multiple K^i can exist in the same sample T , as opposed to supervised 1-hot class label vector), and facilitates on-line acquisition of new knowledge types (*e.g.*, classes).

Example : $T \in \mathbb{R}^{1024}$ is a 32×32 image, the knowledge $i = \text{“A”}$ denotes the presence of letter “A” in an image ($\kappa^i = 1$), and $C^i \in \mathbb{R}$ represents the associated style.

Motivation: A **TzK** model facilitates a joint representation of conditional generators, without the requirement for a-priori knowledge about the nature of the condition which is typically required (*e.g.*, number of classes in a conditional VAE). It supports training of multiple labels per observation, while still maintaining the ability to sample from each label independently. The model allows one to incrementally add new side-information (knowledge) in an on-line fashion without having to retrain from scratch with a fixed number of label categories for example.

Formulation : A **TzK** model is defined via a cycle consistent dual encoder/decoder representation of an abstract joint distribution over observations \mathbf{T} and knowledge type \mathbf{K}^i (see Fig. 1a).

$$p_{\phi^i}^{enc}(\mathbf{T}, \mathbf{K}^i) = p_{\phi}^{enc}(\mathbf{K}^i|\mathbf{T}) p_{\phi}^{enc}(\mathbf{T}) = p_{\phi}^{enc}(\mathcal{C}^i|\mathbf{T}, \boldsymbol{\kappa}^i) p_{\phi}^{enc}(\boldsymbol{\kappa}^i|\mathbf{T}) p_{\phi}^{enc}(\mathbf{T}) \quad (1)$$

$$p_{\psi^i}^{dec}(\mathbf{T}, \mathbf{K}^i) = p_{\psi}^{dec}(\mathbf{T}|\mathbf{K}^i) p_{\psi}^{dec}(\mathbf{K}^i) = p_{\psi}^{dec}(\mathbf{T}|\mathcal{C}^i, \boldsymbol{\kappa}^i) p_{\psi}^{dec}(\boldsymbol{\kappa}^i|\mathcal{C}^i) p_{\psi}^{dec}(\mathcal{C}^i) \quad (2)$$

where $p_{\phi}^{enc}(\boldsymbol{\kappa}^i = 1|\mathbf{T})$, $p_{\psi}^{dec}(\boldsymbol{\kappa}^i = 1|\mathcal{C}^i)$ are discriminators of binary variable $\boldsymbol{\kappa}^i$, and \mathbf{Z} is marginalized out exploiting the probability flow deterministic mapping.

If one were to learn encoders and decoders for each knowledge type independently, then one fails to capture similarities between different kinds of knowledge. For instance, in MNIST dataset the handwritten digits “1” and “7” can have rather similar images. However, capturing such similarity is hard, and typically requires to know in advance the number and relationships of the different kinds of knowledge. In other words, combining multiple independent generators requires joint training, and thus is typically embedded into the model (*e.g.*, the number of classes has to be known a-priori in a multi-class classification model).

We would like to allow on-line acquisition of unknown number of knowledge types. In addition, we would like an explicit formulation for a decoder (generator) which is conditioned on an arbitrary subset of knowledge types (*i.e.*, compositionality between knowledge types). To achieve that, we propose the following formulation for a **TzK** objective. First, we define the joint encoder and decoder likelihoods over \mathbf{T} , and multiple knowledge types, $\bar{\mathbf{K}} = \{\mathbf{K}^i\}_{i=1}^N$, as follows:

$$p_{\phi}^{enc}(\mathbf{T}, \bar{\mathbf{K}}) = p(\bar{\mathbf{K}}|\mathbf{T}) p(\mathbf{T}) \quad (3)$$

$$p_{\psi}^{dec}(\mathbf{T}, \bar{\mathbf{K}}) = p(\mathbf{T}|\bar{\mathbf{K}}) p(\bar{\mathbf{K}}) \quad (4)$$

where ϕ and ψ are model parameters. Finally, $p(\bar{\mathbf{K}})$, $p(\bar{\mathbf{K}}|\mathbf{T})$ are further factorized with Bayes’ rule into individual mappings for different knowledge types,

$$p_{\phi}^{enc}(\mathbf{T}, \bar{\mathbf{K}}) = p_{\phi}^{enc}(\mathbf{T}) \prod_i (p_{\phi}^{enc}(\mathcal{C}^i|\mathbf{T}, \boldsymbol{\kappa}^i) p_{\phi}^{enc}(\boldsymbol{\kappa}^i|\mathbf{T})) \quad (5)$$

$$p_{\psi}^{dec}(\mathbf{T}, \bar{\mathbf{K}}) = \frac{1}{p_{\phi}^{enc}(\mathbf{T})^{N-1}} \prod_i (p_{\psi}^{dec}(\mathbf{T}|\mathcal{C}^i, \boldsymbol{\kappa}^i) p_{\psi}^{dec}(\boldsymbol{\kappa}^i|\mathcal{C}^i) p_{\psi}^{dec}(\mathcal{C}^i)) \quad (6)$$

with the assumption that knowledge types are independent given \mathbf{T} , and have independent priors.

Next, we would like to capture similarity between all existing knowledge types. We do that by combining p_{ϕ}^{enc} and p_{ψ}^{dec} into a single joint likelihood with $\theta = \{\phi, \psi\}$:

$$\mathcal{M}_{\theta}(\bar{\mathbf{K}}, \mathbf{T}) = [p_{\phi}^{enc}(\mathbf{T}, \bar{\mathbf{K}}) p_{\psi}^{dec}(\mathbf{T}, \bar{\mathbf{K}})]^{\frac{1}{2}} \quad (7)$$

Of course, for \mathcal{M}_{θ} to be a valid distribution, p_{ϕ}^{enc} and p_{ψ}^{dec} must represent the same joint distribution. This condition is enforced with the inclusion of a constraint that the KL divergence $\mathcal{D}_{\text{KL}}(p_{\psi}^{dec}(\mathbf{T}, \mathbf{K}^i) \| p_{\phi}^{enc}(\mathbf{T}, \mathbf{K}^i))$ for all knowledge type i should be 0. When the constraint holds for all i , the joint encoder/decoder distributions are the same, and $p(\mathbf{T})$ accurately accounts for similarity between multiple kinds of knowledge, since

$$p_{\phi}^{enc}(\mathbf{T}) = \mathbb{E}_{\hat{\mathbf{K}} \sim p_{\psi}^{dec}(\hat{\mathbf{K}})} [p_{\psi}^{dec}(\mathbf{T}|\hat{\mathbf{K}})] \quad (8)$$

for any subset $\hat{\mathbf{K}} \in \bar{\mathbf{K}}$ of existing knowledge types $\bar{\mathbf{K}}$. We refer to \mathcal{M}_{θ} as the *knowledge consistency* of the model to emphasize the joint capability of the model to “understand” (encode) and “express” (decode). We refer to the constraint the *knowledge gap*,

$$g_{\mathcal{M}_{\theta}}^i(\theta) \equiv \mathcal{D}_{\text{KL}}(p_{\psi}^{dec}(\mathbf{T}, \mathbf{K}^i) \| p_{\phi}^{enc}(\mathbf{T}, \mathbf{K}^i)) = \mathbb{E}_{\mathbf{T}, \mathbf{K}^i \sim p_{\psi}^{dec}(\mathbf{T}, \mathbf{K}^i)} \left[\log \frac{p_{\psi}^{dec}(\mathbf{T}, \mathbf{K}^i)}{p_{\phi}^{enc}(\mathbf{T}, \mathbf{K}^i)} \right] \quad (9)$$

as it represents validity of the assumption that the knowledge consistency $\mathcal{M}_\theta(\bar{\mathbf{K}}, \mathbf{T})$ is a distribution. When the knowledge gap $g_{\mathcal{M}_\theta}^i$ is 0, the model allows accurate (as opposed to approximated) compositionality of all possible subsets of acquired knowledge types. A larger knowledge gap represents approximated compositionality, where interpolating (marginalizing) over multiple knowledge types is inaccurate. Investigating compositionality is out of the scope of this paper. The flow $\mathbf{T} = f_{\mathbf{T}}(\mathbf{Z})$ resembles the approximated encoder in VAE, and in fact can be any valid flow, including identity. In such a case, **TzK** operates similarly to a conditional VAE (CVAE). When we have no knowledge, the knowledge consistency \mathcal{M}_θ is exactly $p(\mathbf{T})$.

Objective : Taking expectation over $\log \mathcal{M}_\theta(\bar{\mathbf{K}}, \mathbf{T})$ w.r.t. $\mathbf{T}, \bar{\mathbf{K}}$, we obtain the *observed knowledge consistency*:

$$\mathbb{E}_{\bar{\mathbf{K}}, \mathbf{T}} [\log(\mathcal{M}_\theta(\bar{\mathbf{K}}, \mathbf{T}))] = -H(\mathbf{T}) + \frac{1}{2} \sum_i I(\mathbf{Z}; \mathbf{K}^i) - H(\mathbf{K}^i) - \mathbb{E}_{\mathbf{T}} [H(\mathbf{K}^i | \mathbf{T})] \quad (10)$$

where $H(\cdot)$ denotes entropy, and $I(\cdot; \cdot)$ is mutual information.

Optimization: We optimize θ by maximizing Eq. (10) subject to the constraint $p_\phi^{enc} = p_\psi^{dec}$:

$$\max_{\theta} \mathbb{E}_{\bar{\mathbf{K}}, \mathbf{T}} [\log(\mathcal{M}_\theta(\bar{\mathbf{K}}, \mathbf{T}))] - \sum_i \lambda_i \cdot g_{\mathcal{M}_\theta}^i(\theta) \quad (11)$$

where λ_i are Lagrange multipliers, θ are the parameters (weights) in \mathcal{M}_θ , and $g_{\mathcal{M}_\theta}^i$ is defined in Eq. (9). Even when the knowledge gap constraint in Eq. (9) does not hold, its presence encourages the model to capture inter-knowledge similarity. In that case, the value of the knowledge gap can be used as an indicator of the violation of the assumption $p_\phi^{enc} = p_\psi^{dec}$.

Training: In practice, we approximate expectations w.r.t. $\mathbf{T}, \{\mathbf{C}^i\}_{i=1}^N$ with one sample, which combined with the reparametrization trick and Monte Carlo (MC) approximation yields an unbiased low variance gradient estimator [19]. Note that we take exact expectation w.r.t. binary variables $\{\kappa^i\}_{i=1}^N$. **TzK** support a unified model for supervised, semi-supervised, and unsupervised training. We use supervised samples for MC expectation approximation over continuous variables $\mathbf{T}, \{\mathbf{C}^i\}_{i=1}^N$. When an observation is missing (e.g., unsupervised \mathbf{C}^i), we sample from the corresponding conditional prior (e.g., $p(\mathbf{C}^i | \mathbf{T}, \kappa^i)$). Semi-supervised training occurs when \mathbf{C}^i is never given externally, while $p(\kappa^i | \mathbf{T})$ is. Similarly, this procedure works for \mathbf{T} , where we sample from the corresponding $p(\mathbf{T} | \mathbf{C}^i, \kappa^i)$. In such cases we “freeze” the discriminators $p(\kappa^i = 1 | \cdot)$, similar to GAN training, effectively exploiting knowledge already encoded in the discriminators. Full derivation of the procedure requires more space than is available here.

3 Experiments

We demonstrate **TzK** by training jointly on MNIST and Omniglot datasets (see Fig. 3 and Fig. 4).

Architecture: Priors are Gaussian, and conditional priors have additional 4 layers of flow from a conditional Gaussian with NN regressors for mean and variance (4 linear layers with 70 units and PReLU). We used $L = 10$ layers of flow step ($K = 1$) with tiling convolution of size 16×16 as described in Fig. 2. We used ADAM with $lr = 1e - 4$ for optimization. **Experiments**: We assign knowledge types per dataset (dataset conditional), and per MNIST label (digit conditional). We set $\mathbf{C}^i \in \mathbb{R}^2; \forall i$ to be unsupervised. **Preprocessing**: We dequantized pixel values for both datasets [17, 20], as described in [5], and padded MNIST to be 32×32 . **Contributions**: We train a single model that allows us to turn a weak prior generative model into a powerful conditional generative model (Fig. 3a). Compared with [5, 6, 16], we did not have to encode a-priori the conditionals (number of classes) into the architecture, allowing **TzK** to jointly train multiple conditional generative models, or incorporate new conditional generative models to an existing **TzK** model in an on-line fashion.

References

- [1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. 2017.

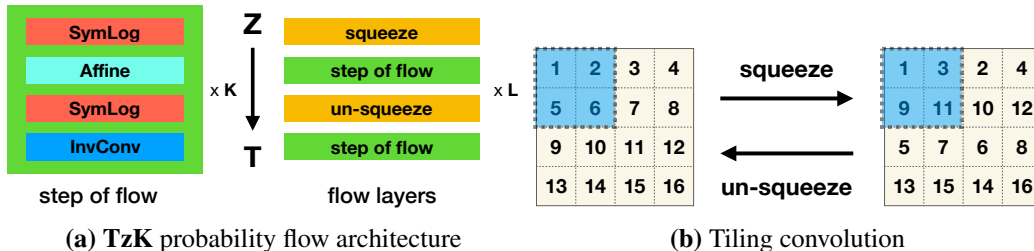
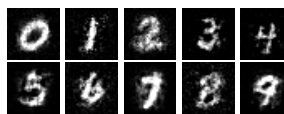


Figure 2: **TzK** architecture. **(a)** We used channel-wise invertible convolution [10], element-wise affine transformation [4], and $SymLog(x) = \log(|x| + 1) \cdot sign(x)$ as invertible activation. In order to propagate information between all dimensions we used squeeze/un-squeeze similar to [5]. **(b)** Inspired by the versatility of CNN ([3, 10]), we tiled the invertible convolution and alternated between squeeze and un-squeeze operations, allowing for reduced memory requirements.

	MNIST	Omniglot
TzK (prior)	5.14	5.14
TzK (dataset conditional) ‡	3.56	3.75
TzK (digit conditional) †	2.34	-
NICE	4.39	-
MADE MoG (conditional) [16] †	1.39	-
MAF MoG (5) (conditional) [16] †	1.51	-
Real NVP (5) [5] †	1.94	-



(a) Bits per pixel for conditional probability density estimation (lower is better) [16]. Label supervision is marked with †, and dataset supervision with ‡.

(b) Samples from digit conditional priors.

Figure 3: Experimental results with unsupervised $\mathcal{C}^i \in \mathbb{R}^2$. **(a)** Negative log-likelihood bits per dimension, with (prior) refers to a non-conditional prior, (dataset conditional) refers to priors conditioned on dataset (MNIST or Omniglot), and (digit conditional) refers to priors conditioned on MNIST digit label (0-9). All other results are taken from [16]. **(b)** The digit conditional priors allows us to directly label sample from manifold of interest in \mathcal{T} . **TzK** formulation allows us to simultaneously train multiple conditional priors without hard-coding the number of priors (e.g., classes) in the architecture. See additional results in Fig. 4.

[2] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural Ordinary Differential Equations. pages 1–18, 2018.

[3] J S Denker, W R Gardner, H P Graf, D Henderson, R E Howard, W Hubbard, L D Jackel, H S Baird, and I Guyon. NEURAL NETWORK RECOGNIZER FOR HAND-WRITTEN ZIP CODE DIGITS. Technical report.

[4] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear Independent Components Estimation. 1(2):1–13, 2014.

[5] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. 2016.

[6] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. MADE: Masked Autoencoder for Distribution Estimation. 37, 2015.

[7] 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.

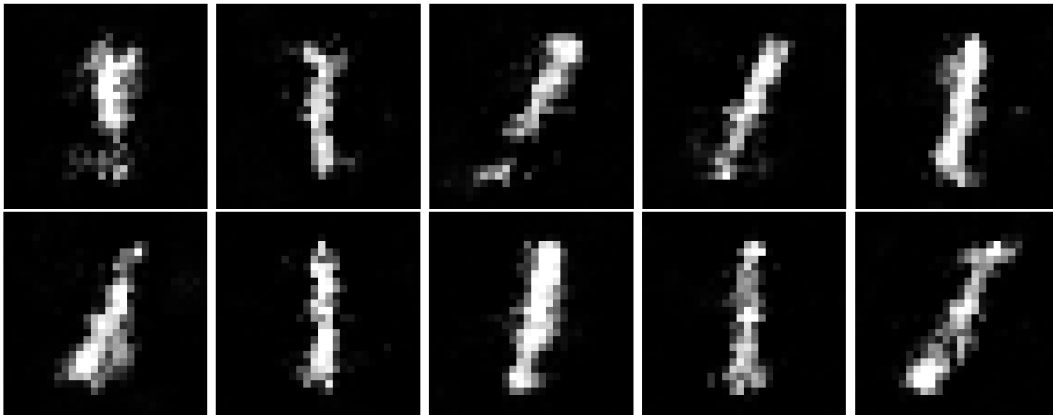
[8] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFLORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models. oct 2018.

[9] Aditya Grover, Manik Dhar, and Stefano Ermon. Flow-GAN: Combining Maximum Likelihood and Adversarial Learning in Generative Models. 2017.

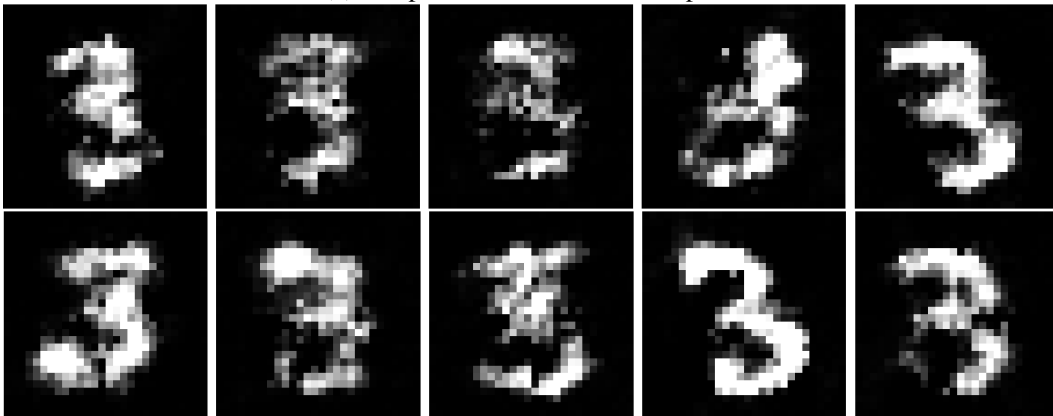
[10] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. pages 1–15, 2018.

- [11] Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving Variational Inference with Inverse Autoregressive Flow. (Nips), 2016.
- [12] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. (MI):1–14, 2013.
- [13] Alireza Makhzani. Implicit Autoencoders. 2018.
- [14] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial Autoencoders. 2015.
- [15] Junier B. Oliva, Avinava Dubey, Manzil Zaheer, Barnabás Póczos, Ruslan Salakhutdinov, Eric P. Xing, and Jeff Schneider. Transformation Autoregressive Networks. 2018.
- [16] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked Autoregressive Flow for Density Estimation. pages 1–17, 2017.
- [17] O. Peczenik and M. Zei. RNADE: The real-valued neural autoregressive density-estimator Benigno, jun 1954.
- [18] Danilo Jimenez Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. 37, 2015.
- [19] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. jan 2014.
- [20] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *Journal of the American Chemical Society*, 136(33):11757–11766, nov 2015.

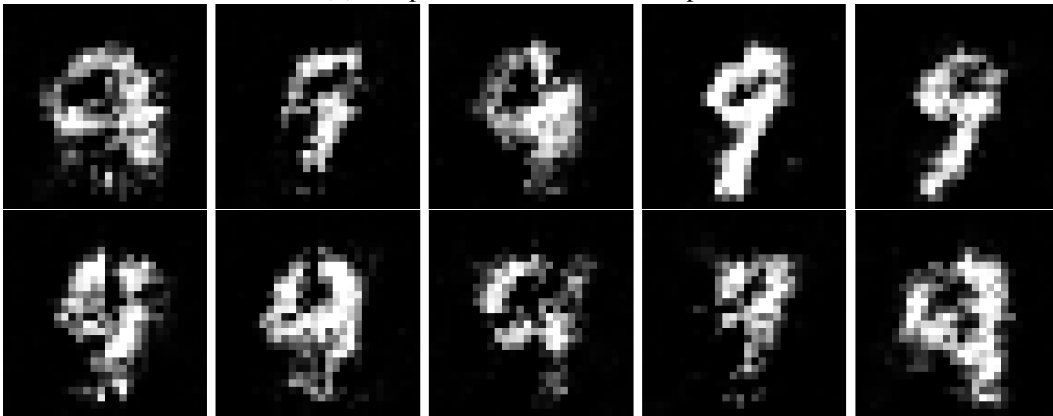
A Appendix



(a) Samples from “1” conditional prior



(b) Samples from “3” conditional prior



(c) Samples from “9” conditional prior

Figure 4: Additional samples from MNIST label conditional priors with $\mathcal{C}^i \in \mathbb{R}^2$.