

CS886: Theory and Applications of Generative Models

Lec 04: Discrete

Yaoliang Yu



UNIVERSITY OF
WATERLOO

FACULTY OF MATHEMATICS
**DAVID R. CHERITON SCHOOL
OF COMPUTER SCIENCE**

October 5, 2026

Recap: MLE

- Given training data $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \sim q(\mathbf{x})$, the **data density**
- Parameterize $p_{\theta}(\mathbf{x})$, the **model density** through **push-forward**:

Theorem: Representation through push-forward

Let r be any **continuous** distribution on $\mathbb{R}^h$. For **any** distribution p on $\mathbb{R}^d$, there exist **push-forward** maps $\mathbf{T} : \mathbb{R}^h \rightarrow \mathbb{R}^d$ such that $Z \sim r \implies \mathbf{T}(Z) \sim p$.

- Estimate θ by minimizing some “distance”:

$$\min_{\theta} \text{KL}(q \| p_{\theta}) \quad \equiv \quad \int -\log p_{\theta}(\mathbf{x}) \cdot q(\mathbf{x}) \, d\mathbf{x} \quad \approx \quad -\frac{1}{n} \sum_{i=1}^n \log p_{\theta}(\mathbf{x}_i)$$

- **Need a training sample from q and an explicit form of p_{θ}**

Change-of-variable Formula

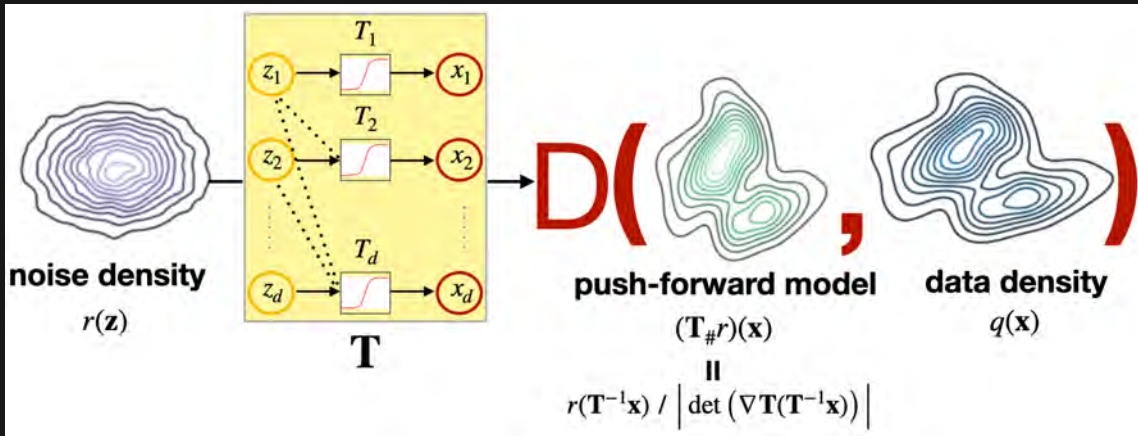
Theorem: Push-forward as change-of-variable

Let r be any **continuous** distribution on $\mathbb{R}^d$. If the **push-forward** map $\mathbf{T} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is **invertible**, then the density of $\mathbf{X} := \mathbf{T}(\mathbf{Z})$ is

$$p(\mathbf{x}) = r(\mathbf{T}^{-1}\mathbf{x}) \cdot |\det(\nabla \mathbf{T}^{-1}\mathbf{x})| = r(\mathbf{T}^{-1}\mathbf{x}) / |\det(\nabla \mathbf{T}(\mathbf{T}^{-1}\mathbf{x}))|$$

- Roughly, $p(\mathbf{x}) d\mathbf{x} = r(\mathbf{z}) d\mathbf{z}$: preservation of mass
- $\mathbf{T} \circ \mathbf{T}^{-1} = \text{Id} \implies \nabla \mathbf{T}(\mathbf{T}^{-1}) \cdot \nabla \mathbf{T}^{-1} = \text{Id}$ and $\det(A^{-1}) = 1/\det(A)$
 - $\mathbf{T} = (T_1, \dots, T_d) : \mathbb{R}^d \rightarrow \mathbb{R}^d$ and $\nabla \mathbf{T} = (\nabla T_1, \dots, \nabla T_d) : \mathbb{R}^d \rightarrow \mathbb{R}^d \otimes \mathbb{R}^d$
- **Input dim = output dim**
- From now on, use the notation $p = \mathbf{T}_{\#}r$ to denote the push-forward

MLE Revisited



$$\min_{T: \mathbb{R}^d \rightarrow \mathbb{R}^d} \text{KL}(q \parallel T_{\#}r) \approx \max_T \frac{1}{n} \sum_{i=1}^n \left[\log r(T^{-1}\mathbf{x}_i) - \log |\det \nabla T(T^{-1}\mathbf{x}_i)| \right]$$

Pick Your Poison

- Inverse in training, easy in sampling

$$\max_{\mathbf{T}} \frac{1}{n} \sum_{i=1}^n \left[\log r(\mathbf{T}^{-1} \mathbf{x}_i) - \log |\det \nabla \mathbf{T}(\mathbf{T}^{-1} \mathbf{x}_i)| \right]$$

- Easy in training, inverse in sampling

$$\max_{\mathbf{S}=\mathbf{T}^{-1}} \frac{1}{n} \sum_{i=1}^n \left[\log r(\mathbf{S} \mathbf{x}_i) + \log |\det \nabla \mathbf{S}(\mathbf{x}_i)| \right]$$

- Bottleneck in inverse $\mathbf{T}^{-1}$ and determinant $\det \nabla$
- Can apply GAN to avoid both inverse and determinant $\rightarrow$ minimax game

Increasing Triangular Map

$$x_1 = T_1(z_1)$$

$$x_2 = T_2(z_1, z_2)$$

$$\vdots$$

$$x_d = T_d(z_1, z_2, z_3, \dots, z_d)$$

$$\nabla \mathbf{T}(\mathbf{z}) = \begin{bmatrix} \frac{\partial T_1}{\partial z_1} & 0 & 0 & \dots & 0 \\ \frac{\partial T_2}{\partial z_1} & \frac{\partial T_2}{\partial z_2} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial T_d}{\partial z_1} & \frac{\partial T_d}{\partial z_2} & \frac{\partial T_d}{\partial z_3} & \dots & \frac{\partial T_d}{\partial z_d} \end{bmatrix}$$

- **Triangular:** j -th output x_j depends only on the first j inputs $z_1, \dots, z_j$
 - $\nabla \mathbf{T}$ is a (lower) triangular matrix
- **Increasing:** $T_j(z_1, \dots, z_{j-1}, z_j)$ is increasing w.r.t. z_j for any $z_1, \dots, z_{j-1}$
 - diagonal of ∇T is positive, i.e. $\frac{\partial T_j}{\partial z_j} > 0$

Practical Implications of Increasing Triangular Maps

- It is easy to compute the inverse $\mathbf{T}^{-1} : \mathbb{R}^d \rightarrow \mathbb{R}^d$
 - Fix x_1 , compute z_1 such that $T_1(z_1) = x_1$
 - T_1 is increasing w.r.t. z_1 , so **bisection** suffices
 - Fix x_j , compute z_j such that $T_j(z_1, \dots, z_{j-1}, z_j) = x_j$
 - T_j is increasing w.r.t. z_j while $z_1, \dots, z_{j-1}$ are already computed, so bisection suffices
- It is easy to compute the determinant $\det(\nabla \mathbf{T})$
 - $\nabla \mathbf{T}$ is triangular, so $\det = \text{product of the diagonal}$

Theoretical Implications of Increasing Triangular Maps

Theorem: Uniqueness for increasing triangular maps

For any two densities r and p on $\mathbb{R}^d$, there exists a **unique** (up to permutation) increasing triangular map $\mathbf{T}$ so that $p = \mathbf{T}_\# r$.

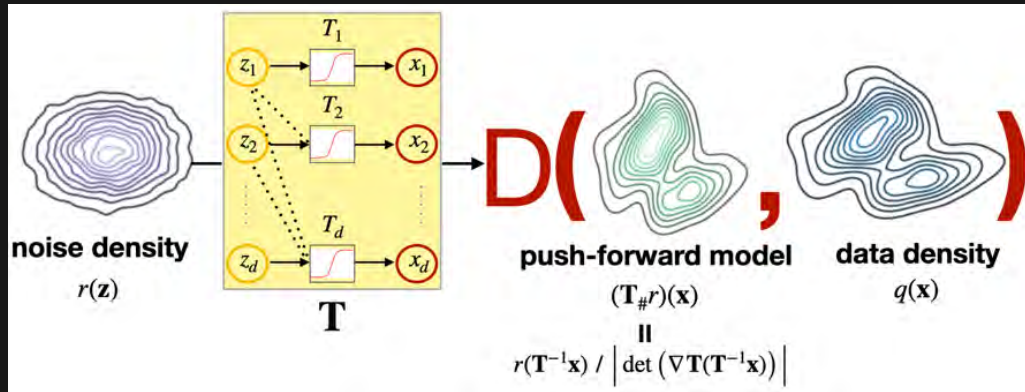
- In theory, **any** property of a **probabilistic** density is captured in the **deterministic** map $\mathbf{T}$, and vice versa!

Example: Linear transformation of Gaussian is Gaussian

Let $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \text{Id})$ and $L = \text{Cholesky}(S)$. Then, $\mathbf{x} = L\mathbf{n} + \boldsymbol{\mu} \sim \mathcal{N}(\boldsymbol{\mu}, S)$.

- We see now this is the only way, when restricted to increasing triangular maps!

MLE Revisited



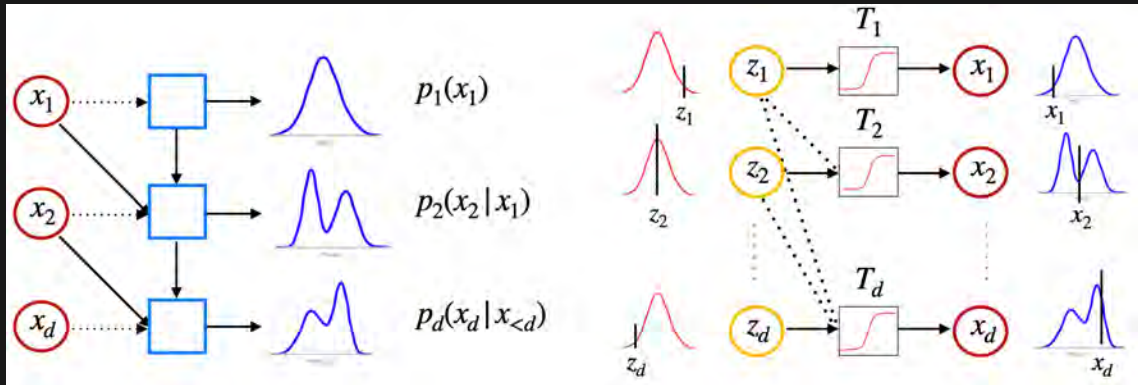
$$\min_{\mathbf{T}: \mathbb{R}^d \rightarrow \mathbb{R}^d} \text{KL}(q \parallel T_\# r) \approx \max_{\mathbf{T}} \frac{1}{n} \sum_{i=1}^n \left[\log r(\mathbf{T}^{-1} \mathbf{x}_i) - \sum_{j=1}^d \log \nabla_j T_j(\mathbf{T}^{-1} \mathbf{x}_i) \right]$$

P. Jaini, K. Selby, and Y. Yu. "Sum-of-squares Polynomial Flow". In: *International Conference on Machine Learning (ICML)*. 2019.

Y. Marzouk, T. Moselhy, M. Parno, and A. Spantini. "Sampling via Measure Transport: An Introduction". In: *Handbook of Uncertainty Quantification*. Ed. by R. Ghanem, D. Higdon, and H. Owahdi. Springer, 2016, pp. 1–41.

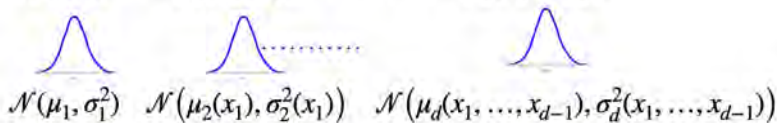
Autoregressive Models

$$p(\mathbf{x}) = \prod_{j=1}^d p_j(x_j | x_1, \dots, x_{j-1})$$

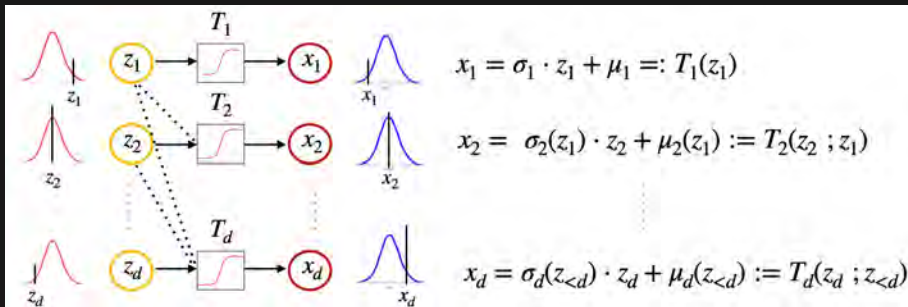


AR with Gaussian Conditionals

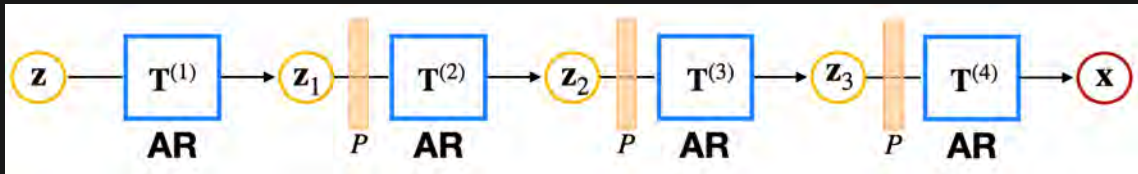
$$p(x) = p_1(x_1) \cdot p_2(x_2|x_1) \cdot \dots \cdot p_d(x_d|x_{<d})$$



$$\mathcal{N}(\mu_1, \sigma_1^2) \quad \mathcal{N}(\mu_2(x_1), \sigma_2^2(x_1)) \quad \mathcal{N}(\mu_d(x_1, \dots, x_{d-1}), \sigma_d^2(x_1, \dots, x_{d-1}))$$



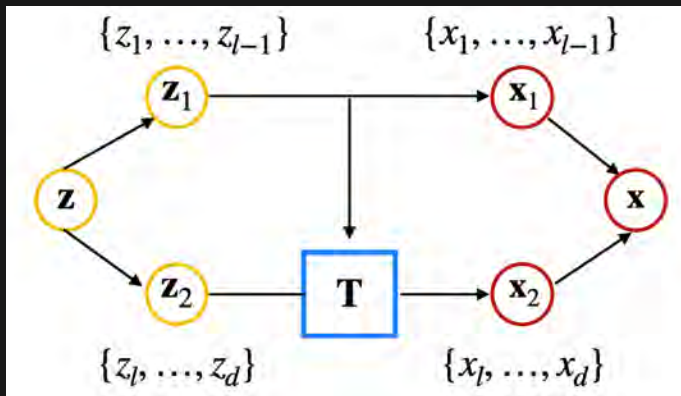
Masked AR Flows



$$(\mathbf{T}_{\#r})(\mathbf{x}) = r(\mathbf{z}) / \det(\nabla \mathbf{T}^{(1)} \mathbf{z}) / \det(\nabla \mathbf{T}^{(2)} \mathbf{z}_1) / \det(\nabla \mathbf{T}^{(3)} \mathbf{z}_2) / \det(\nabla \mathbf{T}^{(4)} \mathbf{z}_3)$$

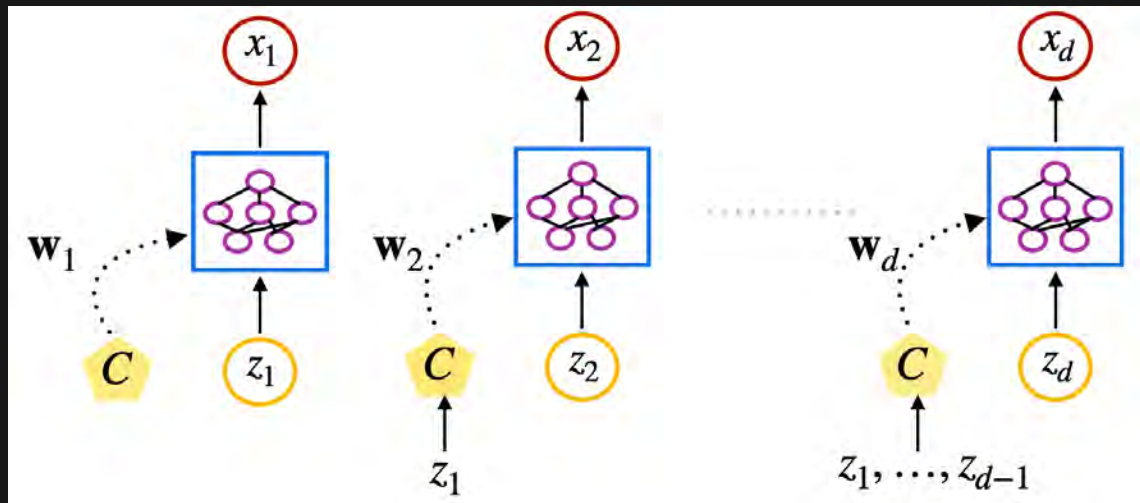
$$x_j = z_j \cdot \exp(\alpha_j(z_1, \dots, z_{j-1})) + \mu_j(z_1, \dots, z_{j-1}) =: T_j(z_1, \dots, z_{j-1}, z_j)$$

- Stack multiple layers to get deep
- What happens if the number of layers goes to ∞ ?

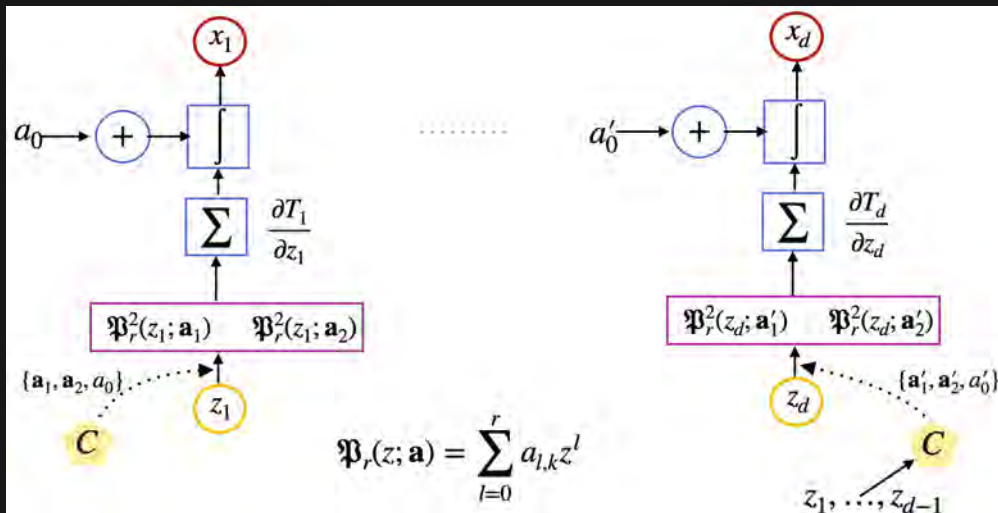


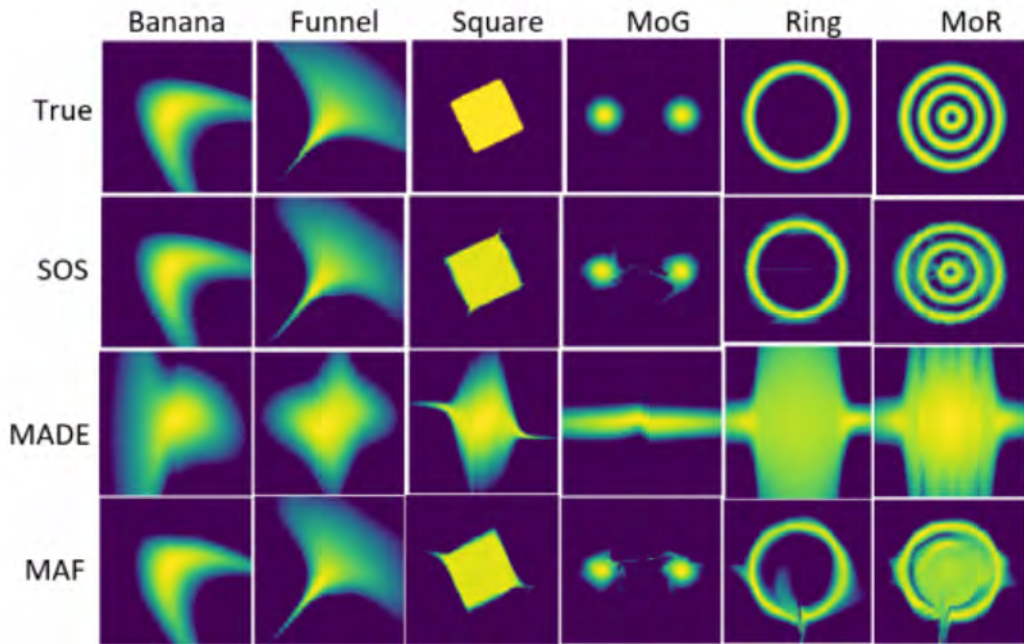
$$T_j(z_j ; z_1, \dots, z_{l-1}) = \exp \left(\alpha_j(z_1, \dots, z_{l-1}) \cdot \mathbf{1}_{j \notin [l-1]} \right) \cdot z_j + \mu_j(z_1, \dots, z_{l-1}) \cdot \mathbf{1}_{j \notin [l-1]}$$

Neural AR Flow



Sum-Of-Squares





Interpreting $\mathbf{T} =: \mathbf{Q}$

- $\mathbf{T} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ pushes the noise Z forward to observation X
- The inverse map $\mathbf{T}^{-1}$ pulls observation X back to noise Z

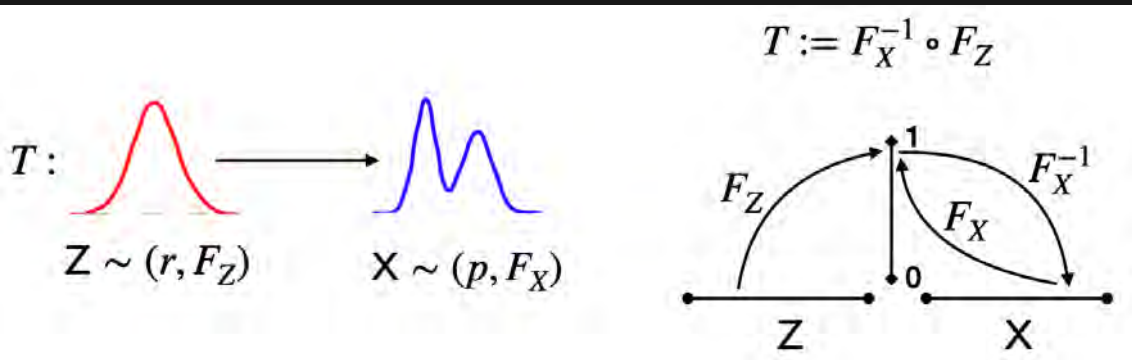
Theorem: Inverse sampling

Let $Z \sim \text{Uniform}(0, 1)$, F be the cdf of X , and $Q := F^{-1}$ is the **quantile function** of X . Then, $Q(Z) \sim F$.

$$\Pr(Q(Z) \leq x) = \Pr(Z \leq Q^{-1}x) = \Pr(Z \leq F(x)) = F(x)$$

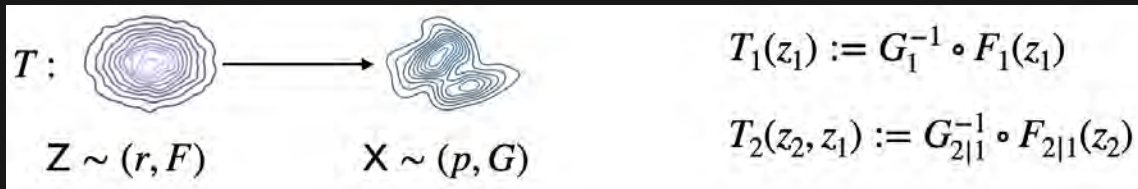
$\mathbf{Q} := \mathbf{T}$ serves as a multivariate generalization of the quantile function!

Univariate Increasing Rearrangement



- $Z \sim F_Z \implies F_Z(Z) \sim \text{Uniform}(0, 1)$
- $U \sim \text{Uniform}(0, 1) \implies F_X^{-1}(U) \sim F_X$
- $Z \sim F_Z \implies T(Z) \sim F_X$ where $T := F_X^{-1} \circ F_Z$ “rearranges” probability mass

Knothe-Rosenblatt Transform



- Iteratively apply univariate increasing rearrangement to the conditionals
- An early precursor of increasing triangular maps

