Deep Generative Models

Fall 2025

Sharif University of Technology

Alireza Mirrokni - 401106617

Problem Set 3

Problem 1: 1×1 Convolution in Glow

(a) Let the input tensor be $X \in \mathbb{R}^{C\times H\times W}$ and the output tensor be $Y \in \mathbb{R}^{C\times H\times W}$. The transformation is defined as:

$$Y_{:,i,j} = WX_{:,i,j}$$

where $W \in \mathbb{R}^{C\times C}$, applied independently to each spatial location (i, j) for $1 \leq i \leq H$ and $1 \leq j \leq W$.

Since the transformation is applied independently to each of the $H \times W$ spatial locations, we can view the total transformation as a block-diagonal operation acting on a flattened version of X. If we treat X and Y as vectors of size $C \cdot H \cdot W$, the total Jacobian matrix J of the transformation has the following block-diagonal structure:

$$J = \text{diag}(\underbrace{W, W, \ldots, W}_{H\cdot W \text{ times}})$$

The determinant of a block-diagonal matrix is the product of the determinants of its diagonal blocks. Therefore, the determinant of the Jacobian of the full mapping is:

$$\det(J) = \prod_{k=1}^{H\cdot W} \det(W) = (\det(W))^{H\cdot W}$$

The log-determinant of the Jacobian is:

$$\log|\det(J)| = \log\left|(\det(W))^{H\cdot W}\right| = H \cdot W \cdot \log|\det(W)|$$

The transformation is a linear mapping. A linear mapping represented by a square matrix (in this case, the total Jacobian J) is invertible if and only if its determinant is non-zero.

$$\det(J) \neq 0 \iff (\det(W))^{H\cdot W} \neq 0 \iff \det(W) \neq 0$$

Thus, the transformation is invertible if and only if $\det(W) \neq 0$.

Given that $\det(W) \neq 0$, the matrix W is invertible. We invert the relationship at each spatial location independently. Starting with the definition:

$$Y_{:,i,j} = WX_{:,i,j}$$

We multiply both sides by W^{-1} from the left:

$$W^{-1}Y_{:,i,j} = W^{-1}(WX_{:,i,j}) \implies W^{-1}Y_{:,i,j} = (W^{-1}W)X_{:,i,j} \implies W^{-1}Y_{:,i,j} = IX_{:,i,j}$$

Therefore, the explicit inverse transformation is:

$$X_{:,i,j} = W^{-1}Y_{:,i,j}.$$

(b) Let $\tilde{U} = U + \text{diag}(\mathbf{s})$. The matrix $\tilde{U}$ is an upper triangular matrix where the diagonal elements are exactly the elements of vector $\mathbf{s}$.

We compute the log-determinant of W:

$$\log|\det(W)| = \log|\det(PL\tilde{U})|.$$

Using the property that the determinant of a product is the product of determinants:

$$\det(W) = \det(P) \cdot \det(L) \cdot \det(\tilde{U}).$$

We evaluate each component:

1. $\det(P)$: Since P is a permutation matrix, its determinant is either $+1$ or -1. Thus, $|\det(P)| = 1$.
2. $\det(L)$: L is a triangular matrix, so its determinant is the product of its diagonal entries. Since the diagonal entries are all 1, $\det(L) = 1$.
3. $\det(\tilde{U})$: $\tilde{U}$ is an upper triangular matrix with diagonal entries $s_1, s_2, \dots, s_C$. Its determinant is the product of these diagonal entries: $\det(\tilde{U}) = \prod_{i=1}^{C} s_i$.

Substituting these back into the absolute determinant expression:

$$|\det(W)| = |(\pm 1) \cdot (1) \cdot (\prod_{i=1}^{C} s_i)| = \left| \prod_{i=1}^{C} s_i \right| = \prod_{i=1}^{C} |s_i|$$

Taking the logarithm:

$$\log|\det(W)| = \log\left(\prod_{i=1}^{C} |s_i|\right) = \sum_{i=1}^{C} \log|s_i|$$

This concludes the proof.

(c) To compute the inverse efficiently without explicitly computing the matrix inverse W^{-1}, we rely on the decomposition $W = PL\tilde{U}$, where $\tilde{U} = U + \text{diag}(\mathbf{s})$. We need to solve the linear system for the input vector x (where $x = X_{:,i,j}$) given the output vector y (where $y = Y_{:,i,j}$):

$$Wx = y \implies PL\tilde{U}x = y$$

We solve this system in three sequential steps:

1. **Inverse Permutation:** Multiply by P^{-1} (which is equal to $P^\top$) to remove the permutation matrix. Let $z_1 = L\tilde{U}x$.

   $$Pz_1 = y \implies z_1 = P^\top y$$

   Implementation: This is simply reordering the elements of vector y.
2. **Forward Substitution:** Solve for $z_2 = \tilde{U}x$ using the lower triangular matrix L.

   $$Lz_2 = z_1$$

   Since L is lower triangular with unit diagonals, we can solve for the components of z_2 iteratively starting from the first element. For the k-th element:

   $$(z_2)_k = (z_1)_k - \sum_{m=1}^{k-1} L_{k,m}(z_2)_m$$

3. **Backward Substitution:** Solve for x using the upper triangular matrix $\tilde{U}$.

$$\tilde{U}x = z_2$$

Since $\tilde{U}$ is upper triangular with diagonal elements s_i, we solve iteratively starting from the last element C down to 1. For the k-th element:

$$x_k = \frac{(z_2)_k - \sum_{m=k+1}^{C} \tilde{U}_{k,m} x_m}{s_k}$$

Computational Complexity:

We analyze the complexity for a single spatial location (per-pixel complexity) with channel dimension C:

- Permutation: Reordering a vector of size C takes $O(C)$ time.
- Forward Substitution: Solving a triangular system of size $C \times C$ requires approximately $C^2/2$ multiplications and additions. Complexity: $O(C^2)$.
- Backward Substitution: Solving a triangular system of size $C \times C$ also requires approximately $C^2/2$ operations. Complexity: $O(C^2)$.

The total computational cost per pixel is dominated by the triangular solves.

$$\text{Total Complexity} = O(C^2)$$

This is significantly more efficient than a general Gaussian elimination inversion ($O(C^3)$) if W were dense and unstructured, though applying a pre-computed dense inverse matrix W^{-1} would also be $O(C^2)$. However, the LU parameterization avoids the initial $O(C^3)$ cost of computing the inverse explicitly and calculating the determinant.

Problem 2: Theory of Continuous Normalizing Flows

(a.i) Let $z(0)$ be the random variable at time $t = 0$ and $z(t)$ be the variable at time t. The transformation from $z(0)$ to $z(t)$ is deterministic and invertible given the ODE dynamics. By the standard change of variables formula for probability densities:

$$p(z(t)) = p(z(0)) \left| \det \left(\frac{\partial z(t)}{\partial z(0)} \right) \right|^{-1}$$

Taking the natural logarithm of both sides:

$$\log p(z(t)) = \log p(z(0)) - \log \left| \det \left(\frac{\partial z(t)}{\partial z(0)} \right) \right|$$

Let $J(t) = \frac{\partial z(t)}{\partial z(0)}$ be the Jacobian of the mapping from $t = 0$ to t. Then:

$$\log p(z(t)) = \log p(z(0)) - \log |\det(J(t))|$$

(a.ii) We differentiate the expression from (a.i) with respect to t. Note that $z(0)$ is the initial condition and does not depend on t, so $\frac{d}{dt} \log p(z(0)) = 0$.

$$\frac{d}{dt} \log p(z(t)) = -\frac{d}{dt} \log |\det(J(t))|$$

(a.iii) First, we prove Jacobi's formula:

$$\frac{d}{dt}\log|\det(A(t))| = \operatorname{Tr}\left(A(t)^{-1}\frac{dA(t)}{dt}\right)$$

Proof: Recall the derivative of a determinant with respect to the matrix elements:

$$\frac{\partial \det(A)}{\partial A_{ij}} = \operatorname{adj}(A)_{ji}$$

Using the chain rule (define $\dot{A} = \frac{dA(t)}{dt}$):

$$\frac{d}{dt}\det(A(t)) = \sum_{i,j}\frac{\partial \det(A)}{\partial A_{ij}}\frac{dA_{ij}}{dt} = \sum_{i,j}\operatorname{adj}(A)_{ji}\frac{dA_{ij}}{dt} = \operatorname{Tr}(\operatorname{adj}(A)\dot{A})$$

Since $A^{-1} = \frac{1}{\det(A)}\operatorname{adj}(A)$, we have $\operatorname{adj}(A) = \det(A)A^{-1}$. Substituting this back:

$$\frac{d}{dt}\det(A(t)) = \det(A)\operatorname{Tr}(A^{-1}\dot{A})$$

Now consider the derivative of the log-determinant:

$$\frac{d}{dt}\log|\det(A(t))| = \frac{1}{\det(A(t))}\frac{d}{dt}\det(A(t)) = \frac{1}{\det(A)}\left(\det(A)\operatorname{Tr}(A^{-1}\dot{A})\right) = \operatorname{Tr}(A^{-1}\dot{A})$$

Applying this formula to our Jacobian term with $A(t) = J(t)$:

$$\frac{d}{dt}\log|\det(J(t))| = \operatorname{Tr}\left(J(t)^{-1}\frac{dJ(t)}{dt}\right)$$

Therefore:

$$\frac{d}{dt}\log p(z(t)) = -\operatorname{Tr}\left(J(t)^{-1}\frac{dJ(t)}{dt}\right)$$

(a.iv) We need to compute $\frac{dJ(t)}{dt}$. Recall $J(t) = \frac{\partial z(t)}{\partial z(0)}$. Since the partial derivatives commute (assuming smoothness):

$$\frac{dJ(t)}{dt} = \frac{d}{dt}\left(\frac{\partial z(t)}{\partial z(0)}\right) = \frac{\partial}{\partial z(0)}\left(\frac{dz(t)}{dt}\right)$$

Substituting the ODE definition $\frac{dz(t)}{dt} = f_\theta(z(t),t)$:

$$\frac{dJ(t)}{dt} = \frac{\partial}{\partial z(0)}f_\theta(z(t),t)$$

Using the chain rule for the derivative with respect to $z(0)$:

$$\frac{\partial f_\theta(z(t),t)}{\partial z(0)} = \frac{\partial f_\theta}{\partial z(t)}\frac{\partial z(t)}{\partial z(0)} = \frac{\partial f_\theta}{\partial z}J(t)$$

Thus, we have proven:

$$\frac{dJ(t)}{dt} = \frac{\partial f_\theta}{\partial z}J(t)$$

(a.v) Substitute the result from (a.iv) into the trace expression from (a.iii):

$$\frac{d}{dt}\log p(z(t)) = -\operatorname{Tr}\left(J(t)^{-1}\left(\frac{\partial f_\theta}{\partial z}J(t)\right)\right)$$

Using the cyclic property of the trace operator, $\operatorname{Tr}(ABC) = \operatorname{Tr}(BCA)$:

$$\operatorname{Tr}\left(J(t)^{-1}\frac{\partial f_\theta}{\partial z}J(t)\right) = \operatorname{Tr}\left(J(t)J(t)^{-1}\frac{\partial f_\theta}{\partial z}\right) = \operatorname{Tr}\left(I\frac{\partial f_\theta}{\partial z}\right) = \operatorname{Tr}\left(\frac{\partial f_\theta}{\partial z}\right)$$

This yields the final formula:

$$\frac{d}{dt}\log p(z(t)) = -\operatorname{Tr}\left(\frac{\partial f_\theta}{\partial z}\right)$$

Problem 3: Duality in Autoregressive Normalizing Flows

(a) **Single Layer Definitions:** Let $\mathbf{u} \in \mathbb{R}^D$ be the input (base/latent) variable and $\mathbf{x} \in \mathbb{R}^D$ be the output (data) variable. We define the transformation direction $f : \mathbf{u} \mapsto \mathbf{x}$ as the generative direction and $f^{-1} : \mathbf{x} \mapsto \mathbf{u}$ as the normalizing direction.

The Masked Autoregressive Flow (MAF) is defined such that the transformation from data $\mathbf{x}$ to noise $\mathbf{u}$ is autoregressive.

Inverse Transformation: This direction is efficient to compute because it is parallelizable.

$$u_i = \frac{x_i - \mu_i(\mathbf{x}_{1:i-1})}{\exp(\alpha_i(\mathbf{x}_{1:i-1}))} \quad \text{for } i = 1, \ldots, D$$

Here, μ_i and α_i are computed by a masked autoencoder (MADE) which takes $\mathbf{x}$ as input. Since all inputs $\mathbf{x}$ are known, all μ_i and α_i can be computed in a single forward pass of the neural network.

Forward Transformation: To sample $\mathbf{x}$ given $\mathbf{u}$, we must invert the equation above:

$$x_i = u_i \cdot \exp(\alpha_i(\mathbf{x}_{1:i-1})) + \mu_i(\mathbf{x}_{1:i-1})$$

Log-Density Expression: The log-likelihood of a data point $\mathbf{x}$ under a single MAF layer is given by the change of variables formula:

$$\log p_{\text{MAF}}(\mathbf{x}) = \log \pi_{\mathbf{u}}(f^{-1}(\mathbf{x})) + \log\left|\det \frac{\partial f^{-1}(\mathbf{x})}{\partial \mathbf{x}}\right|$$

The Jacobian $J = \frac{\partial \mathbf{u}}{\partial \mathbf{x}}$ is a lower-triangular matrix because u_i depends only on $x_{\le i}$. The diagonal entries are $\frac{\partial u_i}{\partial x_i} = \exp(-\alpha_i(\mathbf{x}_{1:i-1}))$. The determinant is the product of these diagonal entries.

$$\log|\det J| = \sum_{i=1}^{D} -\alpha_i(\mathbf{x}_{1:i-1})$$

Thus:

$$\log p_{\text{MAF}}(\mathbf{x}) = -\frac{D}{2}\log(2\pi) - \frac{1}{2}\|f^{-1}(\mathbf{x})\|^2 - \sum_{i=1}^{D}\alpha_i(\mathbf{x}_{1:i-1})$$

Stacked MAF: Consider a flow composed of K autoregressive layers. Let f_k denote the forward (generative) transformation of the k-th layer. The full generative transformation $f : \mathbf{u} \to \mathbf{x}$ is the composition:

$$\mathbf{x} = f(\mathbf{u}) = f_K \circ f_{K-1} \circ \cdots \circ f_1(\mathbf{u})$$

To evaluate the density of a data point $\mathbf{x}$, we invert this mapping to recover the latent representation $\mathbf{u}$. We define a sequence of intermediate variables $\mathbf{z}_0, \ldots, \mathbf{z}_K$, where $\mathbf{z}_K = \mathbf{x}$ (the data) and $\mathbf{z}_0 = \mathbf{u}$ (the base latent code). The inverse (normalizing) pass transforms the data recursively backwards through the layers:

$$\mathbf{z}_{k-1} = f_k^{-1}(\mathbf{z}_k) \quad \text{for } k = K, K-1, \ldots, 1$$

Consequently, the full inverse is $\mathbf{u} = \mathbf{z}_0 = f_1^{-1} \circ \cdots \circ f_K^{-1}(\mathbf{x})$.

The log-density $p_{\text{MAF}}(\mathbf{x})$ is derived using the change of variables formula. It consists of the log-density of the recovered base variable $\mathbf{u}$ plus the log-absolute-determinant of the Jacobian of the total inverse transformation:

$$\log p_{\text{MAF}}(\mathbf{x}) = \log \pi_{\mathbf{u}}(\mathbf{z}_0) + \log \left| \det \frac{\partial \mathbf{z}_0}{\partial \mathbf{z}_K} \right|$$

Applying the chain rule for Jacobians, the total Jacobian decomposes into the product of the Jacobians of each individual layer:

$$\frac{\partial \mathbf{z}_0}{\partial \mathbf{z}_K} = \prod_{k=1}^{K} \frac{\partial \mathbf{z}_{k-1}}{\partial \mathbf{z}_k} = \prod_{k=1}^{K} \frac{\partial f_k^{-1}(\mathbf{z}_k)}{\partial \mathbf{z}_k}$$

Since the log-determinant of a product is the sum of the log-determinants, we arrive at the final density expression:

$$\log p_{\text{MAF}}(\mathbf{x}) = \log \pi_{\mathbf{u}}(f^{-1}(\mathbf{x})) + \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k^{-1}(\mathbf{z}_k)}{\partial \mathbf{z}_k} \right|$$

Efficiency Analysis:

- *Fast Density Evaluation:* To evaluate $\log p(\mathbf{x})$, we map $\mathbf{x} \to \mathbf{u}$. In MAF, μ_i and α_i depend on $\mathbf{x}_{<i}$. Since $\mathbf{x}$ is fully observed during training/evaluation, we can compute all $\{\mu_i, \alpha_i\}_{i=1}^D$ in parallel using one pass of the masked network (MADE).
- *Slow Sampling:* To generate $\mathbf{x}$ from $\mathbf{u}$, we use the forward transformation. We must compute x_1, then use it to compute $\mu_2(x_1), \alpha_2(x_1)$ to get x_2, and so on. This introduces a sequential dependency of length D, requiring D passes through the network.

(b) **Definitions:** IAF is essentially the inverse of MAF. The autoregressive structure is placed on the latent variables $\mathbf{u}$ (or the input to the layer in the generative direction) rather than the output.

Forward Transformation: This is the fast direction for IAF.

$$x_i = u_i \cdot \exp(\alpha_i(\mathbf{u}_{1:i-1})) + \mu_i(\mathbf{u}_{1:i-1})$$

Here, the statistics μ and α depend on $\mathbf{u}$. Since $\mathbf{u}$ is sampled first (from the base density), all u_i are available simultaneously.

Inverse Transformation:

$$u_i = \frac{x_i - \mu_i(\mathbf{u}_{1:i-1})}{\exp(\alpha_i(\mathbf{u}_{1:i-1}))}$$

Here to compute u_i from x_i, we need $\mathbf{u}_{1:i-1}$. This makes the inverse sequential. Specifically, we find u_1 from x_1, then use u_1 to find u_2, etc.

Reversed Conditioning: The fundamental difference lies in the input domain of the autoregressive functions (implemented via masked neural networks). Let $\mathcal{M}_\theta$ be an autoregressive mapping such that the i-th output depends only on inputs with indices less than i.

MAF (Autoregression on Data): The shift and log-scale parameters for the i-th dimension are functions of the observed data prefix $\mathbf{x}_{1:i-1}$.

$$\mu_i^{\text{MAF}} = \mathcal{M}_{\theta,\mu}(\mathbf{x})_i = f(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}), \quad \alpha_i^{\text{MAF}} = \mathcal{M}_{\theta,\alpha}(\mathbf{x})_i = g(\mathbf{x}_1, \ldots, \mathbf{x}_{i-1})$$

Thus, MAF directly models the conditional factorization of the data density:

$$p(\mathbf{x}) = \prod_{i=1}^{D} p(x_i \mid \mathbf{x}_{1:i-1}), \quad \text{where } x_i \mid \mathbf{x}_{<i} \sim \mathcal{N}(\mu_i^{\text{MAF}}, \exp(2\alpha_i^{\text{MAF}}))$$

IAF (Autoregression on Latents): The conditioning is reversed such that the parameters depend on the latent noise prefix $\mathbf{u}_{1:i-1}$.

$$\mu_i^{\text{IAF}} = \mathcal{M}_{\phi,\mu}(\mathbf{u})_i = f(\mathbf{u}_1, \ldots, \mathbf{u}_{i-1}), \quad \alpha_i^{\text{IAF}} = \mathcal{M}_{\phi,\alpha}(\mathbf{u})_i = g(\mathbf{u}_1, \ldots, \mathbf{u}_{i-1})$$

Consequently, IAF does not explicitly model $p(x_i \mid \mathbf{x}_{<i})$ in closed form. Instead, it defines x_i via a parallelizable location-scale transformation of u_i conditioned on the history of the noise $\mathbf{u}_{<i}$.

Efficiency Analysis:

- *Fast Sampling:* Generating $\mathbf{x}$ from $\mathbf{u}$ is parallelizable because $\mu(\mathbf{u})$ and $\alpha(\mathbf{u})$ can be computed in one pass once $\mathbf{u}$ is sampled.
- *Slow Density Evaluation:* To evaluate density of a new data point $\mathbf{x}$, we must find its corresponding $\mathbf{u} = f^{-1}(\mathbf{x})$. Since u_i depends on $\mathbf{u}_{<i}$ via the autoregressive parameters, we must solve for u_1, then u_2, etc., sequentially. This requires D passes.

(c) We wish to prove:

$$D_{\text{KL}}(\pi_{\mathbf{x}}(\mathbf{x}) \| p_{\text{MAF}}(\mathbf{x})) = D_{\text{KL}}(p_{\mathbf{u}}(\mathbf{u}) \| \pi_{\mathbf{u}}(\mathbf{u}))$$

Let $f : \mathbf{u} \to \mathbf{x}$ be the transformation defined by the MAF model. The density $p_{\text{MAF}}(\mathbf{x})$ is defined by the pushforward of the base density $\pi_{\mathbf{u}}$ through f.

$$D_{\text{KL}}(\pi_{\mathbf{x}} \| p_{\text{MAF}}) = \mathbb{E}_{\mathbf{x} \sim \pi_{\mathbf{x}}} \left[\log \pi_{\mathbf{x}}(\mathbf{x}) - \log p_{\text{MAF}}(\mathbf{x})\right]$$

Using the change of variables formula for $p_{\text{MAF}}(\mathbf{x})$:

$$\log p_{\text{MAF}}(\mathbf{x}) = \log \pi_{\mathbf{u}}(f^{-1}(\mathbf{x})) + \log \left|\det J_{f^{-1}}(\mathbf{x})\right|$$

Substituting this into the expectation:

$$\text{LHS} = \mathbb{E}_{\mathbf{x}\sim\pi_{\mathbf{x}}} \left[\log \pi_{\mathbf{x}}(\mathbf{x}) - \log \pi_{\mathbf{u}}(f^{-1}(\mathbf{x})) - \log \left|\det J_{f^{-1}}(\mathbf{x})\right|\right]$$

In the RHS, $p_{\mathbf{u}}(\mathbf{u})$ is the implicit density induced on the latent space by passing the true data distribution $\pi_{\mathbf{x}}$ through the inverse map f^{-1}. That is, if $\mathbf{x} \sim \pi_{\mathbf{x}}$ and $\mathbf{u} = f^{-1}(\mathbf{x})$, then $\mathbf{u} \sim p_{\mathbf{u}}$. Using the change of variables formula for this induced density:

$$p_{\mathbf{u}}(\mathbf{u}) = \pi_{\mathbf{x}}(f(\mathbf{u}))\, |\det J_f(\mathbf{u})|$$

Taking the log:

$$\log p_{\mathbf{u}}(\mathbf{u}) = \log \pi_{\mathbf{x}}(f(\mathbf{u})) + \log |\det J_f(\mathbf{u})|$$

The KL divergence is:

$$D_{\text{KL}}(p_{\mathbf{u}} \| \pi_{\mathbf{u}}) = \mathbb{E}_{\mathbf{u}\sim p_{\mathbf{u}}} \left[\log p_{\mathbf{u}}(\mathbf{u}) - \log \pi_{\mathbf{u}}(\mathbf{u})\right]$$

By the Law of the Unconscious Statistician (LOTUS), we can compute the expectation over $\mathbf{u} \sim p_{\mathbf{u}}$ as an expectation over $\mathbf{x} \sim \pi_{\mathbf{x}}$ via the mapping $\mathbf{u} = f^{-1}(\mathbf{x})$. Substituting the expression for $\log p_{\mathbf{u}}(\mathbf{u})$ and changing the expectation variable:

$$\text{RHS} = \mathbb{E}_{\mathbf{x}\sim\pi_{\mathbf{x}}} \left[\left(\log \pi_{\mathbf{x}}(f(f^{-1}(\mathbf{x}))) + \log \left|\det J_f(f^{-1}(\mathbf{x}))\right|\right) - \log \pi_{\mathbf{u}}(f^{-1}(\mathbf{x}))\right]$$

Simplify terms: $f(f^{-1}(\mathbf{x})) = \mathbf{x}$. Also, by the inverse function theorem, the Jacobian determinant of the forward map is the reciprocal of the Jacobian determinant of the inverse map:

$$\det J_f(f^{-1}(\mathbf{x})) = \frac{1}{\det J_{f^{-1}}(\mathbf{x})} \implies \log |\det J_f| = -\log \left|\det J_{f^{-1}}\right|$$

Substituting these back into the RHS equation:

$$\text{RHS} = \mathbb{E}_{\mathbf{x}\sim\pi_{\mathbf{x}}} \left[\log \pi_{\mathbf{x}}(\mathbf{x}) - \log \left|\det J_{f^{-1}}(\mathbf{x})\right| - \log \pi_{\mathbf{u}}(f^{-1}(\mathbf{x}))\right]$$

Comparing the expanded LHS and RHS, we see they are identical.

Interpretation: The identity shows that training a MAF model via Maximum Likelihood Estimation (which minimizes $D_{\text{KL}}(\pi_{\mathbf{x}} \| p_{\text{MAF}})$) is mathematically equivalent to performing variational inference in the latent space.

Specifically, consider the data distribution $\pi_{\mathbf{x}}$ as a complex ”posterior” we wish to approximate with a base distribution $\pi_{\mathbf{u}}$. The transformation f^{-1} acts as a deterministic variational posterior (or inference network) that maps complex data $\mathbf{x}$ to the latent space $\mathbf{u}$. The term $D_{\text{KL}}(p_{\mathbf{u}} \| \pi_{\mathbf{u}})$ measures how close the mapped data distribution (the implicit density $p_{\mathbf{u}}$) is to the Gaussian prior $\pi_{\mathbf{u}}$.

Because the direction of the mapping in this variational view ($\mathbf{x} \to \mathbf{u}$) is autoregressive in $\mathbf{x}$, the implicit model behaves like an Inverse Autoregressive Flow (IAF) acting as an inference network. Thus, MAF training aligns the ”encoded” data distribution with the prior, effectively performing stochastic variational inference where the variational family is defined by an implicit IAF.

Problem 4: VAEs with Conditional Flows

(ai) We start from the joint KL objective

$$\mathcal{L} = \iint \tilde{p}(x)p(z \mid x) \log\left(\frac{\tilde{p}(x)p(z \mid x)}{q(x \mid z)q(z)}\right) dz\, dx$$

The conditional flow posterior is defined by

$$p(z \mid x) = \int \delta\big(z - F_x(u)\big)\mathcal{N}(u; 0, I)\, du$$

where for each fixed x, the map $F_x : \mathbb{R}^d \to \mathbb{R}^d$ is invertible with inverse $H_x = F_x^{-1}$. Substitute the posterior into $\mathcal{L}$:

$$\mathcal{L} = \int \tilde{p}(x) \left[\int \left(\int \delta\big(z - F_x(u)\big)\mathcal{N}(u; 0, I)\, du\right) \log\left(\frac{\tilde{p}(x)p(z \mid x)}{q(x \mid z)q(z)}\right) dz\right] dx$$

Swap the u and z integrals:

$$\mathcal{L} = \iint \tilde{p}(x)\mathcal{N}(u; 0, I) \left[\int \delta\big(z - F_x(u)\big) \log\left(\frac{\tilde{p}(x)p(z \mid x)}{q(x \mid z)q(z)}\right) dz\right] du\, dx$$

Now apply the shifting property of the Dirac delta in $\mathbb{R}^d$:

$$\int \delta(z - a)f(z)\, dz = f(a)$$

Here $a = F_x(u)$ and $f(z) = \log\left(\frac{\tilde{p}(x)p(z|x)}{q(x|z)q(z)}\right)$, so the inner z-integral becomes evaluation at $z = F_x(u)$:

$$\mathcal{L} = \iint \tilde{p}(x)\mathcal{N}(u; 0, I) \log\left(\frac{\tilde{p}(x)p(F_x(u) \mid x)}{q(x \mid F_x(u))q(F_x(u))}\right) du\, dx$$

It remains to express $p(F_x(u) \mid x)$ in terms of u and the Jacobian determinant of F_x. Using the same definition of $p(\cdot \mid x)$ and plugging in $z = F_x(u)$ gives

$$p(F_x(u) \mid x) = \int \delta\big(F_x(u) - F_x(u')\big)\mathcal{N}(u'; 0, I)\, du'$$

Because F_x is invertible, the equation $F_x(u') = F_x(u)$ has the unique solution

$$u' = H_x(F_x(u)) = u$$

We use the multidimensional delta change-of-variables rule: for an invertible differentiable map g,

$$\delta\big(g(u')\big) = \delta(u' - u'_0) \left|\det\left(\frac{\partial g}{\partial u'}(u'_0)\right)\right|^{-1} \quad \text{when } g(u'_0) = 0$$

Apply this with $g(u') = F_x(u) - F_x(u')$. Then

$$\frac{\partial g}{\partial u'}(u') = -\frac{\partial F_x(u')}{\partial u'}, \qquad \left|\det\left(\frac{\partial g}{\partial u'}(u')\right)\right| = \left|\det\left(\frac{\partial F_x(u')}{\partial u'}\right)\right|$$

so we obtain

$$\delta\big(F_x(u) - F_x(u')\big) = \delta\big(u' - H_x(F_x(u))\big) \left|\det\left(\frac{\partial F_x(u')}{\partial u'}\right)\right|^{-1}_{u'=H_x(F_x(u))}$$

Substitute this into the expression for $p(F_x(u) \mid x)$:

$$p(F_x(u) \mid x) = \int \delta\big(u' - H_x(F_x(u))\big) \left|\det\left(\frac{\partial F_x(u')}{\partial u'}\right)\right|^{-1}_{u'=H_x(F_x(u))} \mathcal{N}(u'; 0, I)\, du'$$

$$= \mathcal{N}\big(H_x(F_x(u)); 0, I\big) \left|\det\left(\frac{\partial F_x(u')}{\partial u'}\right)\right|^{-1}_{u'=H_x(F_x(u))}$$

Finally, since $H_x(F_x(u)) = u$, we simplify to

$$p(F_x(u) \mid x) = \mathcal{N}(u; 0, I) \left|\det\left(\frac{\partial F_x(u')}{\partial u'}\right)\right|^{-1}_{u'=H_x(F_x(u))}$$

Plugging this back into $\mathcal{L}$ yields

$$\mathcal{L} = \iint \tilde{p}(x)\mathcal{N}(u; 0, I) \log\left(\frac{\tilde{p}(x)\mathcal{N}(u; 0, I)}{q(x \mid F_x(u))q(F_x(u))} \left|\det\left(\frac{\partial F_x(u')}{\partial u'}\right)\right|^{-1}_{u'=H_x(F_x(u))}\right) du\, dx$$

which is the desired expression.

(aii) Let us perform the change of variables $v = F_x(u')$. Because F_x is invertible (for fixed x), the inverse transformation is $u' = H_x(v)$. Differentiate $v = F_x(u')$ with respect to u' to obtain the Jacobian matrix

$$\frac{\partial v}{\partial u'} = \frac{\partial F_x(u')}{\partial u'}$$

Since $u' = H_x(v)$ is the inverse mapping, the Jacobians satisfies

$$\frac{\partial u'}{\partial v} = \left(\frac{\partial v}{\partial u'}\right)^{-1} = \left(\frac{\partial F_x(u')}{\partial u'}\right)^{-1}, \quad \text{with } u' = H_x(v)$$

Taking determinants and absolute values gives

$$\left|\det\left(\frac{\partial u'}{\partial v}\right)\right| = \frac{1}{\left|\det\left(\frac{\partial F_x(u')}{\partial u'}\right)\right|}\Bigg|_{u'=H_x(v)}$$

Now we simplify the loss expression from part (i). First, the determinant factor in part (i) is evaluated at $u' = H_x(F_x(u))$. But $H_x(F_x(u)) = u$, so

$$\left|\det\left(\frac{\partial F_x(u')}{\partial u'}\right)\right|^{-1}_{u'=H_x(F_x(u))} = \left|\det\left(\frac{\partial F_x(u)}{\partial u}\right)\right|^{-1}$$

Second, the prior is given as $q(z) = \mathcal{N}(z; 0, I)$, hence

$$q(F_x(u)) = \mathcal{N}(F_x(u); 0, I)$$

Substituting both simplifications into the expression obtained in part (i) yields

$$\mathcal{L} = \iint \tilde{p}(x)\mathcal{N}(u; 0, I) \log\left(\frac{\tilde{p}(x)\mathcal{N}(u; 0, I)}{q(x \mid F_x(u))\mathcal{N}(F_x(u); 0, I)} \left|\det\left(\frac{\partial F_x(u)}{\partial u}\right)\right|^{-1}\right) du\, dx$$

which is exactly the required simplified form.

(aiii) (i) The Jacobian determinant appears in the denominator because an invertible transformation changes volume, and probability mass must be conserved.

More concretely, for fixed x we have an invertible map $z = F_x(u)$ that pushes forward the base variable $u \sim \mathcal{N}(0, I)$ to the latent variable z. Consider a small region A in u-space around u_0. Under the map F_x, this region is sent to a region $F_x(A)$ in z-space. The probability mass assigned to these regions must match:

$$\int_A \mathcal{N}(u; 0, I)\, \mathrm{d}u = \int_{F_x(A)} p(z \mid x)\, \mathrm{d}z$$

For small regions, volumes transform approximately by the Jacobian:

$$\mathrm{Vol}(F_x(A)) \approx \left| \det \left(\frac{\partial F_x(u_0)}{\partial u} \right) \right| \mathrm{Vol}(A)$$

If the map expands volume (large determinant), then the same probability mass is spread over a larger region in z-space, so the density must decrease, giving a factor of the inverse determinant. This is exactly the change-of-variables formula:

$$p(z \mid x) = \mathcal{N}(u; 0, I) \left| \det \left(\frac{\partial F_x(u)}{\partial u} \right) \right|^{-1} \quad \text{with } u = H_x(z)$$

Equivalently, in the delta-function derivation, the identity that converts $\delta(z - F_x(u))$ into a delta in u introduces the same inverse Jacobian factor, which is why the determinant shows up in the denominator.

(ii) The conditional flow $F_x(\cdot)$ makes $p(z \mid x)$ strictly more expressive than any diagonal Gaussian because it represents a strictly larger family of conditional densities.

A diagonal Gaussian posterior has the form

$$p_{\mathrm{diag}}(z \mid x) = \mathcal{N}\big(z; \mu(x), \mathrm{diag}(\sigma^2(x))\big)$$

which is always unimodal, symmetric in the sense of being an axis-aligned ellipsoid after centering, and it cannot represent nonlinear dependencies among coordinates beyond independent scaling.

A conditional flow posterior is the pushforward of a standard Gaussian through an x-dependent invertible transformation:

$$z = F_x(u), \qquad u \sim \mathcal{N}(0, I)$$

This can represent:

- full (non-diagonal) covariance and correlations among latent coordinates,
- nonlinear distortions (skewness, curved manifolds of high density, heavy tails depending on the chosen flow),
- complex conditional shapes that vary with x beyond a mean and diagonal variance.

Moreover, diagonal Gaussians are included as a special case of conditional flows: if we choose an affine, componentwise invertible map

$$F_x(u) = \mu(x) + D(x)u, \quad \text{where } D(x) = \operatorname{diag}(\sigma(x)) \text{ with all entries positive}$$

then z is exactly diagonal Gaussian with mean $\mu(x)$ and diagonal covariance $D(x)D(x)^\top$. Since conditional flows allow far more general invertible maps than these affine-diagonal ones, the set of posteriors representable by conditional flows strictly contains the diagonal Gaussian family, hence is strictly more expressive.

(bi) We are given

$$F_x(u) = F(\sigma u + x), \qquad q(x \mid z) = \mathcal{N}\big(x; F^{-1}(z), \sigma I\big)$$

where $\sigma > 0$ is a small constant and F is an unconditional invertible flow.

Let $w = \sigma u + x$. Then $F_x(u) = F(w)$. Using invertibility of F,

$$F^{-1}(F_x(u)) = F^{-1}(F(w)) = w = \sigma u + x$$

Therefore the conditional likelihood evaluated at $z = F_x(u)$ becomes

$$q(x \mid F_x(u)) = \mathcal{N}\big(x; \sigma u + x, \sigma I\big)$$

This is a Gaussian with mean $\sigma u + x$ and covariance $\sigma^2 I$, so its log-density is

$$\log q(x \mid F_x(u)) = -\frac{d}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|x - (\sigma u + x)\|^2$$

But $x - (\sigma u + x) = -\sigma u$, hence

$$\|x - (\sigma u + x)\|^2 = \|\sigma u\|^2 = \sigma^2\|u\|^2$$

Substituting this yields

$$\log q(x \mid F_x(u)) = -\frac{d}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sigma^2\|u\|^2 = -\frac{d}{2}\log(2\pi\sigma^2) - \frac{1}{2}\|u\|^2$$

This is the required simplification.

(bii) Let $w = \sigma u + x$. Then $F_x(u) = F(w)$. By the chain rule,

$$\frac{\partial F_x(u)}{\partial u} = \frac{\partial F(w)}{\partial w}\frac{\partial w}{\partial u}$$

Since $w = \sigma u + x$, we have $\frac{\partial w}{\partial u} = \sigma I$. Taking determinants and absolute values,

$$\left|\det\left(\frac{\partial F_x(u)}{\partial u}\right)\right| = \left|\det\left(\frac{\partial F(w)}{\partial w}\right)\right| |\det(\sigma I)|$$

Because $\det(\sigma I) = \sigma^d$, we get

$$\left|\det\left(\frac{\partial F_x(u)}{\partial u}\right)\right| = \sigma^d \left|\det\left(\frac{\partial F(w)}{\partial w}\right)\right|$$

Taking logs,

$$\log\left|\det\left(\frac{\partial F_x(u)}{\partial u}\right)\right| = \log(\sigma^d) + \log\left|\det\left(\frac{\partial F(w)}{\partial w}\right)\right| = d\log\sigma + \log\left|\det\left(\frac{\partial F(w)}{\partial w}\right)\right|$$

This is exactly the desired identity.

(biii) Start from the loss expression obtained previously:

$$\mathcal{L} = \iint \tilde{p}(x)\,\mathcal{N}(u;0,I)\,\log\left(\frac{\tilde{p}(x)\,\mathcal{N}(u;0,I)}{q(x \mid F_x(u))\,\mathcal{N}(F_x(u);0,I)}\left|\det\left(\frac{\partial F_x(u)}{\partial u}\right)\right|^{-1}\right)\mathrm{d}u\,\mathrm{d}x$$

Expand the logarithm into a sum:

$$\mathcal{L} = \iint \tilde{p}(x)\,\mathcal{N}(u;0,I)\Bigg[\log\tilde{p}(x) + \log\mathcal{N}(u;0,I) - \log q(x \mid F_x(u)) - \log\mathcal{N}(F_x(u);0,I) - \log\left|\det\left(\frac{\partial F_x(u)}{\partial u}\right)\right|\Bigg]\mathrm{d}u\,\mathrm{d}x$$

Now use the result of (i):

$$\log q(x \mid F_x(u)) = -\frac{d}{2}\log(2\pi\sigma^2) - \frac{1}{2}\|u\|^2$$

Crucially, this contains no trainable parameters of F (and it does not depend on x except through constants that cancel), because the mean uses $F^{-1}(F(\sigma u + x)) = \sigma u + x$, which removes any dependence on the internal parameters of F. Thus, $-\log q(x \mid F_x(u))$ contributes only terms that depend on u and the fixed constant σ, but not on the trainable flow.

Next, expand the standard Gaussian terms:

$$\log\mathcal{N}(u;0,I) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\|u\|^2, \qquad -\log\mathcal{N}(F_x(u);0,I) = \frac{d}{2}\log(2\pi) + \frac{1}{2}\|F_x(u)\|^2$$

Substituting these expansions into the bracketed expression shows that the only terms involving the flow are

$$\frac{1}{2}\|F_x(u)\|^2 - \log\left|\det\left(\frac{\partial F_x(u)}{\partial u}\right)\right|$$

All remaining terms depend only on $\tilde{p}(x)$, u, and constants such as σ and d. Since $\tilde{p}(x)$ is fixed data and σ is fixed, these remaining terms contribute an additive constant with respect to optimizing the parameters of F. Therefore, up to an additive constant,

$$\mathcal{L} = \mathbb{E}_{x\sim\tilde{p}(x),\,u\sim\mathcal{N}(0,I)}\left[\frac{1}{2}\|F_x(u)\|^2 - \log\left|\det\left(\frac{\partial F_x(u)}{\partial u}\right)\right|\right] + \text{const}$$

Finally, substitute $F_x(u) = F(\sigma u + x)$:

$$\mathcal{L} = \mathbb{E}_{x\sim\tilde{p}(x),\,u\sim\mathcal{N}(0,I)}\left[\frac{1}{2}\|F(\sigma u + x)\|^2 - \log\left|\det\left(\frac{\partial F(\sigma u + x)}{\partial u}\right)\right|\right] + \text{const}$$

which is the claimed collapse.

This shows that the conditional-flow VAE objective reduces to standard flow maximum-likelihood training on the perturbed input

$$y = x + \sigma u, \qquad u \sim \mathcal{N}(0,I)$$

which is exactly input dequantization (adding small noise) with noise level σ.

Problem 5: VP-NFs are not universal

(a) By the change of variables formula,

$$p_\theta(x) = p_Z(f_\theta^{-1}(x)) \left| \det \left(\frac{\partial f_\theta^{-1}(x)}{\partial x} \right) \right| = \frac{1}{J} p_Z(f_\theta^{-1}(x))$$

and in particular the density is globally bounded by the maximum of the base density:

$$p_\theta(x) \leq \frac{1}{J} \sup_z p_Z(z) = \frac{1}{J} p_Z(0) = \frac{1}{2\pi J}$$

Now choose $\epsilon > 0$ such that $0.9 - \epsilon > \frac{1}{2\pi J}$. Then for every x,

$$p_\theta(x) \leq \frac{1}{2\pi J} < 0.9 - \epsilon$$

so the set $A = \{x \in \mathbb{R}^2 : p_\theta(x) \geq 0.9 - \epsilon\}$ is empty. Since $A = \varnothing$, we have $\bar{A} = B \setminus A = B$, so the hinted event $E = \bar{A}$ is simply $E = B$, where

$$B = [-0.5, 0.5] \times [-0.5, 0.5]$$

On B, the target density satisfies $p(x, y) = 0.9$ by construction, and $\mathrm{Vol}(B) = 1$, hence

$$p(B) = \int_B p(x, y) \,\mathrm{d}x \,\mathrm{d}y = 0.9$$

On the other hand, since $A = \varnothing$ we have $p_\theta(x, y) < 0.9 - \epsilon$ everywhere, and in particular on B, so

$$p_\theta(B) = \int_B p_\theta(x, y) \,\mathrm{d}x \,\mathrm{d}y \leq (0.9 - \epsilon)\mathrm{Vol}(B) = 0.9 - \epsilon$$

Therefore,

$$|p(B) - p_\theta(B)| \geq 0.9 - (0.9 - \epsilon) = \epsilon > 0$$

which proves that there exists a measurable event E (namely $E = B = \bar{A}$) such that $|p_\theta(E) - p(E)| > 0$ when $A = \varnothing$.

Finally, apply Pinsker's inequality in the form

$$\sup_E |p(E) - p_\theta(E)| \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(p \| p_\theta)}$$

Since the supremum is at least the value at $E = B$, we obtain

$$\epsilon \leq |p(B) - p_\theta(B)| \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(p \| p_\theta)}$$

and hence

$$D_{\mathrm{KL}}(p \| p_\theta) \geq 2\epsilon^2 > 0$$

Thus for the given target p, every constant-Jacobian flow model has a strictly positive KL gap for some $\epsilon > 0$, so this family cannot be a universal distribution approximator under KL divergence.

(b) Define the measurable set in data space

$$A = \{x \in \mathbb{R}^2 : p_\theta(x) \geq 0.9 - \epsilon\}$$

and define its preimage in latent space

$$C = f_\theta^{-1}(A) = \{z \in \mathbb{R}^2 : f_\theta(z) \in A\}$$

Because f_θ is invertible and has constant Jacobian determinant J, the change of variables formula implies that Lebesgue volume scales by the constant factor J:

$$\mathrm{Vol}(A) = \int_A 1 \,\mathrm{d}x\,\mathrm{d}y = \int_C \left|\det\left(\frac{\partial f_\theta(z)}{\partial z}\right)\right| \mathrm{d}z = J \int_C 1 \,\mathrm{d}z = J\,\mathrm{Vol}(C)$$

Equivalently, $\mathrm{Vol}(C) = \mathrm{Vol}(A)/J$. In particular, for volume-preserving flows ($J = 1$), $\mathrm{Vol}(C) = \mathrm{Vol}(A)$.

Next, rewrite the defining condition for A in latent variables. Since $p_\theta(f_\theta(z)) = \frac{1}{J} p_Z(z)$, we have

$$f_\theta(z) \in A \iff p_\theta(f_\theta(z)) \geq 0.9 - \epsilon \iff p_Z(z) \geq J(0.9 - \epsilon)$$

For $z \in \mathbb{R}^2$ with $z \sim \mathcal{N}(0, I)$,

$$p_Z(z) = \frac{1}{2\pi} \exp\left(-\frac{\|z\|^2}{2}\right)$$

Thus

$$p_Z(z) \geq J(0.9 - \epsilon) \iff \exp\left(-\frac{\|z\|^2}{2}\right) \geq 2\pi J(0.9 - \epsilon)$$

If $2\pi J(0.9 - \epsilon) > 1$, then the inequality is impossible and $C = \varnothing$, so $\mathrm{Vol}(C) = 0$.

Otherwise, when $0 < 2\pi J(0.9 - \epsilon) \leq 1$, take logs to obtain

$$-\frac{\|z\|^2}{2} \geq \log\bigl(2\pi J(0.9 - \epsilon)\bigr) \iff \|z\|^2 \leq -2\log\bigl(2\pi J(0.9 - \epsilon)\bigr)$$

Therefore C is a disk in $\mathbb{R}^2$ centered at the origin with radius

$$r = \sqrt{-2\log\bigl(2\pi J(0.9 - \epsilon)\bigr)}$$

Its volume (area) is

$$\mathrm{Vol}(C) = \pi r^2 = -2\pi \log\bigl(2\pi J(0.9 - \epsilon)\bigr) = 2\pi \log\left(\frac{1}{2\pi J(0.9 - \epsilon)}\right)$$

A simple upper bound follows from $\log t \leq t - 1$ applied to $t = \frac{1}{2\pi J(0.9-\epsilon)} \geq 1$:

$$\mathrm{Vol}(C) = 2\pi \log t \leq 2\pi(t - 1) \leq 2\pi t = \frac{1}{J(0.9 - \epsilon)}$$

Combining with $\mathrm{Vol}(A) = J\,\mathrm{Vol}(C)$ gives the corresponding bound in data space:

$$\mathrm{Vol}(A) = J\,\mathrm{Vol}(C) \leq \frac{1}{0.9 - \epsilon}$$

(c) Fix a VP-NF (more generally, any constant-Jacobian flow) with

$$\left|\det\left(\frac{\partial f_\theta(z)}{\partial z}\right)\right| = J$$

and base density $p_Z(z) = (2\pi)^{-1}\exp\left(-\frac{\|z\|^2}{2}\right)$. Then

$$p_\theta(x) = \frac{1}{J}p_Z\big(f_\theta^{-1}(x)\big), \qquad \sup_x p_\theta(x) = \frac{1}{2\pi J}$$

Choose $\epsilon > 0$ so that $0 < 0.9 - \epsilon < \frac{1}{2\pi J}$ Then $A = \{x : p_\theta(x) \geq 0.9 - \epsilon\}$ is nonempty, because the threshold is strictly below the maximum value of p_θ.

Now let

$$B = [-0.5, 0.5] \times [-0.5, 0.5], \qquad \bar{A} = B \setminus A$$

We will show that $\bar{A}$ has positive area, hence gives a strict probability mismatch. By the change-of-variables calculation from the previous part, the preimage $C = f_\theta^{-1}(A)$ is exactly the set where the base density exceeds the corresponding threshold:

$$z \in C \iff p_\theta(f_\theta(z)) \geq 0.9 - \epsilon \iff \frac{1}{J}p_Z(z) \geq 0.9 - \epsilon \iff p_Z(z) \geq J(0.9 - \epsilon)$$

Since p_Z is radially decreasing, C is a disk centered at the origin. In particular,

$$\mathrm{Vol}(A) = J\,\mathrm{Vol}(C)$$

Moreover, as $0.9 - \epsilon$ increases to $(2\pi J)^{-1}$ from below, the disk C shrinks to the point $\{0\}$, so $\mathrm{Vol}(C)$ (and hence $\mathrm{Vol}(A)$) can be made arbitrarily small by choosing ϵ so that $0.9 - \epsilon$ is sufficiently close to $(2\pi J)^{-1}$. Therefore we may choose ϵ (still satisfying $0 < 0.9 - \epsilon < (2\pi J)^{-1}$) such that

$$\mathrm{Vol}(A) < 1 = \mathrm{Vol}(B)$$

Then

$$\mathrm{Vol}(\bar{A}) = \mathrm{Vol}(B \setminus A) \geq \mathrm{Vol}(B) - \mathrm{Vol}(A) > 0$$

Take the measurable event $E = \bar{A}$. On B the target density is $p(x, y) = 0.9$ by construction, hence

$$p(E) = \int_E 0.9\,\mathrm{d}x\,\mathrm{d}y = 0.9\,\mathrm{Vol}(E)$$

On the other hand, by definition of $E = B \setminus A$, for every $(x, y) \in E$ we have $p_\theta(x, y) < 0.9 - \epsilon$, so

$$p_\theta(E) = \int_E p_\theta(x, y)\,\mathrm{d}x\,\mathrm{d}y \leq \int_E (0.9 - \epsilon)\,\mathrm{d}x\,\mathrm{d}y = (0.9 - \epsilon)\,\mathrm{Vol}(E)$$

Therefore

$$|p(E) - p_\theta(E)| \geq 0.9\,\mathrm{Vol}(E) - (0.9 - \epsilon)\,\mathrm{Vol}(E) = \epsilon\,\mathrm{Vol}(E) > 0$$

since $\mathrm{Vol}(E) = \mathrm{Vol}(\bar{A}) > 0$.

Now apply Pinsker's inequality:

$$\sup_E |p(E) - p_\theta(E)| \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(p \| p_\theta)}$$

The supremum is at least the value at our specific event $E = \bar{A}$, hence

$$\epsilon \, \mathrm{Vol}(E) \leq |p(E) - p_\theta(E)| \leq \sqrt{\frac{1}{2} D_{\mathrm{KL}}(p \| p_\theta)}$$

Squaring both sides gives the strictly positive lower bound

$$D_{\mathrm{KL}}(p \| p_\theta) \geq 2\epsilon^2 \mathrm{Vol}(E)^2 > 0$$

(d) A family of models is universal under KL if, for every target distribution p, one can find models p_{θ_n} with $D_{\mathrm{KL}}(p \| p_{\theta_n}) \to 0$.

Parts (a) and (c) exhibit a specific target distribution p for which every constant-Jacobian flow model p_θ has a strictly positive KL gap:

$$D_{\mathrm{KL}}(p \| p_\theta) \geq c$$

for some $c > 0$ (obtained from Pinsker's inequality using an event E with nonzero probability mismatch). Since the KL divergence cannot be made arbitrarily small for this p, the family of VP-NFs (and more generally constant-Jacobian flows) cannot approximate every distribution, so it is not a universal distribution approximator.