```python
    def log_prob(self, x):
        z, log_det_J = self.f(x)
        log_p = (log_standard_normal(z) + log_det_J.unsqueeze(1))
        return log_p
```

Listing 4.11 A flow-based prior class

4.4.1.7 Remarks

In practice, we can use any density estimator to model $p_\lambda(\mathbf{z})$. For instance, we can use an autoregressive model [26] or more advanced approaches like resampled priors [27] or hierarchical priors [51]. Therefore, there are many options! However, there is still an open question **how** to do that and **what** role the prior (the marginal) should play. As I mentioned in the beginning, Bayesianists would say that the marginal should impose some constraints on the latent space or, in other words, our prior knowledge about it. I am a Bayesiast deep down in my heart and this way of thinking is very appealing to me. However, it is still unclear what is a good latent representation. This question is as old as mathematical modeling. I think that it would be interesting to look at optimization techniques, maybe applying a gradient-based method to all parameters/weights at once is not the best solution. Anyhow, I am pretty sure that modeling the prior is more important than many people think and plays a crucial role in VAEs.

4.4.2 Variational Posteriors

In general, variational inference searches for the best posterior approximation within a parametric family of distributions. Hence, recovering the true posterior is possible only if it happens to be in the chosen family. In particular, with widely used variational families such as diagonal covariance Gaussian distributions, the variational approximation is likely to be insufficient. Therefore, designing tractable and more expressive variational families is an important problem in VAEs. Here, we present two families of conditional normalizing flows that can be used for that purpose, namely, Householder flows [20] and Sylvester flows [16]. There are other interesting families and we refer the reader to the original papers, e.g., the generalized Sylvester flows [17] and the Inverse Autoregressive Flows [18].

The general idea about using the normalizing flows to parameterize the variational posteriors is to start with a relatively simple distribution like the Gaussian with the diagonal covariance matrix and then transform it to a complex distribution through a series of invertible transformations [19]. Formally speaking, we start with the latents $\mathbf{z}^{(0)}$ distributed according to $\mathcal{N}(\mathbf{z}^{(0)}|\mu(\mathbf{x}, \sigma^2(\mathbf{x})$ and then after applying a series of invertible transformations $\mathbf{f}^{(t)}$, for $t = 1, \ldots, T$, the last iterate gives a random variable $\mathbf{z}^{(T)}$ that has a more flexible distribution. Once we choose transformations $\mathbf{f}^{(t)}$ for which the Jacobian-determinant can be computed, we aim at optimizing the following objective:

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}^{(0)}|\mathbf{x})}\Big[\ln p(\mathbf{x}|\mathbf{z}^{(T)}) + \sum_{t=1}^{T} \ln \Big|\det \frac{\partial \mathbf{f}^{(t)}}{\partial \mathbf{z}^{(t-1)}}\Big|\Big] - \mathrm{KL}\big(q(\mathbf{z}^{(0)}|\mathbf{x})||p(\mathbf{z}^{(T)})\big). \tag{4.54}$$

In fact, the normalizing flow can be used to enrich the posterior of the VAE with small or even none modifications in the architecture of the encoder and the decoder.

4.4.2.1 Variational Posteriors with Householder Flows [20]

Motivation

First, we notice that any full-covariance matrix $\mathbf{\Sigma}$ can be represented by the eigenvalue decomposition using eigenvectors and eigenvalues:

$$\mathbf{\Sigma} = \mathbf{U}\mathbf{D}\mathbf{U}^{\top}, \tag{4.55}$$

where $\mathbf{U}$ is an orthogonal matrix with eigenvectors in columns and $\mathbf{D}$ is a diagonal matrix with eigenvalues. In the case of the vanilla VAE, it would be tempting to model the matrix $\mathbf{U}$ to obtain a full-covariance matrix. The procedure would require a linear transformation of a random variable using an orthogonal matrix $\mathbf{U}$. Since the absolute value of the Jacobian-determinant of an orthogonal matrix is 1, for $\mathbf{z}^{(1)} = \mathbf{U}\mathbf{z}^{(0)}$ one gets $\mathbf{z}^{(1)} \sim \mathcal{N}(\mathbf{U}\boldsymbol{\mu}, \mathbf{U}\ \mathrm{diag}(\boldsymbol{\sigma}^2)\ \mathbf{U}^{\top})$. If $\mathrm{diag}(\boldsymbol{\sigma}^2)$ coincides with true $\mathbf{D}$, then it would be possible to resemble the true full-covariance matrix. Hence, the main goal would be to model the orthogonal matrix of eigenvectors.

Generally, the task of modeling an orthogonal matrix in a principled manner is rather non-trivial. However, first we notice that any orthogonal matrix can be represented in the following form [53, 54]:

Theorem 4.1 (The Basis-Kernel Representation of Orthogonal Matrices) *For any $M \times M$ orthogonal matrix $\mathbf{U}$, there exist a full-rank $M \times K$ matrix $\mathbf{Y}$ (the basis) and a nonsingular (triangular) $K \times K$ matrix $\mathbf{S}$ (the kernel), $K \leq M$, such that:*

$$\mathbf{U} = \mathbf{I} - \mathbf{Y}\mathbf{S}\mathbf{Y}^{\top}. \tag{4.56}$$

The value K is called the *degree* of the orthogonal matrix. Further, it can be shown that any orthogonal matrix of degree K can be expressed using the product of Householder transformations [53, 54], namely:

Theorem 4.2 *Any orthogonal matrix with the basis acting on the K-dimensional subspace can be expressed as a product of exactly K Householder matrices:*

$$\mathbf{U} = \mathbf{H}_K \mathbf{H}_{K-1} \cdots \mathbf{H}_1, \tag{4.57}$$

where $\mathbf{H}_k = \mathbf{I} - \mathbf{S}_{kk}\mathbf{Y}_{\cdot k}(\mathbf{Y}_{\cdot k})^{\top}$, for $k = 1, \ldots, K$.

Theoretically, Theorem 4.2 shows that we can model any orthogonal matrix in a principled fashion using K Householder transformations. Moreover, the Householder matrix $\mathbf{H}_k$ is *orthogonal* matrix itself [55]. Therefore, this property and the Theorem 4.2 put the Householder transformation as a perfect candidate for formulating a volume-preserving flow that allows to approximate (or even capture) the true full-covariance matrix.

Householder Flows

The *Householder transformation* is defined as follows. For a given vector $\mathbf{z}^{(t-1)}$, the reflection hyperplane can be defined by a vector (a *Householder vector*) $\mathbf{v}_t \in \mathbb{R}^M$ that is orthogonal to the hyperplane, and the reflection of this point about the hyperplane is [55]

$$\mathbf{z}^{(t)} = \left(\mathbf{I} - 2\frac{\mathbf{v}_t\mathbf{v}_t^{\top}}{||\mathbf{v}_t||^2}\right)\mathbf{z}^{(t-1)} \tag{4.58}$$

$$= \mathbf{H}_t\mathbf{z}^{(t-1)}, \tag{4.59}$$

where $\mathbf{H}_t = \mathbf{I} - 2\frac{\mathbf{v}_t\mathbf{v}_t^{\top}}{||\mathbf{v}_t||^2}$ is called the *Householder matrix*.

The most important property of $\mathbf{H}_t$ is that it is an orthogonal matrix and hence the absolute value of the Jacobian-determinant is equal to 1. This fact significantly simplifies the objective (4.54) because $\ln\left|\det\frac{\partial\mathbf{H}_t\mathbf{z}^{(t-1)}}{\partial\mathbf{z}^{(t-1)}}\right| = 0$, for $t = 1, \ldots, T$. Starting from a simple posterior with the diagonal covariance matrix for $\mathbf{z}^{(0)}$, the series of T linear transformations given by (4.58) defines a new type of volume-preserving flow that we refer to as the *Householder flow* (HF). The vectors $\mathbf{v}_t$, $t = 1, \ldots, T$, are produced by the encoder network along with means and variances using a linear layer with the input $\mathbf{v}_{t-1}$, where $\mathbf{v}_0 = \mathbf{h}$ is the last hidden layer of the encoder network. The idea of the Householder flow is schematically presented in Fig. 4.14. Once the encoder returns the first Householder vector, the Householder

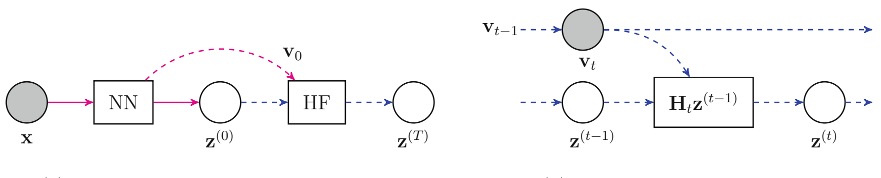

Fig. 4.14 A schematic representation of the encoder network with the Householder flow. (**a**) The general architecture of the VAE+HF: The encoder returns means and variances for the posterior and the first Householder vector that is further used to formulate the Householder flow. (**b**) A single step of the Householder flow that uses linear Householder transformation. In both panels solid lines correspond to the encoder network and the dashed lines are additional quantities required by the HF