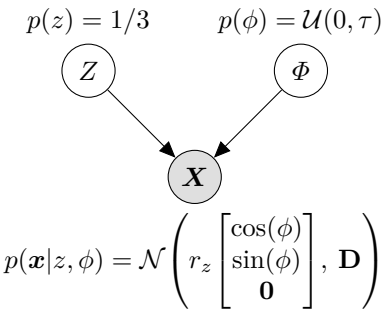
A Appendix

A.1 The concentric-rings datasets

The generative process for the concentric rings is depicted in the graphical model at right (we use $\tau := 2\pi$). The covariance matrix $\mathbf{D}$ is diagonal with $\sigma^2_{\text{big}} := 0.1^2$ for the first three diagonal entries and $\sigma^2_{\text{small}} := 0.01^2$ for the remaining seven. The three radius lengths are $r \in \{1.0, 2.0, 3.0\}$. We can visualize the distribution of (X_1, X_2), that is, the "thick" parts of this distribution, by explicitly marginalizing out Z and Φ (and ignoring the other dimensions of $\boldsymbol{X}$). Anticipating that the distribution should be invariant to the angle of (X_1, X_2), we define the length $\ell := \sqrt{x_1^2 + x_2^2}$. Then we obtain



$$
\begin{aligned}
p(x_1, x_2) &= \sum_z \int_\phi p(x_1, x_2|\phi, z) p(\phi) p(z) \, \mathrm{d}\phi \\
&\propto \sum_z \int_\phi \mathcal{N}\left(r_z \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \end{bmatrix}, \, \sigma^2_{\text{big}} \mathbf{I} \right) \mathrm{d}\phi \\
&\propto \sum_z \int_\phi \exp\left\{ -\frac{\ell^2 - 2 r_z (x_1 \cos\phi + x_2 \sin\phi) + r_z^2}{2\sigma^2_{\text{big}}} \right\} \mathrm{d}\phi \\
&\propto \sum_z I_0\left(\frac{r_z \ell}{\sigma^2_{\text{big}}} \right) \exp\left\{ -\frac{\ell^2 - r_z^2}{2\sigma^2_{\text{big}}} \right\} \\
&\propto \sum_z \tilde{I}_0\left(\frac{r_z \ell}{\sigma^2_{\text{big}}} \right) \exp\left\{ -\frac{\ell^2 - 2 r_z \ell + r_z^2}{2\sigma^2_{\text{big}}} \right\} \\
&\propto \sum_z \tilde{I}_0\left(\frac{r_z \ell}{\sigma^2_{\text{big}}} \right) \mathcal{N}\left(\ell; r_z, \, \sigma^2_{\text{big}} \right),
\end{aligned}
$$

with $\tilde{I}_0\left(\frac{r\ell}{\sigma^2_{\text{big}}}\right)$ a scaled version of the modified Bessel function of the first kind (to wit, `i0e` in `scipy`). Since $\sigma_{\text{big}} = 0.1$ is small compared to the inter-ring differences (1.0), $r_z \ell \approx r_z^2$. Thus, the distribution is approximately a one-dimensional Gaussian mixture model along any radius, with the Bessel functions providing the mixing weights.

A.2 Energy evolution during Langevin dynamics

Fig. 9 shows the evolution of the energy across LD in models fully trained on CIFAR-10.

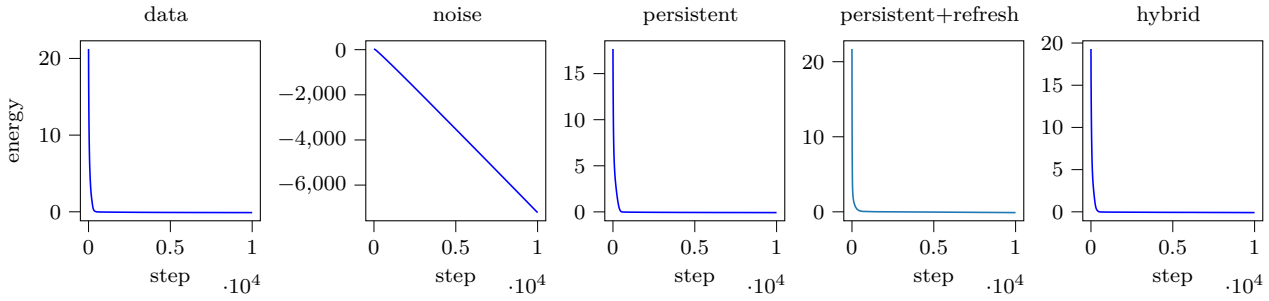


Figure 9: Energy vs. Langevin steps in models trained with various schemes, identified in the column labels, on CIFAR-10.

A.3 The computational loss

Nijkamp et al. (2020) identify the difference in average energies,

$$d := \Big\langle E(\boldsymbol{X}, \boldsymbol{\theta}) \Big\rangle_{\boldsymbol{X}} - \Big\langle E(\hat{\boldsymbol{X}}, \boldsymbol{\theta}) \Big\rangle_{\hat{\boldsymbol{X}}}. \tag{7}$$

as a critical quantity during EBM training. (We leave open for now under precisely what distribution the second expectation is taken, but assume that it depends on the parameters $\boldsymbol{\theta}$.) They make, in particular, the empirical observation that d and the fluctuations in the average magnitude of the force $\partial E/\partial \boldsymbol{x}$ correlate highly and positively at the same training batch, but negatively at lags of two or three batches. Thus stable training requires d to balance roughly between positive and negative values across gradient updates.

The computational loss is a gradient. It is tempting to interpret Eq. 7 as the loss function itself because its gradient superficially resembles Eq. 2. But the averaging brackets in the second term of Eq. 7 depend on the parameters, so the gradient cannot pass inside. Nijkamp and colleagues point out this similarity to the loss but rightly resist the temptation to assimilate the two. But what then is the relationship?

Let us re-express the energy function (without loss of generality) as the product of a fixed-scale energy $\tilde{E} \in [-1, 1]$, and a "coldness" parameter $\beta > 0$: $E = \beta \tilde{E}$. This parameter could be explicit, or it could be merely a fictitious stand-in for the net effect of all the parameters on the energy magnitude. In either case, the loss gradient, Eq. 2, with respect to this parameter is

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\beta} \mathrm{D}_{\mathrm{KL}}\{p(\boldsymbol{X}) \| \hat{p}(\boldsymbol{X}; \boldsymbol{\theta})\} &= \mathbb{E}_{\boldsymbol{X}}\left[\frac{\partial E}{\partial \beta}(\boldsymbol{X}, \boldsymbol{\theta})\right] - \mathbb{E}_{\hat{\boldsymbol{X}}}\left[\frac{\partial E}{\partial \beta}(\hat{\boldsymbol{X}}, \boldsymbol{\theta})\right] \\
&= \frac{1}{\beta}\Big(\mathbb{E}_{\boldsymbol{X}}[E(\boldsymbol{X}, \boldsymbol{\theta})] - \mathbb{E}_{\hat{\boldsymbol{X}}}\Big[E(\hat{\boldsymbol{X}}, \boldsymbol{\theta})\Big]\Big) = \frac{d}{\beta}.
\end{aligned}$$

Thus, the difference in expected energies is proportional to the gradient of the loss with respect to the energy magnitude.

This explains the relationships observed in (Nijkamp et al., 2020). If Eq. 7 were a proxy for the loss, gradient descent (with sufficiently small gradient steps) would always drive it downwards. Instead, d is a loss *gradient*, so during successful training runs it oscillates around zero. Negative values of d imply that the coldness β needs to be adjusted upward by gradient descent. Equivalently, negative d implies that the energy $E = \beta \tilde{E}$, and its gradient with respect to the data $\partial E/\partial \boldsymbol{x}$, the force, are too small. Positive values of d likewise imply that the force is too large. This explains the correlation between the force magnitude and d. Likewise, gradient descent will increase negative d and decrease positive d, or equivalently increase forces that are too small and decrease forces that are too large. This explains the negative cross-correlation at short lags.

The difference in expected energies therefore corresponds to the change made to the overall magnitude of the energy at that point in training. Large fluctuations in the magnitude updates indicate instability and are undesirable.

A.4 The temperature parameter

In the limit as the step size ϵ approaches zero, the discretized Langevin dynamics

$$\hat{\boldsymbol{X}}_{l+1} = \hat{\boldsymbol{X}}_l - \epsilon \frac{\partial E}{\partial \boldsymbol{x}}(\hat{\boldsymbol{X}}_l, \boldsymbol{\theta}) + \sqrt{2\epsilon}\hat{\boldsymbol{Z}}_l \tag{8}$$

has as its stationary distribution Eq. 1, which we repeat here for convenience:

$$\hat{p}(\boldsymbol{x}; \boldsymbol{\theta}) \propto \exp\{-E(\boldsymbol{x}, \boldsymbol{\theta})\}. \tag{1}$$

In practice when running Langevin dynamics, it is useful to be able to scale the gradient and noise terms independently: the latter must be small for stability, but the former must not be too small lest numerical precision of the gradient be lost. This is particular true early in training, when the energy landscape produced by the neural network is very flat.

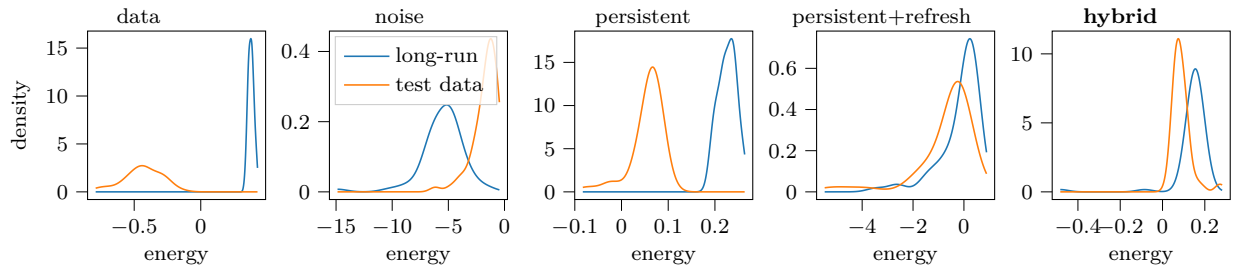


Figure 10: Distributions of energies assigned by models after training on the Oxford Flowers. Blue: long-run samples; orange: test (held-out) the Oxford Flowers samples. Plot titles specify the MCMC initialization scheme used during training.

Including the parameter T in Eq. 3 allows for independent scaling of the gradient and noise. Of course, this does change the stationary distribution; in particular, the Langevin update can be re-written as

$$\begin{aligned} \hat{\boldsymbol{X}}_{l+1} &= \hat{\boldsymbol{X}}_l - \epsilon \frac{\partial E}{\partial \boldsymbol{x}}(\hat{\boldsymbol{X}}_l, \boldsymbol{\theta}) + \sqrt{2\epsilon T}\,\hat{\boldsymbol{Z}}_l \\ \eta \coloneqq \epsilon T \implies \hat{\boldsymbol{X}}_{l+1} &= \hat{\boldsymbol{X}}_l - \eta \frac{1}{T}\frac{\partial E}{\partial \boldsymbol{x}}(\hat{\boldsymbol{X}}_l, \boldsymbol{\theta}) + \sqrt{2\eta}\,\hat{\boldsymbol{Z}}_l, \end{aligned} \tag{9}$$

and so the stationary distribution becomes

$$\hat{p}(\boldsymbol{x}; \boldsymbol{\theta}) \propto \exp\{-E(\boldsymbol{x}, \boldsymbol{\theta})/T\},$$

as can be seen by comparing Eqs. 8 and 1 with Eq. 9 (and grouping $1/T$ with the energy).

In the literature, T is frequently omitted from the stationary distribution in order to reduce clutter, as we have in Eq. 1, and in the loss and its gradients, Eqs. 2, 5, and 6. Indeed, because *normalized* probabilities are rarely computable (or computed), T is in some sense irrelevant to the reported results—it merely sets the "units" of the energy. However, in Fig. 2, we display the model densities themselves, and therefore have taken T into account.

A.5 The relative-entropy gradient for EBMs

Although it is well known, for completeness we include the derivation of the gradient of the relative entropy for energy-based models, here keeping the temperature parameter explicit:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}} \mathrm{D}_{\mathrm{KL}}\{p(\boldsymbol{X}) \| \hat{p}(\boldsymbol{X}; \boldsymbol{\theta})\} &= \mathbb{E}_{\boldsymbol{X}}\left[\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}} \log \frac{p(\boldsymbol{X})}{\hat{p}(\boldsymbol{X}; \boldsymbol{\theta})}\right] \\ &= \mathbb{E}_{\boldsymbol{X}}\left[\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}} \log \frac{\int_{\hat{\boldsymbol{x}}} \exp\{-E(\hat{\boldsymbol{x}}, \boldsymbol{\theta})/T\}\,\mathrm{d}\hat{\boldsymbol{x}}}{\exp\{-E(\boldsymbol{X}, \boldsymbol{\theta})/T\}}\right] \\ &= \frac{\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\theta}} \int_{\hat{\boldsymbol{x}}} \exp\{-E(\hat{\boldsymbol{x}}, \boldsymbol{\theta})/T\}\,\mathrm{d}\hat{\boldsymbol{x}}}{\int_{\hat{\boldsymbol{x}}} \exp\{-E(\hat{\boldsymbol{x}}, \boldsymbol{\theta})/T\}\,\mathrm{d}\hat{\boldsymbol{x}}} + \mathbb{E}_{\boldsymbol{X}}\left[\frac{1}{T}\frac{\partial E}{\partial \boldsymbol{\theta}}(\boldsymbol{X}, \boldsymbol{\theta})\right] \\ &= \frac{1}{T}\left(\mathbb{E}_{\boldsymbol{X}}\left[\frac{\partial E}{\partial \boldsymbol{\theta}}(\boldsymbol{X}, \boldsymbol{\theta})\right] - \mathbb{E}_{\hat{\boldsymbol{X}}}\left[\frac{\partial E}{\partial \boldsymbol{\theta}}(\hat{\boldsymbol{X}}, \boldsymbol{\theta})\right]\right). \end{aligned} \tag{10}$$

Ignoring the temperature parameter (or setting it to unity) produces Eq. 2 of the main text.