

Emergence of nonequilibrium latent cycles in unsupervised generative modelingMarco Baiesi^{1,2} and Alberto Rosso³¹*Department of Physics and Astronomy, University of Padova, Via Marzolo 8, I-35131 Padova, Italy*²*INFN, Sezione di Padova, Via Marzolo 8, I-35131 Padova, Italy*³*LPTMS, CNRS, Université Paris-Saclay, 91405 Orsay, France*

(Received 12 December 2025; accepted 27 April 2026; published 7 July 2026)

We show that nonequilibrium dynamics can play a constructive role in unsupervised machine learning by inducing the spontaneous emergence of latent-state cycles. We introduce a model in which visible and hidden variables interact through two independently parametrized transition matrices, defining a Markov chain whose steady state is intrinsically out of equilibrium. Likelihood maximization drives this system toward nonequilibrium steady states with finite entropy production, reduced self-transition probabilities, and persistent probability currents in the latent space. These cycles are not imposed by the architecture but arise from training, and models that develop them reproduce the empirical distribution of data classes more faithfully, with a clear correlation between agreement with the data and entropy production. Compared with equilibrium approaches such as restricted Boltzmann machines, our model breaks the detailed balance between the forward and backward conditional transitions and relies on a log-likelihood gradient that depends explicitly on the last two steps of the Markov chain. Hence, this exploration of the interface between nonequilibrium statistical physics and modern machine learning suggests that introducing irreversibility into latent-variable models can improve the fidelity of the generated data distribution.

DOI: [10.1103/8gqs-fkh6](https://doi.org/10.1103/8gqs-fkh6)**I. INTRODUCTION**

Unsupervised generative modeling aims to learn a probability distribution $p(\mathbf{x})$ from a dataset $\mathcal{D}$ of observed configurations [1]. A widespread strategy is to introduce latent variables $\mathbf{z} \in \mathcal{Z}$ and define a joint distribution $p(\mathbf{x}, \mathbf{z})$ whose marginal over $\mathbf{x}$ approximates the data distribution. Classic examples include energy-based models, such as restricted Boltzmann machines (RBMs) [2,3], where one specifies an energy function $E(\mathbf{x}, \mathbf{z})$ and sets $p(\mathbf{x}, \mathbf{z}) \sim e^{-E(\mathbf{x}, \mathbf{z})}$. In such architectures, the alternation between visible and hidden layers is implemented through Gibbs sampling steps that satisfy a detailed balance. Hence, the learning dynamics is tied to an equilibrium steady state, and the latent-variable transitions are necessarily reversible. While RBMs have been successful in modeling structured data (see, e.g., [4–7]), their reliance on reversible Gibbs updates tends to limit exploration of the configuration space: once a hidden configuration $\mathbf{z}$ is activated by a given $\mathbf{x}$, the sampler typically returns to the same neighborhood [8–10], a behavior reminiscent of slow mixing in equilibrium Monte Carlo methods [11].

Several other classes of latent-variable models have been developed to address limitations of equilibrium generative architectures. Helmholtz machines [12] and deep Boltzmann machines [13] introduced paired “recognition” and “generative” networks, trained, for example, by the wake-sleep phases [12–14], but their sampling dynamics follows a one-way

generative path rather than an alternating interaction between visible and latent variables. Modern approaches such as variational autoencoders [15,16] and diffusion-based generative models [17–22] adopt the same structure: they specify a directed generative process and use a backward inference mechanism to propagate information in the opposite direction. In these frameworks, recognition and generation proceed along separate pathways, and the latent variables do not participate in a stochastic alternation with the visible layer.

Equilibrium dynamics and detailed balance are not universal organizing principles in natural information-processing systems. In many biological contexts, functionality relies on operating out of equilibrium: energy dissipation (but also kinetic, nondissipative features [23]) enables enhanced precision through kinetic proofreading [24,25], sustained directional transport by molecular motors [26–28], adaptive chemotactic responses [29], and sensitivities that surpass equilibrium bounds [30]. This perspective motivates us to ask whether controlled violations of detailed balance can play a similarly constructive role in latent-variable generative models, beyond the equilibrium paradigm underlying standard architectures.

We propose a generative model whose visible/latent alternation is governed by a generic nonequilibrium Markov chain in a steady state. We introduce two independently parametrized transition matrices,

$$A_{\mathbf{z}\mathbf{x}} \equiv p(\mathbf{x}|\mathbf{z}), \quad (1)$$

$$\bar{A}_{\mathbf{x}\mathbf{z}} \equiv p(\mathbf{z}|\mathbf{x}), \quad (2)$$

which need not satisfy detailed balance. The composition of these transitions defines a discrete-time Markov process in the

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

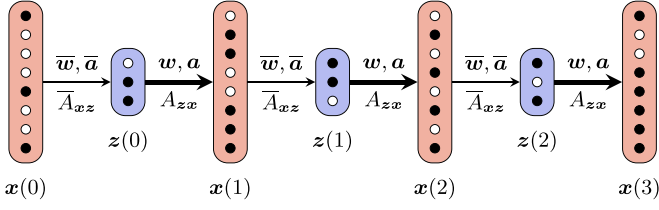


FIG. 1. Markov chain dynamics alternating visible states $x(t)$ and hidden states $z(t)$. This sequence has $T = 3$ double steps $\{x \rightarrow z \rightarrow\}$. The Markov jump probabilities $\bar{A}_{xz}$ of the $x \rightarrow z$ transitions are determined by parameters $\{\bar{w}, \bar{a}\}$ differing from the parameters $\{w, a\}$ that govern jump probabilities A_{zx} of the $z \rightarrow x$ transitions.

bipartite space $\mathcal{X} \cup \mathcal{Z}$ (see Fig. 1). Because no energy function is available, the stationary distribution has no closed-form expression. Nevertheless, we introduce an approximate log-likelihood gradient that provides a stable and fast-converging training rule.

A central observation of this study is that training a generative model based on generic Markov transitions never selects the reversible, equilibrium limit: the learned dynamics always violates detailed balance. Moreover, the models that reproduce the data classes most faithfully, with the lowest Kullback–Leibler (KL) divergence between generated and actual data, are those that operate far from equilibrium. To quantify this distance, we study the trajectories of the hidden states through the properties of the Markov matrix

$$M_{z'z} = \sum_x A_{zx} \bar{A}_{xz}, \quad (3)$$

and characterize its behavior in terms of entropy production, persistent probability currents, reduced diagonal occupancy, and long decorrelation times.

Interestingly, likelihood maximization is sufficient to drive the hidden-layer dynamics to self-organize into regular and irreversible cycles. This cyclic behavior is not imposed by the architecture but emerges spontaneously during training as M develops strong directional flows and long-range temporal structure. Models exhibiting clearer latent-state cycles reproduce more faithfully the empirical distribution of data classes.

These results point to a mechanism through which nonequilibrium dynamics can enhance generative modeling: cycling through hidden states reduces autocorrelation among fantasy samples and encourages a more thorough exploration of the data modes. This suggests that irreversibility may be beneficial not only for sampling, as in nonreversible Monte Carlo methods [31–34], but also for learning itself.

The remainder of the paper introduces the parametrized transition matrices defining the model (Sec. II), derives the associated approximate log-likelihood gradient (Sec. III), and analyzes an example of nonequilibrium steady states that arise during training through entropy production, decorrelation times, and the structure of the latent-to-latent transition matrix (Sec. IV). We end with some conclusions (Sec. V).

II. MODEL

In this paper, we focus on digital data $\mathbf{x} = (x_1, \dots, x_D)$ as a D -dimensional array of binary or bit variables $x_i \in \{0, 1\}$.

Therefore, there are 2^D possible data configurations in the ensemble $\mathcal{X}$. Similarly, a hidden variable $\mathbf{z} = (z_1, \dots, z_L)$ is described by its components $z_i \in \{0, 1\}$ and has a size $L \ll D$.

A state $\mathbf{z}$ can jump to any of the $\mathbf{x}$ states via a Markov matrix A , defined in (1), whose element A_{zx} represents the probability of jumping from $\mathbf{z}$ to $\mathbf{x}$. Similarly, a state $\mathbf{x}$ can transition back to any of the $\mathbf{z}$ states through a Markov matrix $\bar{A}$ [Eq. (2)], whose element $\bar{A}_{xz}$ gives the probability of jumping from $\mathbf{x}$ to $\mathbf{z}$. In both cases, the conditions

$$\sum_{x \in \mathcal{X}} A_{zx} = 1, \quad \forall \mathbf{z}, \quad \sum_{z \in \mathcal{Z}} \bar{A}_{xz} = 1, \quad \forall \mathbf{x}, \quad (4)$$

ensure the normalization of each row of the Markov matrices. We denote as $p(\mathbf{x})$ the stationary probability of visible states and as $\bar{p}(\mathbf{z})$ the stationary probability of hidden states. Given the transition matrices A and $\bar{A}$ they satisfy the following equations:

$$p(\mathbf{x}) = \sum_{z \in \mathcal{Z}} \bar{p}(\mathbf{z}) A_{zx}, \quad (5)$$

$$\bar{p}(\mathbf{z}) = \sum_{x \in \mathcal{X}} p(\mathbf{x}) \bar{A}_{xz}. \quad (6)$$

Following common practice in unsupervised machine learning, we train the model by maximizing the log-likelihood

$$\begin{aligned} \mathcal{L} &= \frac{1}{N_D} \sum_m \ln p(\mathbf{x}^{(m)}) \\ &= \sum_{\mathbf{x}} p_{\text{data}}(\mathbf{x}) \ln p(\mathbf{x}) \\ &= \langle \ln p(\mathbf{x}) \rangle_{\mathbf{x} \in \mathcal{D}}, \end{aligned} \quad (7)$$

averaged in the dataset $\mathcal{D} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N_D)})$.

The gradient descent to maximize it is performed on the parameters that define the matrices A and $\bar{A}$. For these Markov matrices, we chose the exponential structure typically used for RBMs:

$$A_{zx} = \frac{e^{\sum_{\alpha,i} x_i w_{i\alpha} z_{\alpha} + \sum_i x_i a_i}}{\sum_{x'} e^{\sum_{\alpha,i} x'_i w_{i\alpha} z_{\alpha} + \sum_i x'_i a_i}} = \frac{e^{\mathbf{x} \cdot (\mathbf{w} \cdot \mathbf{z} + \mathbf{a})}}{\sum_{x'} e^{\mathbf{x}' \cdot (\mathbf{w} \cdot \mathbf{z} + \mathbf{a})}}, \quad (8)$$

$$\bar{A}_{xz} = \frac{e^{\sum_{\alpha,i} z_{\alpha} \bar{w}_{\alpha i} x_i + \sum_{\alpha} z_{\alpha} \bar{a}_{\alpha}}}{\sum_{z'} e^{\sum_{\alpha,i} z'_{\alpha} \bar{w}_{\alpha i} x_i + \sum_{\alpha} z'_{\alpha} \bar{a}_{\alpha}}} = \frac{e^{\mathbf{z} \cdot (\bar{\mathbf{w}} \cdot \mathbf{x} + \bar{\mathbf{a}})}}{\sum_{z'} e^{\mathbf{z}' \cdot (\bar{\mathbf{w}} \cdot \mathbf{x} + \bar{\mathbf{a}})}}. \quad (9)$$

Here, we denote the local biases for visible and hidden units by $\mathbf{a}$ and $\bar{\mathbf{a}}$, respectively. The weights are given by two separate matrices: $\mathbf{w}$ of size $D \times L$ for the transitions $\mathbf{z} \rightarrow \mathbf{x}$, and $\bar{\mathbf{w}}$ of size $L \times D$ for the transitions $\mathbf{x} \rightarrow \mathbf{z}$. Note that, unlike RBMs, $\mathbf{w}$ and $\bar{\mathbf{w}}$ are not transposes of each other.

The exponential parametrization of the transition matrices allows for bitwise factorization,

$$A_{zx} = \prod_{i=1}^D \frac{e^{x_i (\sum_{\alpha} w_{i\alpha} z_{\alpha} + a_i)}}{e^{\sum_{\alpha} w_{i\alpha} z_{\alpha} + a_i} + 1} = \prod_{i=1}^D p(x_i | \mathbf{z}). \quad (10)$$

Similarly, the transition probabilities from $\mathbf{x}$ to $\mathbf{z}$ factor as

$$\bar{A}_{xz} = \prod_{\alpha=1}^L \frac{e^{z_{\alpha} (\sum_i \bar{w}_{\alpha i} x_i + \bar{a}_{\alpha})}}{e^{\sum_i \bar{w}_{\alpha i} x_i + \bar{a}_{\alpha}} + 1} = \prod_{\alpha=1}^L p(z_{\alpha} | \mathbf{x}). \quad (11)$$

These two factorizations enable efficient sampling of the Markov chain: the first equation allows sampling each component x_i given the hidden state z , and the second one allows sampling z_α given $\mathbf{x}$.

III. GRADIENT APPROXIMATION

In principle, we should evaluate the derivatives of $\ln p(\mathbf{x})$ with respect to the parameters that define the matrices $\mathbf{w}$, $\bar{\mathbf{w}}$, and the vectors $\mathbf{a}$, $\bar{\mathbf{a}}$, computed on the data points $\mathbf{x}$. The stationary distribution $p(\mathbf{x})$ is the eigenvector associated with the eigenvalue 1 of the matrix $\sum_z \bar{A}_{x'z} A_{zx}$. However, expressing $p(\mathbf{x})$ explicitly in terms of parameters and then differentiating it is too complex. We therefore introduce an approximation that restricts the dependence on the parameters to the last two steps of the Markov chain:

$$p(\mathbf{x}) = \sum_z \bar{p}(z) A_{zx} = \sum_z \sum_{x'} A_{zx} \bar{A}_{x'z} p(\mathbf{x}'). \quad (12)$$

In particular, the dependence on the parameters that define $\mathbf{w}$ and $\mathbf{a}$ is confined to the matrix A appearing in the second term of the equation, whereas the dependence on $\bar{\mathbf{w}}$ and $\bar{\mathbf{a}}$ is limited to the matrix $\bar{A}$ in the third term.

To compute the derivatives of the log-likelihood $\mathcal{L}$, it is useful to first compute the partial derivatives of the two Markov matrices with respect to their parameters,

$$\partial_{w_{j\beta}} A_{zx} = A_{zx} z_\beta (x_j - \mu_{x_j|z}), \quad (13a)$$

$$\partial_{a_j} A_{zx} = A_{zx} (x_j - \mu_{x_j|z}), \quad (13b)$$

$$\partial_{\bar{w}_{\beta j}} \bar{A}_{xz} = \bar{A}_{xz} x_j (z_\beta - \mu_{z_\beta|\mathbf{x}}), \quad (13c)$$

$$\partial_{\bar{a}_\beta} \bar{A}_{xz} = \bar{A}_{xz} (z_\beta - \mu_{z_\beta|\mathbf{x}}), \quad (13d)$$

where $\mu_{x_j|z} = \sum_{x'} x'_j A_{zx}$ is the average of component j of the vector $\mathbf{x}'$ given z ; indeed, A_{zx} is the conditional probability of $\mathbf{x}'$ given z . Similarly, $\mu_{z_\beta|\mathbf{x}} = \sum_{z'} z'_\beta \bar{A}_{xz'}$ is the average of component β of the vectors $\mathbf{z}'$ given $\mathbf{x}$.

Thanks to the factorized exponential structure chosen for the parametrization [see (10) and (11)], these two quantities have simple analytical expressions:

$$\mu_{x_j|z} = \frac{1}{1 + e^{-(\mathbf{w} \cdot \mathbf{z} + \mathbf{a})_j}}, \quad (14)$$

$$\mu_{z_\beta|\mathbf{x}} = \frac{1}{1 + e^{-(\bar{\mathbf{w}} \cdot \mathbf{x} + \bar{\mathbf{a}})_\beta}}. \quad (15)$$

Note that (14) depends only on z , while (15) depends only on $\mathbf{x}$.

The gradient components of a point's log-likelihood $\ln p(\mathbf{x})$ with respect to $w_{j\beta}$ or a_j can be computed using $p(\mathbf{x}) = \sum_z \bar{p}(z) A_{zx}$, under the approximation that $\bar{p}(z)$ is independent of $w_{j\beta}$ or a_j . From Eqs. (13a) and (13b), we have

$$\partial_{w_{j\beta}} \ln p(\mathbf{x}) = \frac{\langle A_{zx} z_\beta (x_j - \mu_{x_j|z}) \rangle_{\bar{p}(z)}}{\langle A_{zx} \rangle_{\bar{p}(z)}}, \quad (16a)$$

$$\partial_{a_j} \ln p(\mathbf{x}) = \frac{\langle A_{zx} (x_j - \mu_{x_j|z}) \rangle_{\bar{p}(z)}}{\langle A_{zx} \rangle_{\bar{p}(z)}}. \quad (16b)$$

Here, $\mathbf{x}$ is a given point while $\langle \cdot \rangle_{\bar{p}(z)}$ denotes the average over the z drawn from the model. The same applies to the derivatives with respect to $\bar{w}_{\beta j}$ or $\bar{a}_\beta$, where one should

use $p(\mathbf{x}) = \sum_z \sum_{x'} A_{zx} \bar{A}_{x'z} p(\mathbf{x}')$, together with Eqs. (13c) and (13d):

$$\partial_{\bar{w}_{\beta j}} \ln p(\mathbf{x}) = \frac{\langle A_{zx} x'_j (z_\beta - \mu_{z_\beta|\mathbf{x}}) \rangle_{p(\mathbf{x}',z)}}{\langle A_{zx} \rangle_{\bar{p}(z)}}, \quad (16c)$$

$$\partial_{\bar{a}_\beta} \ln p(\mathbf{x}) = \frac{\langle A_{zx} (z_\beta - \mu_{z_\beta|\mathbf{x}}) \rangle_{p(\mathbf{x}',z)}}{\langle A_{zx} \rangle_{\bar{p}(z)}}. \quad (16d)$$

Here, $\mathbf{x}$ is a given point, while $\langle \cdot \rangle_{p(\mathbf{x}',z)}$ denotes the average over $\mathbf{x}'$ and z drawn from the model, with $p(\mathbf{x}',z) = p(\mathbf{x}') A_{x'z}$.

To evaluate the approximate gradient, two averages must be taken into account. The first corresponds to the data average over the true samples $\mathbf{x}$ in (7) applied to the approximate gradient components in (16). This is done by randomly selecting a minibatch $\mathcal{N} \subset \mathcal{D}$ of N data points (with $N \ll N_D$). In practice, this replaces the log-likelihood of full data, $\mathcal{L}$, with a stochastic estimate $\mathcal{L}_{\mathcal{N}}$ [1].

We also adopt the so-called “centering trick” [35], which replaces the data $\mathbf{x}$ with their deviation from the data average, $\mathbf{x} - \langle \mathbf{x} \rangle_{x \in \mathcal{D}}$, in the exponents of the transition matrix $\bar{A}_{xz}$. In practice, this modifies the gradient (16c) into

$$\partial_{\bar{w}_{\beta j}} \ln p(\mathbf{x}) = \frac{\langle A_{zx}, (x'_j - \langle x_j \rangle_{x \in \mathcal{D}}) (z_\beta - \mu_{z_\beta|\mathbf{x}}) \rangle_{p(\mathbf{x}',z)}}{\langle A_{zx} \rangle_{\bar{p}(z)}}. \quad (17)$$

An analogous variant that removes the averages of the hidden variables was not necessary. For RBMs, the centering trick stabilizes learning and makes it independent of the binarization convention (0, 1 or 1, 0). We adopt this trick because we find it reduces the likelihood that some hidden units of our model never activate.

The second average is taken over the model distribution, i.e., over the states $\mathbf{x}'$ and the hidden variables z obtained from steady-state sampling of the Markov chain defined by the transition matrices A and $\bar{A}$, as sketched in Fig. 1. In practice, using (10) and (11), we run a Markov chain for T double steps, producing the sequence

$$\mathcal{T} = [\mathbf{x}'(0), z(0), \mathbf{x}'(1), z(1), \dots, \mathbf{x}'(T-1), z(T-1)]. \quad (18)$$

The prime notation reminds us that these $\mathbf{x}'$ are “fantasy particles” generated by the model and not true data. To ensure efficient sampling, we employ *persistent contrastive divergence*. The final fantasy state $\mathbf{x}'(T)$ is reused as the initial state $\mathbf{x}'(0)$ for the next Markov chain, updated with the new transition matrices. This persistent initialization keeps the chain close to its steady state because the parameters evolve only slowly.

After computing the stochastic gradients of the log-likelihood, we update the transition matrices using a stochastic gradient-ascent step with the RMSprop method, tuned with an appropriate learning rate. A new iteration then restarts with a new minibatch.

A. Exact gradient and fixed points

To clarify the effect of the approximation, it is useful to decompose the exact derivative of the log-likelihood into two contributions. Let $K_\theta \equiv \bar{A}_\theta A_\theta$ denote the Markov operator acting on visible states (one full $\mathbf{x} \rightarrow z \rightarrow \mathbf{x}$ double step), so

that their stationary distribution satisfies

$$p_\theta = p_\theta K_\theta. \quad (19)$$

Here, for clarity, we make explicit the dependence of the operators and of the stationary distribution on the model parameters θ , which is otherwise left implicit throughout the paper. Using this fixed-point relation, the log-likelihood can be written as

$$\begin{aligned} \mathcal{L}(\theta) &= \sum_{\mathbf{x}} p_{\text{data}}(\mathbf{x}) \ln p_\theta(\mathbf{x}) \\ &= \sum_{\mathbf{x}} p_{\text{data}}(\mathbf{x}) \ln [(p_\theta K_\theta)(\mathbf{x})]. \end{aligned} \quad (20)$$

Differentiating and using $p_\theta K_\theta = p_\theta$ yields

$$\partial_\theta \mathcal{L} = \sum_{\mathbf{x}} p_{\text{data}}(\mathbf{x}) \frac{\partial_\theta (p_\theta K_\theta)(\mathbf{x})}{p_\theta(\mathbf{x})}. \quad (21)$$

The derivative splits as

$$\partial_\theta (p_\theta K_\theta) = p_\theta (\partial_\theta K_\theta) + (\partial_\theta p_\theta) K_\theta, \quad (22)$$

leading to

$$\partial_\theta \mathcal{L} = \sum_{\mathbf{x}} \frac{p_{\text{data}}(\mathbf{x})}{p_\theta(\mathbf{x})} \{ [p_\theta (\partial_\theta K_\theta)](\mathbf{x}) + [(\partial_\theta p_\theta) K_\theta](\mathbf{x}) \}. \quad (23)$$

The approximation used in this work amounts to neglecting the second term.

In the well-specified regime where $p_\theta = p_{\text{data}}$, the weights $p_{\text{data}}(\mathbf{x})/p_\theta(\mathbf{x})$ reduce to unity, and the two contributions in (23) vanish separately upon summation over $\mathbf{x}$. The first term vanishes because the derivative of the row sums of a Markov matrix is zero, implying

$$\sum_{\mathbf{x}'} [p_\theta (\partial_\theta K_\theta)](\mathbf{x}') = 0. \quad (24)$$

The second term vanishes because normalization of the stationary distribution implies

$$\sum_{\mathbf{x}} \partial_\theta p_\theta(\mathbf{x}) = 0. \quad (25)$$

Therefore, in this regime, the exact and the approximate gradients share the same fixed points. Away from the optimum (or in misspecified settings where $p_\theta \neq p_{\text{data}}$), the two gradients may differ, although the approximate rule provides stable and effective training in practice.

IV. EXAMPLE

A. Data and parameters

We tested the proposed unsupervised machine learning method on a selected part of the MNIST dataset of handwritten digits [36]. From such visually interpretable data, we extract the first 1000 instances of the digits 0, 1, and 2. Hence, the dataset contains $N_D = 3000$ digits, equally balanced in three classes, and is complex yet small enough for a precise estimate of the log-likelihood. We then digitize each sample, assigning $x_i = 1$ to pixels with a gray level above 50% and $x_i = 0$ otherwise. As a result, our data point $\mathbf{x}^{(m)}$ is a sequence of $D = 784$ bits, which can be visualized as an image by reshaping them into a matrix 28×28 .

We train a model with $L = 5$ hidden units, using an RM-Sprop gradient ascent method with a learning rate $\eta = 0.1$. We monitor the learning process by checking the state of the system every 30 gradient ascent steps, a span that we term an *epoch*. During the run, composed of 120 epochs, we schedule the size of the minibatch to increase linearly from $N = 10$ to $N = 50$. In each step, we run a Markov chain with $T = 40$ steps. To gather statistics, we train 30 independent models that differ only in their random initialization.

We use the general (compositional) hidden representation introduced in Sec. II: each of the L hidden units is a Bernoulli variable activated in a $\mathbf{x} \rightarrow \mathbf{z}$ step with probability $p(z_\alpha = 1 | \mathbf{x}) = \mu_{z_\alpha | \mathbf{x}}$ from (15), so that the hidden state $\mathbf{z}$ explores all 2^L binary configurations. The visible variables are centered on their empirical data mean to stabilize the gradient updates. The initialization of weights is described in the Appendix, together with other technical details on stabilization of the code and computation of the log-likelihood.

B. Results

Our goal is to understand how the generative performance of the Markov-matrix architecture relates to the dynamical properties of the Markov chain it induces, and in particular to how far its stationary state lies from equilibrium.

As a first measure of generative performance, we monitor the log-likelihood $\mathcal{L}$ maximized during the training of 30 models [Fig. 2(a)]. As a complementary and more stringent measure of generative quality, we also quantify how faithfully the model reproduces the class statistics of the data through the KL divergence between the true class distribution P^D and the one generated by the model, P^e ,

$$\text{KL}(P^D \| P^e) = \sum_{k \in \{0,1,2\}} P^D(k) \ln \frac{P^D(k)}{P^e(k)}, \quad (26)$$

where $P^D(k) = 1/3$ reflects the balanced three-class dataset and $P^e(k)$ is the stationary occupancy of the hidden chain projected onto the digit classes,

$$P^e(k) = \sum_{\mathbf{z}} \bar{p}(\mathbf{z}) \delta_{k, c(\mathbf{z})}, \quad (27)$$

with $\bar{p}(\mathbf{z})$ the empirical stationary probability of state $\mathbf{z}$ and $c(\mathbf{z}) \in \{0, 1, 2\}$ the digit class assigned to it. The assignment compares the mean image generated by state $\mathbf{z}$, i.e., the conditional average $\mu_{x_i | \mathbf{z}}$ of (14), with the empirical mean image $\langle x_i \rangle_{\mathbf{x} \in \mathcal{D}_k}$ of each class (the average over the data points of class k), and selects the best-overlapping one,

$$c(\mathbf{z}) = \arg \max_{k \in \{0,1,2\}} \sum_{i=1}^D \mu_{x_i | \mathbf{z}} \langle x_i \rangle_{\mathbf{x} \in \mathcal{D}_k}. \quad (28)$$

The same map $c(\mathbf{z})$ defines the class-projected transition matrix $\tilde{M}_{c(\mathbf{z}'), c(\mathbf{z})}$ shown in Fig. 3. To characterize the dynamics, we then follow the trajectories of the hidden units $\mathbf{z}$ and empirically estimate the transition matrix $M_{\mathbf{z}' \mathbf{z}}$ of Eq. (3) by counting the transitions on long trajectories and normalizing each row. The stationary distribution $\bar{p}(\mathbf{z})$ is likewise obtained empirically. As a diagnostic of time-reversal symmetry breaking in the nonequilibrium steady state, we compute the entropy

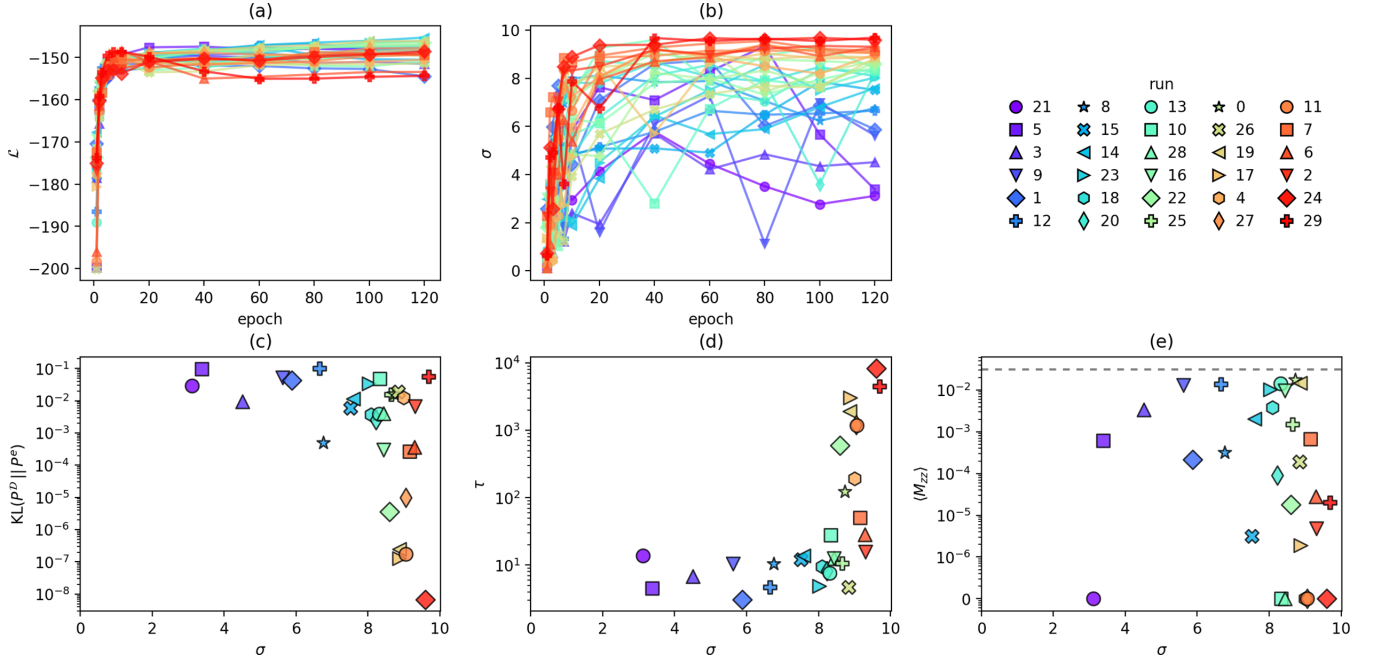


FIG. 2. Summary of the 30 independently trained models: (a) Log-likelihood $\mathcal{L}$ and (b) entropy production rate σ as functions of the training epoch. The color of each run encodes its final σ . The remaining panels report, at the last epoch and as functions of σ : (c) the KL divergence (26) between the true and generated class distributions; (d) the decorrelation time (30); and (e) the mean diagonal element of the hidden transition matrix, always below the random baseline $1/2^L$ (dashed line). Low KL is achieved only at large σ .

production rate

$$\sigma = \sum_{z' < z} [\bar{p}(z')M_{z'z} - \bar{p}(z)M_{zz'}] \ln \frac{\bar{p}(z')M_{z'z}}{\bar{p}(z)M_{zz'}}. \quad (29)$$

Figure 2(b) shows the evolution of σ during training: the models reach comparable log-likelihoods but spread over a wide range of positive final entropy production, none approaching the reversible limit $\sigma = 0$. While $\mathcal{L}$ displays no systematic dependence on σ , Fig. 2(c) shows that the KL divergence may become very low only as σ grows. The benefit of operating far from equilibrium therefore emerges as a more faithful reproduction of the data classes (a lower KL).

To inspect in detail the features learned by the model, Fig. 3 shows two representative runs, selected by their final entropy production σ and class KL divergence: one with large σ and very low KL [Fig. 3(a)] and one with smaller σ and higher KL [Fig. 3(b)]. Each panel displays the learned weights $\mathbf{w}$, $\bar{\mathbf{w}}$, $\bar{\mathbf{a}}$ and the visible bias $\mathbf{a}$, the class-projected transition matrix $\bar{M}$ between the digit classes, the per-unit average activations indicated above the $\mathbf{w}$ panels, and, in the lower row, the average images generated along a sampled sequence of hidden states $(z(0), z(1), \dots)$.

In the high- σ , low-KL run [Fig. 3(a)], the hidden units capture clean, interpretable digit features and the three classes are almost equally represented. The class-projected matrix $\bar{M}$ has a dominant off-diagonal entry in every row, producing strongly irreversible trajectories that cycle regularly through the digits. In the low- σ run [Fig. 3(b)], the dynamics displays only partial directionality and switches between short subcycles. This added stochasticity may underlie the observed increase in the KL divergence.

We confirm this picture using additional dynamical indicators. First, to quantify the regularity of the dynamics, we

compute the decorrelation time

$$\tau = -1/\ln |\lambda_2|, \quad (30)$$

where λ_2 is the second-largest eigenvalue (in modulus) of M . As shown in Fig. 2(d), τ correlates strongly with σ : models with large entropy production develop long correlation times, reaching up to $\tau \approx 10^4$ steps, consistent with nearly deterministic cycles.

To further assess dynamical structure, we examine the mean diagonal element of the transition matrix,

$$\langle M_{zz} \rangle = \frac{\text{Tr}(M)}{2^L}. \quad (31)$$

Figure 2(e) shows that no trained model exhibits self-transition probabilities that exceed the random baseline $1/2^L$. Even low- σ models satisfy $\langle M_{zz} \rangle < 1/2^L$, indicating that training consistently leads to nonequilibrium steady states in which successive hidden states, and consequently the generated fantasy samples, are more decorrelated than in independent random sampling.

V. CONCLUSIONS

We introduced a generative model in which visible and latent variables interact through two independently parametrized transition kernels forming a nonequilibrium Markov chain. In comparison with classical RBM equilibrium sampling, this construction breaks the detailed balance between the forward and backward conditional transitions, which leads to qualitatively new behavior. Likelihood maximization drives the system toward nonequilibrium steady states characterized by irreversible probability currents,

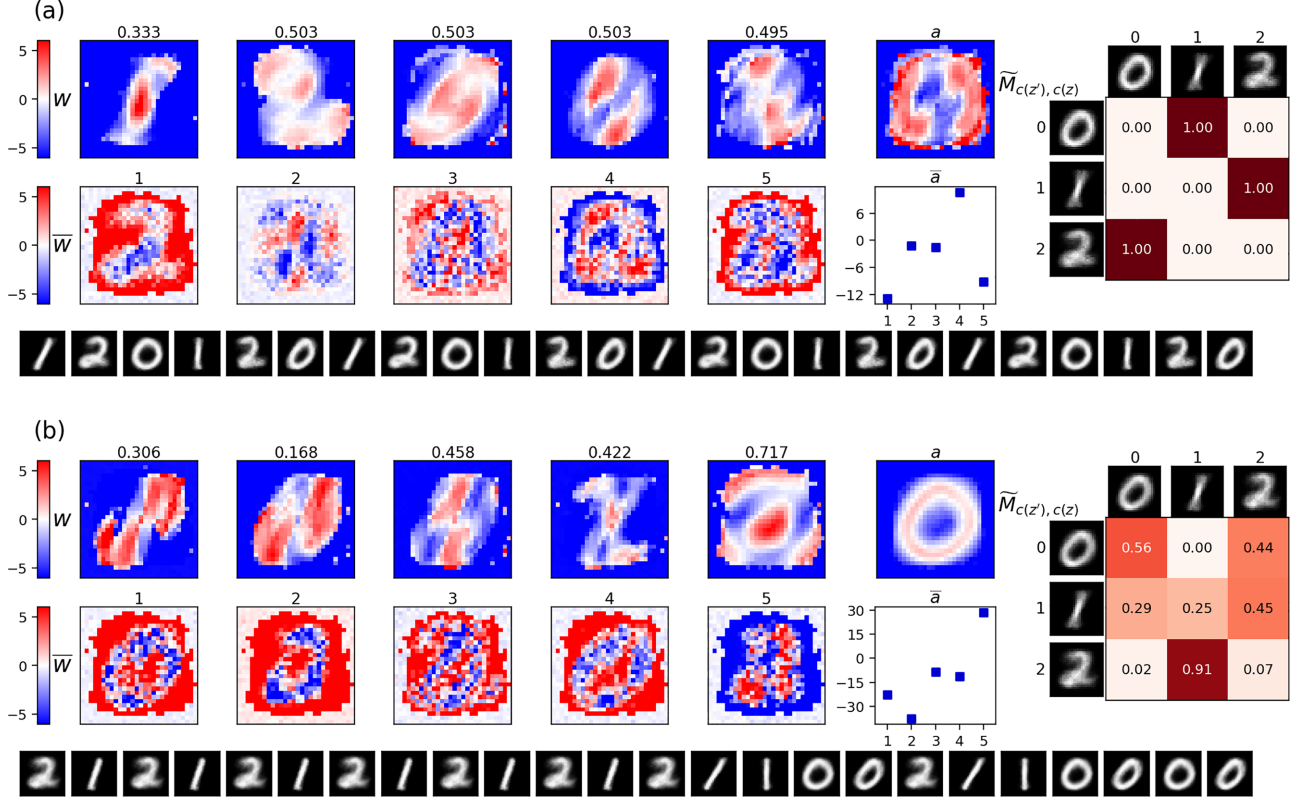


FIG. 3. Two representative trained models, selected for their final entropy production σ and class-distribution KL divergence: (a) run 24, with high σ and low KL and (b) run 3, with low σ and high KL. Each panel shows the weights $w_{i\alpha}$ for the L hidden units (first row, D -dimensional arrays reshaped to 28×28 images, with the empirical unit occupancy reported on top), the corresponding weights $\bar{w}_{\alpha i}$ (second row), the hidden biases $\bar{a}$ and the visible bias a (right), and the class-projected transition matrix $\tilde{M}_{c(z'), c(z)}$ between digit classes $\{0, 1, 2\}$ (far right; each row/column by the class' representative mean image). The bottom row shows a sequence of mean generated images $\mu_{x|z(t)}$ along a sampled trajectory of hidden states. In the high- σ , low-KL run (a) the dynamics cycles regularly $1 \rightarrow 2 \rightarrow 0 \rightarrow 1 \dots$, while the lower- σ run (b) cycles less coherently.

reduced self-transition probabilities, and the emergence of latent-state cycles.

A quantitative comparison with standard RBMs trained on the same dataset (reported in Appendix B) shows that the RBMs reach slightly lower log-likelihoods and, when used as a generative model, tend to remain trapped in a single digit for many consecutive samples. In contrast, our model naturally develops cyclic latent dynamics that promotes a regular alternation between digits.

These cycles are not imposed *a priori* but arise spontaneously during training. Models that develop a clearer cyclic structure exhibit higher entropy production, longer decorrelation times, and more regular alternation through latent modes. At the same time, they reproduce more faithfully the empirical distribution of classes in the dataset (with a lower KL) even though the log-likelihood itself does not single out these far-from-equilibrium solutions. This suggests that cycling improves the quality of the generative process by decorrelating successive fantasy samples and encouraging a fuller exploration of the data modes.

Our findings bring a nonequilibrium perspective to problems that are usually tackled using equilibrium RBM approaches. Several works focused on understanding and improving the slow sampling caused by the energy landscape that RBMs progressively sculpt during their learning

[8–10, 37–39], clarifying the equilibrium and nonequilibrium regimes that arise during learning [40, 41] and the dynamical transitions that can shape training trajectories in energy-based models [42]. These approaches improve the efficiency of Monte Carlo sampling while preserving the detailed balance. In contrast, the mechanism uncovered in this work relies on explicitly breaking reversibility: the most performant models are those whose latent dynamics operates far from equilibrium and organizes into directed cycles.

Recent developments in diffusion-based generative models have highlighted similar connections between learning, sampling efficiency, and nonequilibrium thermodynamics. In particular, entropy production has been shown to constrain both the speed and accuracy of diffusion model generation [22]. Our findings confirm that nonequilibrium principles can be exploited to design generative models: introducing controlled irreversibility at the latent level can improve exploration and generative performance.

In summary, this work provides a proof of principle that nonequilibrium dynamics can enhance unsupervised generative modeling, even in the minimal setting of a two-layer architecture. The learning rule derived here relies on a log-likelihood gradient that depends on the last two steps of the Markov chain, a structure absent from equilibrium energy-based models. Our results open several directions for future

research, including extensions to continuous latent spaces, deeper chains, and a systematic investigation of whether nonequilibrium cycles can provide computational advantages in large-scale generative tasks.

ACKNOWLEDGMENTS

The authors are particularly grateful to Marco Boscolo for identifying a bug in an earlier version of the code, and would like to thank him, Manjodh Singh, Aurélien Decelle, Francesco La Rovere, Nicolò Montagner, Raffaele Sabatini, Beatriz Seoane, Peter Sollich, and Lorenzo Rosset for the many fruitful discussions. M.B. acknowledges support for visits to LPTMS from the Centre National de la Recherche Scientifique (CNRS) and Université Paris-Saclay, and thanks LPTMS and LPTHE (Sorbonne Université) for their hospitality.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available. The data are available from the authors upon reasonable request.

APPENDIX A: TECHNICAL DETAILS

1. Initialization of weights

The backward weights and biases $\bar{\mathbf{w}}$ and $\bar{\mathbf{a}}$ are set to zero so that the hidden states $\mathbf{z}$ are sampled uniformly during the early stages of learning. The forward weights $\mathbf{w}$ are initialized to small independent Gaussian values,

$$w_{i\alpha} \sim \mathcal{N}(0, \sigma_0^2), \quad \sigma_0 = \frac{2}{\sqrt{D+L}},$$

so that every hidden unit starts close to the uninformative point and the units differentiate spontaneously during learning. The visible bias $\mathbf{a}$ is initialized to zero and learned together with $\mathbf{w}$.

2. Factorized transition probabilities

A numerically more stable variant replaces $A_{\mathbf{zx}}$ by

$$A_{\mathbf{zx}}^* = \prod_{i=1}^D \left[\frac{e^{x_i (\sum_{\alpha} w_{i\alpha} z_{\alpha} + a_i)}}{e^{\sum_{\alpha} w_{i\alpha} z_{\alpha} + a_i} + 1} Q \right], \quad (\text{A1})$$

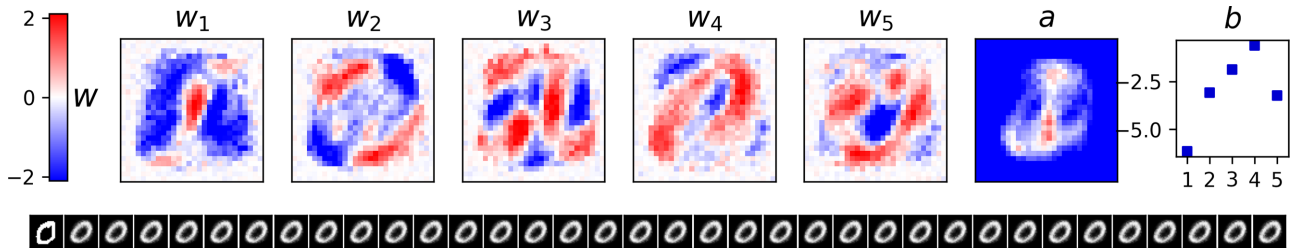


FIG. 4. A standard RBM trained on the same dataset with $L = 5$ binary hidden units, shown for comparison with the nonequilibrium model. The first five panels display the learned weight maps $w_{i\alpha}$ of the hidden units, followed by the trained visible bias a and the hidden bias b (rightmost scatter). The bottom row shows a trajectory sampled with contrastive divergence, starting from a data point “0”: the chain remains trapped in the same digit mode for the whole shown sequence, illustrating the slow mixing of RBM sampling caused by energy barriers in the learned landscape [8–10].

where $Q \approx 1.5$ – 1.8 compensates for typical sigmoid magnitudes. The ratios $A_{\mathbf{zx}}^*/\langle A_{\mathbf{zx}}^* \rangle_{\bar{p}(\mathbf{z})}$ can then replace $A_{\mathbf{zx}}/\langle A_{\mathbf{zx}} \rangle_{\bar{p}(\mathbf{z})}$ in gradient computations, while the log-likelihoods computed with A^* include a correction term $-D \ln Q$. In the numerical experiments, the hidden units are sampled independently from the factorized Bernoulli form (11), so that the latent state explores all 2^L compositional configurations.

For the numerical experiments in the next section, we focus on nonequilibrium steady states with a one-hot latent representation. The corresponding transition probabilities $\bar{A}_{\mathbf{zx}}$ are then normalized in a softmax fashion.

3. Efficient log-likelihood estimation

A fast on-the-fly estimate of the log-likelihood for a mini-batch $\mathcal{N}$ is obtained from

$$\mathcal{L}_{\mathcal{N}} \approx \frac{1}{N} \sum_{\mathbf{z} \in \mathcal{N}} \ln \langle A_{\mathbf{zx}} \rangle_{\mathbf{z}},$$

where the average is over sampled $\mathbf{z}$ states and time steps t . A more precise estimate uses long Markov chains (e.g., $T = 10^5$) and all $N_{\mathcal{D}}$ data points, which requires evaluating $T \times N_{\mathcal{D}}$ transition probabilities. When the latent space is small, instead, one can use the empirical occupancy $\bar{p}_e(\mathbf{z})$ obtained from the T Markov samples and compute each $\ln p(\mathbf{x})$ using (5). This reduces the number of evaluations to $2^L \times N_{\mathcal{D}}$ or $L \times N_{\mathcal{D}}$, feasible because the latent space is small ($2^L = 32$ for $L = 5$).

APPENDIX B: COMPARISON WITH A STANDARD RBM

For comparison, we trained a standard RBM on the same dataset, with the same number $L = 5$ of binary hidden units, using RMSprop together with contrastive divergence with $k = 10$ Gibbs steps and the centering trick [35].

The RBM defines a joint distribution over visible and hidden configurations $(\mathbf{x}, \mathbf{z})$

$$p_{\theta}(\mathbf{x}, \mathbf{z}) = \frac{e^{-E_{\theta}(\mathbf{x}, \mathbf{z})}}{Z_{\theta}}, \quad Z_{\theta} = \sum_{\mathbf{x}, \mathbf{z}} e^{-E_{\theta}(\mathbf{x}, \mathbf{z})}.$$

With binary visible and hidden units, the energy reads

$$E_{\theta}(\mathbf{x}, \mathbf{z}) = - \sum_i a_i x_i - \sum_{\mu} b_{\mu} z_{\mu} - \sum_{i, \mu} x_i w_{i\mu} z_{\mu}.$$

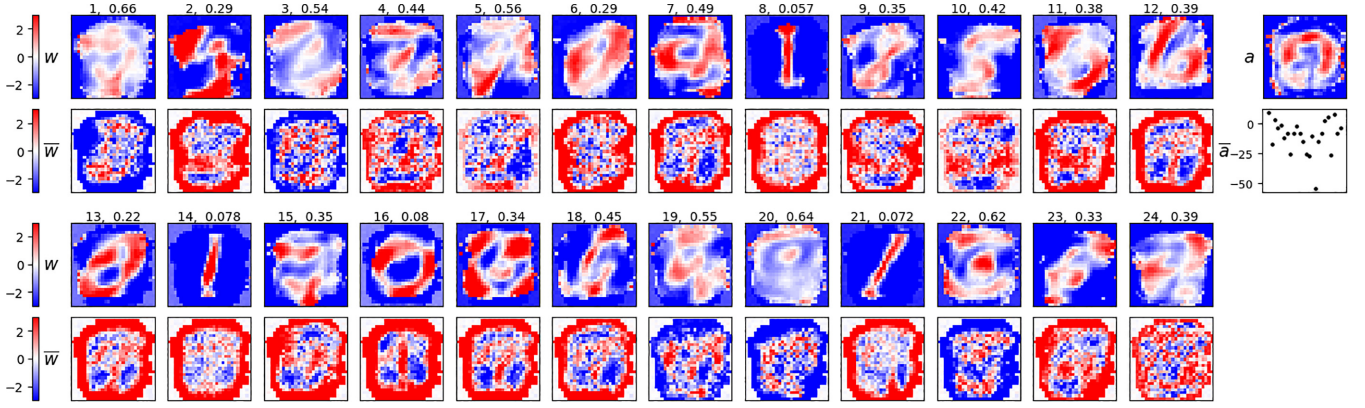


FIG. 5. Weights w (odd rows), $\bar{w}$ (even rows), and visible bias a and hidden bias $\bar{a}$ (top-right panels) learned by the model using $L = 24$ hidden units and trained on MNIST. Labels above the w panels show the unit index and average activation probability in the Markov chain.

The visible distribution is obtained by marginalizing over the hidden variables,

$$p_{\theta}(\mathbf{x}) = \sum_{\mathbf{z}} p_{\theta}(\mathbf{x}, \mathbf{z}).$$

The log-likelihood of the dataset is defined as

$$\mathcal{L}_{\theta} = \frac{1}{N_{\mathcal{D}}} \sum_{\mathbf{x} \in \mathcal{D}} \ln p_{\theta}(\mathbf{x}).$$

Because the number of hidden units is small ($L = 5$), the log-likelihood can be computed exactly by summing over hidden configurations. Taking advantage of the factorized form of the RBM energy function leads to

$$Z_{\theta} = \sum_{\mathbf{z}} e^{\sum_{\mu} b_{\mu} z_{\mu}} \prod_i (1 + e^{a_i + \sum_{\mu} w_{i\mu} z_{\mu}}),$$

which allows a direct evaluation of $p_{\theta}(\mathbf{x})$ and therefore of $\mathcal{L}_{\theta}$.

With $L = 5$ binary hidden units, the RBM reaches a best average log-likelihood of about $\mathcal{L} \approx -154$, below the $\mathcal{L} \approx -145$ to -148 attained by our nonequilibrium Markov-matrix model on the same data. A representative realization with the best log-likelihood is shown in Fig. 4: its bottom panel shows that, when used as a generative model, the RBM frequently becomes trapped in a single digit mode for many consecutive samples, consistent with the well-known slow mixing of RBM sampling caused by energy barriers in the learned energy landscape [8–10]. By contrast, our model cycles regularly through the digit classes (Fig. 3), reproducing them more faithfully.

APPENDIX C: SCALABILITY TEST ON MNIST

To evaluate how well the model scales beyond the illustrative examples presented in the main text, we trained it on a subset of the MNIST dataset that includes all ten handwritten digit classes (using the first 500 samples from each class). The purpose of this experiment is not to reach state-of-the-art



FIG. 6. Markov chain generated by the model shown in Fig. 5. Each panel shows the visible configuration together with the hidden binary pattern that generated it, displayed above the image, which is the digitized average output of that hidden configuration. The sequence progresses row by row and illustrates how digits emerge from combinations of shared hidden features, and confirms that the model is capable of generating sequences of independent digits.

performance but rather to confirm that the learning dynamics remain stable and that the model can extract compositional features in a realistic, high-dimensional scenario.

The model uses $L = 24$ hidden units that encode 2^{24} possible configurations. Its parameters are trained using RMSprop with a learning rate of 0.02. Training involves 12 000 gradient update steps, with minibatch sizes progressively increasing from 40 to 240 data samples. The Markov chains used to generate fantasy samples have length $T = 50$ during the first half of training, which is then increased to $T = 100$ for the second half.

Figure 5 shows the learned parameters. Each hidden unit develops a spatial structure that resembles the features of elementary digits, such as strokes or loops. Many hidden units are activated across multiple digits, indicating that the model

learns reusable visual primitives rather than digit-specific templates.

To illustrate the generative behavior of the model, Fig. 6 shows a trajectory of the Markov chain. The binary strings displayed above each image indicate which hidden units are active in that configuration. The digit shown corresponds to the digitized average output associated with that hidden configuration, which helps visualize the corresponding feature composition. The sequence qualitatively suggests that the dynamics operates out of equilibrium. Certain digit sequences appear very frequently (e.g., from $1 \rightarrow 6 \rightarrow 7$ or $1 \rightarrow 2 \rightarrow 7$). Such directional preferences indicate the presence of probability currents in configuration space, consistent with the nonequilibrium character of the model discussed in the main text.

-
- [1] P. Mehta, M. Bukov, C.-H. Wang, A. G. Day, C. Richardson, C. K. Fisher, and D. J. Schwab, A high-bias, low-variance introduction to machine learning for physicists, *Phys. Rep.* **810**, 1 (2019).
 - [2] P. Smolensky, Information Processing in Dynamical Systems: Foundations of Harmony Theory, in *Parallel Distributed Processing*, edited by D. E. Rumelhart and J. L. McClelland (MIT Press, Cambridge, 1986), Vol. 1, pp. 194–281.
 - [3] G. E. Hinton, A practical guide to training restricted Boltzmann machines, *Neural Networks: Tricks of the Trade*, 2nd ed. (Springer, Berlin, 2012), pp. 599–619.
 - [4] C. Malbrancke, D. Bikard, S. Cocco, R. Monasson, and J. Tubiana, Machine learning for evolutionary-based and physics-inspired protein design: Current and future synergies, *Curr. Opin. Struct. Biol.* **80**, 102571 (2023).
 - [5] J. Fernandez-de Cossio-Diaz, P. Hardouin, F.-X. Lyonnet du Moutier, A. Di Gioacchino, B. Marchand, Y. Ponty, B. Sargueil, R. Monasson, and S. Cocco, Designing molecular RNA switches with restricted Boltzmann machines, *Nat. Commun.* **16**, 11223 (2025).
 - [6] A. Decelle, B. Seoane, and L. Rosset, Unsupervised hierarchical clustering using the learning dynamics of restricted Boltzmann machines, *Phys. Rev. E* **108**, 014110 (2023).
 - [7] A. Braghetto, E. Orlandini, and M. Baiesi, Interpretable machine learning of amino acid patterns in proteins: A statistical ensemble approach, *J. Chem. Theory Comput.* **19**, 6011 (2023).
 - [8] C. Roussel, S. Cocco, and R. Monasson, Barriers and dynamical paths in alternating Gibbs sampling of restricted Boltzmann machines, *Phys. Rev. E* **104**, 034109 (2021).
 - [9] E. Agoritsas, G. Catania, A. Decelle, and B. Seoane, Explaining the effects of non-convergent MCMC in the training of energy-based models, in *International Conference on Machine Learning, Honolulu, HI* (PMLR, 2023), pp. 322–336.
 - [10] N. Béreux, A. Decelle, C. Furtlehner, L. Rosset, and B. Seoane, Fast training and sampling of restricted Boltzmann machines, in *Proceedings of the 13th International Conference on Learning Representations-ICLR 2025* (ICLR, Singapore, 2025).
 - [11] M. E. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics* (Clarendon Press, Oxford, 1999).
 - [12] G. E. Hinton, P. Dayan, B. J. Frey, and R. M. Neal, The “wake-sleep” algorithm for unsupervised neural networks, *Science* **268**, 1158 (1995).
 - [13] R. Salakhutdinov and G. Hinton, Deep Boltzmann machines, in *Artificial Intelligence and Statistics, Clearwater Beach, FL* (PMLR, 2009), pp. 448–455.
 - [14] J. Bornschein and Y. Bengio, Reweighted wake-sleep, in *Proceedings of the International Conference on Learning Representations (ICLR, San Diego, CA, 2015)*.
 - [15] D. P. Kingma and M. Welling, Auto-encoding variational Bayes, [arXiv:1312.6114](https://arxiv.org/abs/1312.6114).
 - [16] D. J. Rezende, S. Mohamed, and D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, in *International Conference on Machine Learning, Beijing, China* (PMLR, 2014), pp. 1278–1286.
 - [17] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in *International Conference on Machine Learning, Lille, France* (PMLR, 2015), pp. 2256–2265.
 - [18] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, Score-based generative modeling through stochastic differential equations, [arXiv:2011.13456](https://arxiv.org/abs/2011.13456).
 - [19] G. Biroli, T. Bonnaire, V. De Bortoli, and M. Mézard, Dynamical regimes of diffusion models, *Nat. Commun.* **15**, 9957 (2024).
 - [20] M. Kamb and S. Ganguli, An analytic theory of creativity in convolutional diffusion models, [arXiv:2412.20292](https://arxiv.org/abs/2412.20292).
 - [21] A. Sclocchi, A. Favero, and M. Wyart, A phase transition in diffusion models reveals the hierarchical nature of data, *Proc. Natl. Acad. Sci. USA* **122**, e2408799121 (2025).
 - [22] K. Ikeda, T. Uda, D. Okanohara, and S. Ito, Speed-accuracy relations for diffusion models: Wisdom from nonequilibrium thermodynamics and optimal transport, *Phys. Rev. X* **15**, 031031 (2025).
 - [23] M. Baiesi and C. Maes, Life efficiency does not always increase with the dissipation rate, *J. Phys. Commun.* **2**, 045017 (2018).
 - [24] A. Murugan, D. A. Huse, and S. Leibler, Speed, dissipation, and error in kinetic proofreading, *Proc. Natl. Acad. Sci. USA* **109**, 12034 (2012).
 - [25] P. Gaspard, Kinetics and thermodynamics of DNA polymerases with exonuclease proofreading, *Phys. Rev. E* **93**, 042420 (2016).
 - [26] F. Jülicher, A. Ajdari, and J. Prost, Modeling molecular motors, *Rev. Mod. Phys.* **69**, 1269 (1997).

- [27] A. Parmeggiani, T. Franosch, and E. Frey, Phase coexistence in driven one-dimensional transport, *Phys. Rev. Lett.* **90**, 086601 (2003).
- [28] T. Chou, K. Mallick, and R. K. Zia, Non-equilibrium statistical mechanics: From a paradigmatic model to biological transport, *Rep. Prog. Phys.* **74**, 116601 (2011).
- [29] P. Sartori and S. Pigolotti, Thermodynamics of error correction, *Phys. Rev. X* **5**, 041039 (2015).
- [30] A. C. Barato, D. Hartich, and U. Seifert, Efficiency of cellular information processing, *New J. Phys.* **16**, 103024 (2014).
- [31] P. Diaconis, S. Holmes, and R. M. Neal, Analysis of a non-reversible Markov chain sampler, *Ann. Appl. Probab.* **10**, 726 (2000).
- [32] E. P. Bernard, W. Krauth, and D. B. Wilson, Event-chain Monte Carlo algorithms for hard-sphere systems, *Phys. Rev. E* **80**, 056704 (2009).
- [33] M. Michel, S. C. Kapfer, and W. Krauth, Generalized event-chain Monte Carlo: Constructing rejection-free global-balance algorithms from infinitesimal steps, *J. Chem. Phys.* **140**, 054116 (2014).
- [34] M. Vucelja, Lifting—a nonreversible Markov chain Monte Carlo algorithm, *Am. J. Phys.* **84**, 958 (2016).
- [35] G. Montavon and K.-R. Müller, Deep Boltzmann machines and the centering trick, *Neural Networks: Tricks of the Trade* (Springer, Berlin, 2012), pp. 621–637.
- [36] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE* **86**, 2278 (1998).
- [37] A. Barra, G. Genovese, P. Sollich, and D. Tantari, Phase transitions in restricted Boltzmann machines with generic priors, *Phys. Rev. E* **96**, 042156 (2017).
- [38] J. Tubiana and R. Monasson, Emergence of compositional representations in restricted Boltzmann machines, *Phys. Rev. Lett.* **118**, 138301 (2017).
- [39] E. Ventura, S. Cocco, R. Monasson, and F. Zamponi, Unlearning regularization for Boltzmann machines, *Mach. Learn.: Sci. Technol.* **5**, 025078 (2024).
- [40] D. S. Salazar, Nonequilibrium thermodynamics of restricted Boltzmann machines, *Phys. Rev. E* **96**, 022131 (2017).
- [41] A. Decelle, C. Furtlehner, and B. Seoane, Equilibrium and non-equilibrium regimes in the learning of restricted Boltzmann machines, *J. Stat. Mech.* (2022) 114009.
- [42] D. Bachtis, G. Biroli, A. Decelle, and B. Seoane, Cascade of phase transitions in the training of energy-based models, *Adv. Neural Infor. Proc. Systems* **37**, 55591 (2024).