GENERATIVE ENERGY-BASED PREDICTIVE CODING NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Predictive-coding networks (PCNs) perform inference by relaxing hidden activities to minimise layer-wise prediction errors. Although this process is commonly described in terms of energy minimisation, the resulting state objective does not by itself define a probability distribution over sensory inputs and therefore cannot directly support input-space generation. In this work, we propose the Generative Energy-Based Predictive Coding Network (GEB-PCN), which turns the state energy of a forward PCN into a trainable input-space energy model. We define a joint energy over inputs and hidden activities by combining layer-wise prediction errors with an output energy, and obtain the energy of an input by evaluating this objective at its inferred hidden state. Contrastive learning uses finite-step inferred states to distinguish data from model-generated samples, without requiring an explicit top-down generative hierarchy. To generate samples, we develop a coupled PC–Langevin procedure that jointly refines hidden activities and sensory inputs. Under the original benchmark configurations, GEB-PCN attains the lowest domain FID among all compared generators on MNIST and Fashion-MNIST, well below the best-performing baselines. FiLM conditioning of the joint energy further demonstrates class-conditional generation for generative PCNs.

1 INTRODUCTION

Predictive coding (PC) (Rao & Ballard, 1999; Millidge et al., 2022) represents inputs and hidden activities through prediction errors. Generative PC predicts sensory variables from higher-level causes, whereas discriminative PC predicts hidden activities in the forward direction. Probabilistic treatments derive these errors from an explicit hierarchical model and a variational approximation (Friston et al., 2007; Marino, 2022), whereas machine-learning treatments ask when activity relaxation approximates error backpropagation (Whittington & Bogacz, 2017). These interpretations share an optimisation motif, but they do not make every squared-error PC objective a likelihood. Recent methods use a generative hierarchy for sampling (Oliviers et al., 2024) or combine both directions for inference (Oliviers et al., 2026). We study a different question: can the state energy of a *forward* PC network regularise an input-space energy-based model (EBM) and support generation without an explicit decoder hierarchy?

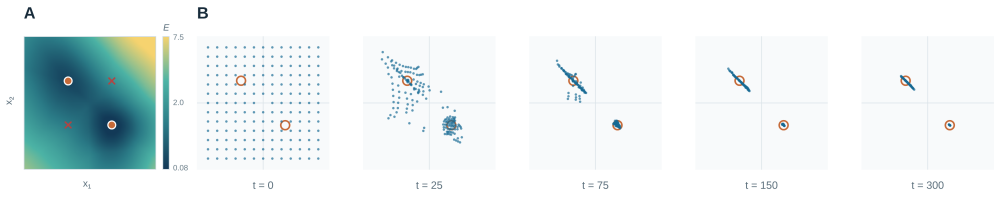

Figure 1: PC energy and input-space generation on class-clamped XOR. **A**: Equilibrium energy for the positive XOR class; orange circles mark valid modes and red crosses mark invalid configurations. **B**: Deterministic joint input–state descent from a regular grid at successive steps t, converging toward the two valid modes. The colour bar reports energy E, and all panels use the same input coordinates.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

EBMs (Du & Mordatch, 2019; Nijkamp et al., 2019) provide a direct construction: contrastive training assigns low energy to data and high energy to negative samples, and input gradients guide stochastic generation. Our GEB-PCN combines an output energy with forward prediction-error penalties, evaluates that joint energy after hidden-state inference, and couples hidden-state updates with input-space Langevin updates. Gradients are stopped through the inference trajectory and truncated at PC value nodes (vodes); the resulting update avoids unrolling state inference but is not claimed to be fully synapse-local. Figure 1 gives a two-dimensional view of the resulting energy and generation dynamics.

Our contributions are:

- We introduce GEB-PCN, a generative energy-based framework that incorporates the prediction-error objective of a forward PCN into an input-space energy, enabling generation without an explicit top-down decoder hierarchy.
- The framework combines contrastive learning at inferred hidden states with a coupled PC–Langevin sampler, coupling hidden-state inference and input-space generation through a shared energy function.
- Experiments on MNIST, Fashion-MNIST, and CIFAR-10 assess the framework's generation and inference capabilities against existing generative PC models, including fidelity and coverage trade-offs.
- Sampling analyses identify how PC force, replay initialisation, and coupled updates affect finite-horizon generation and sampling cost.

2 Background and Notation

In this section, we establish the predictive coding (PC) notation for a forward hierarchy and review the energy-based model (EBM) formulation of Du & Mordatch (2019), which motivates our approach.

Predictive coding (PC). A forward hierarchy of L layers associates each selected layer ℓ with an inferred activity $\boldsymbol{x}^{(\ell)}$ and weights $\boldsymbol{w}^{(\ell-1)}$ feeding that layer. Its bottom-up prediction and associated error are

$$\boldsymbol{v}^{(\ell)} \;=\; \boldsymbol{w}^{(\ell-1)}\, f\left(\boldsymbol{x}^{(\ell-1)}\right), \qquad \boldsymbol{\varepsilon}^{(\ell)} \;=\; \boldsymbol{x}^{(\ell)} - \boldsymbol{v}^{(\ell)}, \tag{1}$$

The associated state objective is the sum of squared prediction errors,

$$\mathcal{F}\Big(\boldsymbol{x}^{(1:L)}, \boldsymbol{\theta}\Big) \;= \sum_{\ell \in \mathcal{P}} \tfrac{1}{2}\, {\boldsymbol{\varepsilon}^{(\ell)}}^{\top} \boldsymbol{\varepsilon}^{(\ell)}, \tag{2}$$

where $\boldsymbol{\theta}$ collects the weights and $\mathcal{P}$ indexes layers carrying a *value node* (vode). With unit-covariance Gaussian conditionals, an individual squared error is a negative conditional log-density up to a constant (Marino, 2022). Interpreting a sum of such terms as variational free energy additionally requires a complete generative model, priors, and an explicit variational approximation (Friston et al., 2007). Equation equation 2 is therefore used only as a state objective in our forward hierarchy. State inference updates these activities with the weights frozen,

$$\boldsymbol{x}^{(\ell)} \;\leftarrow\; \boldsymbol{x}^{(\ell)} - \gamma\, \nabla_{\boldsymbol{x}^{(\ell)}} \mathcal{F}, \qquad \text{for } \mathcal{T} \text{ steps}, \tag{3}$$

at rate γ. This forward construction differs from canonical top-down generative PC (Rao & Ballard, 1999; Oliviers et al., 2024); bidirectional PC combines the two directions (Oliviers et al., 2026). Importantly, the forward conditionals suggested by Equation (1) do not themselves specify a normalised density over $\boldsymbol{x}^{(0)}$. The input density in our method is instead defined by contrastive EBM training.

Energy-based models (EBM). An EBM (Du & Mordatch, 2019) prescribes a density through a scalar energy $E_{\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathbb{R}$ via the Gibbs distribution

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{x}) \;&=\; \frac{\exp(-E_{\boldsymbol{\theta}}(\boldsymbol{x}))}{Z(\boldsymbol{\theta})}, \\ Z(\boldsymbol{\theta}) \;&=\; \int \exp(-E_{\boldsymbol{\theta}}(\boldsymbol{x}'))\; d\boldsymbol{x}', \end{aligned} \tag{4}$$

whose partition function Z is intractable. Maximum-likelihood training contrasts data against model samples,

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{ML}} &= \mathbb{E}_{\boldsymbol{x}^+ \sim p_D}\left(\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\boldsymbol{x}^+)\right) \\ &\quad - \mathbb{E}_{\boldsymbol{x}^- \sim p_{\boldsymbol{\theta}}}\left(\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\boldsymbol{x}^-)\right),\end{aligned} \tag{5}$$

and negatives can be approximated by Langevin dynamics,

$$\tilde{\boldsymbol{x}}^k = \tilde{\boldsymbol{x}}^{k-1} - \tfrac{\delta}{2} \nabla_{\boldsymbol{x}} E_{\boldsymbol{\theta}}\left(\tilde{\boldsymbol{x}}^{k-1}\right) + \sigma\, \boldsymbol{\omega}^k, \quad \boldsymbol{\omega}^k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \tag{6}$$

where a suitably matched noise scale and sufficiently small step size recover the usual unadjusted Langevin approximation. Finite-step clipping and bounded pixels, as used below, define a practical sampling kernel rather than an exact draw from Equation (4). Replacing the model expectation by a short chain is a contrastive-divergence approximation (Hinton, 2002); persistent contrastive divergence instead carries chain state across parameter updates (Tieleman, 2008). Because a finite chain depends on the sampler, the resulting update need not equal the likelihood gradient; in particular, the gradient induced by the negative-sample distribution can be numerically important when it is not treated as fixed (Du et al., 2021).

3 Method

Building on the PC and EBM formulations above, we introduce the Generative Energy-Based Predictive Coding Network (GEB-PCN). The model uses a joint energy for hidden-state inference and input-space generation, so an input can be evaluated at its inferred activities and refined as those activities evolve. We describe the energy construction, its contrastive training, and the coupled PC–Langevin updates used for finite-step generation.

3.1 The GEB-PCN Energy

GEB-PCN connects predictive-coding inference with energy-based generation by assigning each input an energy through hidden-state relaxation. In this formulation, forward prediction errors couple the input to inferred neural activities, while an output potential constrains the resulting representation. Together, these components define a shared energy for hidden-state inference and input-space generation.

For an input $x \equiv \boldsymbol{x}^{(0)}$, let h collect the inferred hidden activities at layers indexed by $\mathcal{P} \subseteq \{1, \ldots, L\}$. At each selected layer, the prediction error $\boldsymbol{\varepsilon}^{(\ell)}$ measures the discrepancy between the activity $\boldsymbol{x}^{(\ell)}$ and its feedforward prediction. The joint energy is

$$G_{\boldsymbol{\theta}}(x, h) = \lambda_{\mathrm{pc}} \sum_{\ell \in \mathcal{P}} \tfrac{1}{2}\, \boldsymbol{\varepsilon}^{(\ell)\top} \boldsymbol{\varepsilon}^{(\ell)} + E_{\mathrm{out},\boldsymbol{\theta}}(x, h), \tag{7}$$

where λ_{pc} controls the relative contribution of predictive consistency and the output potential. The output term can take the form of a quadratic energy, $E_{\mathrm{out},\boldsymbol{\theta}}(x, h) = \|f_{\boldsymbol{\theta}}(x, h)\|_2^2 / d$, for a d-dimensional output, or a directly learned scalar energy. For the quadratic form, clamping an output activity to zero gives $\varepsilon_{\mathrm{out}} = 0 - f_{\boldsymbol{\theta}}(x, h)$ and $E_{\mathrm{out}} = \|\varepsilon_{\mathrm{out}}\|^2 / d$. Keeping this target fixed prevents the output term from vanishing merely because a freely inferred output activity matches its prediction. The quadratic output energy therefore requires no output-state inference and remains defined even when internal PC states are disabled.

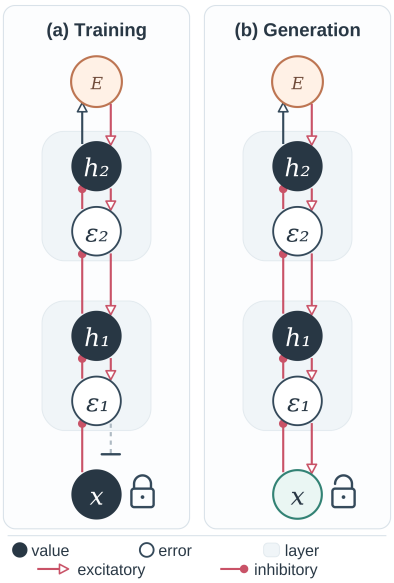

Figure 2: Neural implementation of GEB-PCN.

From state inference to input energy. With the input held fixed, hidden activities relax under the *total* joint energy. This allows the output potential to influence the inferred representation, while the layer-wise errors maintain

its consistency with the input. Writing $h^\star(x)$ for a local equilibrium reached by this relaxation, the corresponding input energy is

$$E_{\boldsymbol{\theta}}(x) = G_{\boldsymbol{\theta}}(x, h^\star(x)) \,. \tag{8}$$

This construction defines an input-space energy, rather than a latent-variable likelihood identity implied by the forward PC conditionals. The profile in Equation (8) is an equilibrium ideal: for convergent iterative inference, it is obtained in the limit $\mathcal{T} \to \infty$, consistent with PC analyses that let hidden activities settle before learning (Whittington & Bogacz, 2017). Reaching this state can require many iterations and depends on hyperparameters, whereas a fixed inference budget limits computation (Salvatori et al., 2024). For tractability, we use a fixed finite $\mathcal{T}$ in training and carry hidden states alongside changing inputs during generation, without assuming proximity to $h^\star(x)$. Consequently, contrastive training uses finite-state joint energies, while each generation step uses $\partial_x G_{\boldsymbol{\theta}}(x_t, h_t)$ at the current hidden state.

Neural organisation and computation. The architecture in Figure 2 implements the model as an explicit energy landscape over value neurons and error neurons. Value neurons carry the inferred activities, while error neurons compute local prediction errors that contribute to the joint energy $G_\theta(x, h)$ through weighted error terms and an output potential. During inference, the sensory input x is clamped, and the hidden value neurons relax to minimise this joint energy through local, bidirectional interactions. During generation, the sensory input becomes free, allowing the network to jointly explore the energy landscape over both sensory and hidden states. This organisation provides a biologically grounded implementation of predictive coding, where computation emerges from local energy minimisation rather than explicit top-down generative hierarchies.

3.2 Contrastive Training at Inferred States

We train $\boldsymbol{\theta}$ with the contrastive loss of Du & Mordatch (2019). For a data point $\boldsymbol{x}^{(0)+}$ and a negative $\boldsymbol{x}^{(0)-}$, let $h_{\mathcal{T}}^{\pm}$ be the finite-step inferred states and $E^{\pm} = G_{\boldsymbol{\theta}}(\boldsymbol{x}^{(0)\pm}, h_{\mathcal{T}}^{\pm})$.

$$\begin{aligned} \mathcal{L} \;&=\; \left(E^{+} - E^{-}\right) \\ &\quad+\; \beta \left(\left(E^{+} - c\right)^2 + \left(E^{-} - c\right)^2 \right), \end{aligned} \tag{9}$$

Here $\beta \geq 0$ controls optional energy regularisation. The contrast pushes E^+ down and E^- up; for $\beta > 0$, the penalty anchors both near c. As independent, unconstrained scalars with $\beta > 0$, their targets are $E^+ = c - 1/(2\beta)$ and $E^- = c + 1/(2\beta)$. The shared network and a non-negative quadratic head can restrict attainability of these targets.

Each gradient step has two phases. **(i) State inference**: with $\boldsymbol{\theta}$ frozen, positive and negative vode states are updated using the gradient of $G_{\boldsymbol{\theta}}$, yielding finite-step inferred states. **(ii) Parameter update**: one Adam step updates $\boldsymbol{\theta}$ using $\nabla_{\boldsymbol{\theta}} \mathcal{L}$ evaluated at those states. Gradients through the inference iterations are stopped. This is an envelope-style update at the inferred states, subject to the Vode truncation described above. Fixed-point and optimisation layers can also avoid unrolling by applying implicit differentiation to their equilibrium conditions (Bai et al., 2019; Blondel et al., 2022). Our stopped-state update uses partial derivatives at the inferred states. It equals the derivative of the profiled energy only at stationarity, which the finite-reference probes below do not certify.

Algorithm 1 GEB-PCN training (one parameter update)

Require: data $\boldsymbol{x}^{(0)+}$; replay buffer $\mathcal{B}$; budgets $K, \mathcal{T}$; rates γ, α.

1. Draw $\boldsymbol{x}^{(0)-}$ from the replay/noise mixture with ratio r; initialise h^-.
2. Refine $(\boldsymbol{x}^{(0)-}, h^-)$ for K joint steps using Equation (10), with $\lambda_s = 0$.
3. Fix $\boldsymbol{x}^{(0)+}, \boldsymbol{x}^{(0)-}$; reinitialise $h^{\pm}$ and take $\mathcal{T}$ updates $h^{\pm} \leftarrow h^{\pm} - \gamma\, \partial_h G_{\boldsymbol{\theta}}(\boldsymbol{x}^{(0)\pm}, h^{\pm})$.
4. Store $\boldsymbol{x}^{(0)-}$ in $\mathcal{B}$.
5. Update $\boldsymbol{\theta}$ by Adam at rate α using Equation (9), treating sampled inputs and inferred states as fixed.

3.3 Joint PC–Langevin Sampling

To enable scalable inference without solving for hidden-state equilibrium at every iteration, we couple predictive coding (PC) state inference (Whittington & Bogacz, 2017) with Langevin energy-gradient

sampling (Du & Mordatch, 2019). Under fixed parameters $\boldsymbol{\theta}$, the input image x and internal state h are updated jointly in a unified dynamical system:

$$\begin{aligned} h_{t+1} &= h_t - \gamma \nabla_h G_{\boldsymbol{\theta}}(x_t, h_t), \\ x_{t+1} &= x_t - \delta \nabla_x G_{\boldsymbol{\theta}}(x_t, h_t) + \sigma \boldsymbol{\omega}_t, \qquad \boldsymbol{\omega}_t \sim \mathcal{N}(0, I), \end{aligned} \tag{10}$$

where γ and δ denote the step sizes for hidden-state inference and image Langevin dynamics, respectively. Hidden states evolve deterministically alongside the stochastic pixel dynamics, tracking non-stationary inputs in real time. The exact implementation, including numerical bounds, is specified in Appendix B.

Replay Memory and Initialisation. To avoid the prohibitive cost of sampling negative particles from scratch at each training step, we incorporate persistent contrastive divergence via a replay buffer (Tieleman, 2008). Training negative chains are initialized by mixing stored buffer samples with uniform noise at ratio r, allowing MCMC dynamics to persist across parameter updates. Setting $r = 0$ naturally recovers short-run non-persistent sampling (Nijkamp et al., 2019). During evaluation, generation initializes from either the persistent image buffer or uniform noise depending on the sampling protocol, while internal states are initialized once per chain and refined alongside image updates.

Mechanistic Influence of Output Energy on Pixels. Under gradient truncation, the output potential $E_{\text{out},\boldsymbol{\theta}}(h)$ does not backpropagate directly into the pixel space through the independent state h. Instead, it alters the trajectory of h by introducing an explicit gradient force $\nabla_h E_{\text{out},\boldsymbol{\theta}}(h)$. This shift modifies the predictive error term $\varepsilon_h = h - u_{\boldsymbol{\theta}}(x)$, which subsequently governs pixel updates via the feedforward transposed Jacobian ($\nabla_x G \propto J_u(x)^\top \varepsilon_h$). Joint state–image dynamics thus allow output potential constraints to indirectly steer pixel synthesis across iterative sampling steps, bypassing the need for state equilibrium prior to each pixel update.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate our method on standard image benchmarks: MNIST (LeCun et al., 1998) and Fashion-MNIST (Xiao et al., 2017) (28×28 grayscale), as well as CIFAR-10 (Krizhevsky, 2009) (32×32 RGB). For the finalized MNIST evaluation, the QMNIST `test50k` set (Yadav & Bottou, 2019) is employed as an external reference partition.

Baselines and comparison settings. We benchmark GEB-PCN against an EBM (Du & Mordatch, 2019), MCPC (Oliviers et al., 2024), bPC (Oliviers et al., 2026), genPC (Rao & Ballard, 1999), hybridPC (Tscshantz et al., 2023), a VAE (Kingma & Welling, 2014), and DCPC (Sennesh et al., 2024). The following architectural and sampling differences inform these comparisons:

- **Architectures and training:** The EBM has a scalar head without PC states (398k-parameter MLP on grayscale; IGEBM residual network on CIFAR-10). MNIST GEB-PCN, MCPC and bPC have $431/407/550$k parameters; Fashion GEB-PCN uses a hybrid convolutional-MLP. Grayscale checkpoints use epoch 100 except genPC/hybridPC (released 25-epoch recipe (Oliviers et al., 2026)) and pre-trained DCPC ($2{,}000$ epochs).
- **Sampling protocols:** GEB-PCN and EBM start from replay; MCPC returns decoder means after stochastic activity updates; bPC, genPC and hybridPC use ancestral means; VAE and DCPC use prior-sampled decoder means.

4.2 Image generation and evaluation sensitivity

Generation endpoints across benchmarks. Table 1 reports FID and IS for the evaluated generators. On MNIST, GEB-PCN has the lowest FID (0.87 ± 0.02), followed by the EBM (1.20 ± 0.16); VAE and MCPC are near 3.0, bPC, genPC and hybridPC range from 7.55 to 8.70, and DCPC has 47.12. Its IS (8.75 ± 0.02) is close to the table-high EBM value (8.79 ± 0.08). On Fashion-MNIST, GEB-PCN has both the lowest FID (3.33 ± 0.89) and highest IS (7.33 ± 0.13), ahead of the VAE's 6.44 ± 0.07

Table 1: Generation endpoints. Grayscale results use domain ResNet-18, 10,000 images and five seeds (mean ± sample SD), with QMNIST/Fashion test references. CIFAR uses Inception-v3-2048, 50,000 generated samples, all 50,000 CIFAR-10 training images as FID references, and one training seed. Bold marks the best value in each column.

	MNIST		Fashion-MNIST		CIFAR-10	
Model	FID ↓	IS ↑	FID ↓	IS ↑	FID ↓	IS ↑
GEB-PCN (ours)	$\mathbf{0.87 \pm 0.02}$	8.75 ± 0.02	$\mathbf{3.33 \pm 0.89}$	$\mathbf{7.33 \pm 0.13}$	**62.93**	**5.15**
EBM	1.20 ± 0.16	$\mathbf{8.79 \pm 0.08}$	14.61 ± 1.08	6.01 ± 0.08	78.75	3.79
MCPC (Oliviers et al., 2024)	3.04 ± 0.29	7.28 ± 0.09	8.72 ± 0.73	6.19 ± 0.07	112.03	4.11
bPC (Oliviers et al., 2026)	8.70 ± 0.63	5.84 ± 0.16	7.40 ± 0.43	6.03 ± 0.06	190.70	2.49
genPC (Rao & Ballard, 1999)	8.32 ± 0.46	5.84 ± 0.09	10.72 ± 0.87	5.62 ± 0.07	200.05	2.26
hybridPC (Tscshantz et al., 2023)	7.55 ± 0.34	5.96 ± 0.05	8.28 ± 0.53	5.83 ± 0.08	194.27	2.47
VAE (Kingma & Welling, 2014)	3.00 ± 0.26	7.65 ± 0.10	7.92 ± 0.51	6.44 ± 0.07	175.67	2.65
DCPC (Sennesh et al., 2024)†	47.12 ± 0.16	1.76 ± 0.01	63.35 ± 0.16	2.22 ± 0.01	122.89	3.91

Configurations, evaluation protocols and CIFAR-10 selection are detailed in Appendices C and D.6.

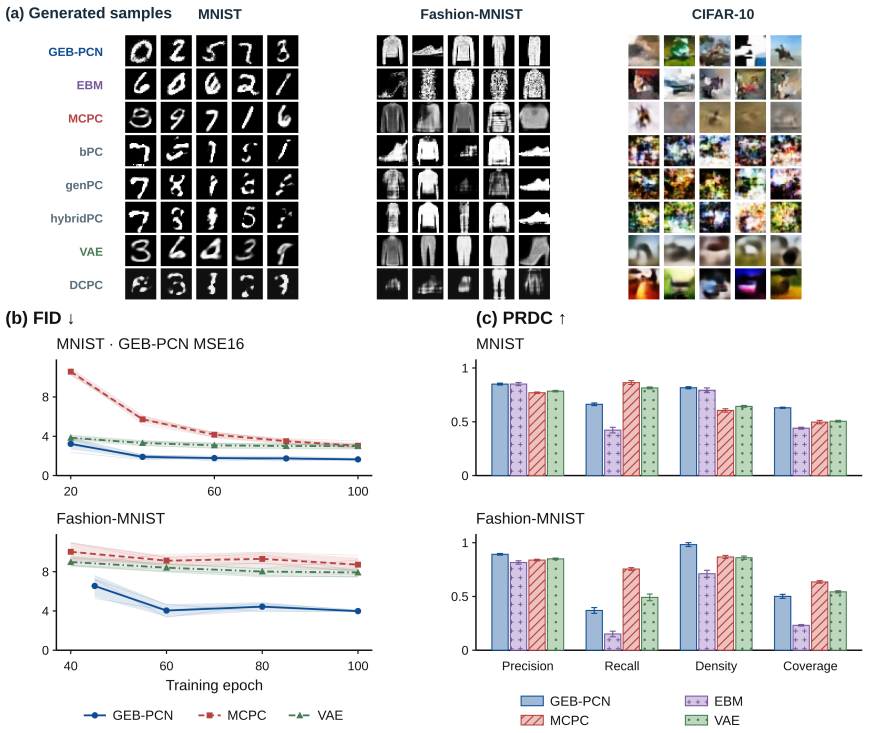

Figure 3: **Samples, training trajectories and coverage.** **(a)** Five samples per main-table model; bPC, genPC and hybridPC use ancestral decoder means, and MNIST GEB-PCN is selected for digit diversity. **(b)** Domain FID trajectories on 10,000 images. **(c)** PRDC for GEB-PCN, MCPC, EBM and VAE on fixed 5,000-image subsets. Lines and bars show five-seed means and sample SD.

IS. The next-lowest FIDs are bPC's 7.40 ± 0.43 and VAE's 7.92 ± 0.51; MCPC, hybridPC, genPC and EBM range from 8.28 to 14.61, and DCPC is at 63.35. Under the CIFAR-10 50,000-sample protocol, GEB-PCN records the lowest FID (62.93) and highest IS (5.15). The EBM is next in FID (78.75), while MCPC has the next-highest IS (4.11); MCPC, DCPC, VAE, bPC, hybridPC and genPC span FID 112.03–200.05. The MNIST and Fashion-MNIST results aggregate five seeds, whereas CIFAR-10 reports one seed; architectures and sampling budgets differ across methods, and reference populations differ across datasets.

Training Dynamics. The progression of generative performance over training is illustrated in Figure 3b. On MNIST, all three evaluated models exhibit a monotonic decrease in mean FID across

all monitored checkpoints. On Fashion-MNIST, GEB-PCN experiences its primary FID improvement between epochs 45 and 60, subsequently stabilizing with minor fluctuations around an FID of 4.0 across an independent three-seed evaluation cohort.

Fidelity versus Coverage Trade-offs (PRDC). We compare GEB-PCN with three complementary generators in Figure 3c. MCPC represents generative PC: among the PC comparators in Table 1, it leads MNIST FID and IS and Fashion-MNIST IS, although bPC has lower Fashion-MNIST FID. The EBM provides an input-space energy model without internal PC states, and the VAE represents latent encoder–decoder generation. Precision, Recall, Density and Coverage (PRDC) on the same fixed-size evaluation subsets then expose quality–coverage differences that FID alone cannot show:

- **MNIST:** GEB-PCN's Precision (0.850) is nearly identical to the EBM's (0.851), while its Density (0.816) and Coverage (0.630) exceed the EBM's ($0.793/0.441$), MCPC's ($0.606/0.497$) and VAE's ($0.643/0.505$). Its Recall (0.663) is below MCPC's 0.864 and VAE's 0.815, but above the EBM's 0.422. KID on the same feature space supports the fidelity comparison.
- **Fashion-MNIST:** GEB-PCN has the highest Precision (0.892) and Density (0.983) of the four models. Its Recall (0.370) and Coverage (0.501) are below MCPC's ($0.755/0.635$) and VAE's ($0.492/0.543$), but above the EBM's ($0.151/0.232$).

Taken together, GEB-PCN exceeds the EBM on seven of eight PRDC comparisons across the two datasets; the remaining MNIST Precision values are effectively tied (0.850 vs. 0.851). Relative to MCPC and VAE, it achieves higher Precision and Density at the cost of Recall, placing it toward the high-fidelity end of the quality–coverage trade-off.

Scaling to RGB Images: CIFAR-10. Moving to CIFAR-10 makes finite-step image-space sampling more demanding. The 18.59M-parameter convolutional GEB-PCN uses replay throughout its long-run training (Tieleman, 2008). After the monitored FID began to rise, we returned to the preceding epoch-80 checkpoint and continued training with an identity-preserving particle buffer. This late-stage strengthening of replay carries partially refined images across parameter updates, aiming to supply evolved negatives within a finite sampling budget. During the continuation, a fraction of particles is refreshed with perturbed training images, while hidden PC states are reinitialised.

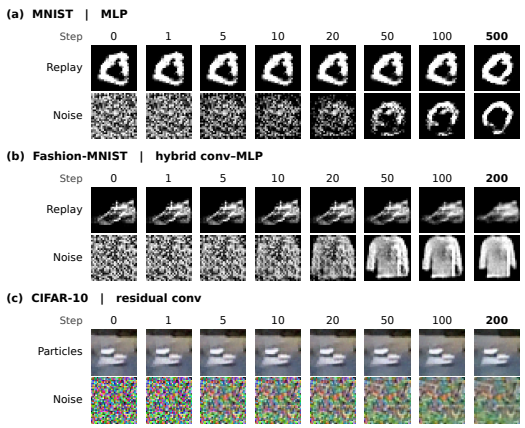

Figure 4: **Generation trajectories.** Replay/particle and noise starts over joint-update steps.

Across this continuation, particle-start FID improves from 89.12 at epoch 81 to 72.19 at epoch 85 under the same evaluation protocol. The reported GEB-PCN endpoint then achieves the best CIFAR-10 FID (62.93) and IS (5.15) in Table 1; the EBM obtains 78.75 and 3.79 under the same $50{,}000$-image evaluation. These results support late-stage particle replay as a practical way to improve GEB-PCN's observed RGB generation quality.

4.3 Class-conditional generation

To incorporate conditional constraints, we employ Feature-wise Linear Modulation (FiLM; Perez et al., 2018) to modulate state transitions $x \to h$ via a static, condition-specific embedding z_y (Figure 5**a**). Unlike stochastic latent EBMs (Nie et al., 2021), this design enforces an invariant conditioning signal during iterative (x, h) sampling.

The conditional FiLM GEB-PCN achieves performance highly comparable to a conditional VAE baseline (Figure 5**b**), attaining competitive recall (0.729 vs. 0.714) and label accuracy (0.939 vs. 0.964) alongside strong conditional fidelity (cFID $= 3.121$, global FID $= 0.743$).

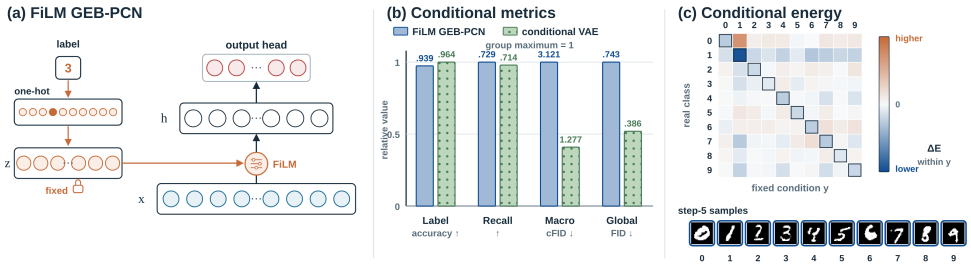

Figure 5: **FiLM conditioning and its energy signature. (a)** The clamped code z modulates the image-to-state path. **(b)** Step-5 FiLM versus the conditional VAE; each metric's maximum mean is one and labels give raw means. **(c)** Column-centred test energy across real classes and fixed conditions (blue is lower); below are first-archived step-5 samples for targets 0–9.

Landscape diagnostics reveal two key functional properties (Figure 5**c**): (i) **Intra-class separation:** at fixed $y = 1$, the PC term separates real digit-1 images from other digits (AUC 0.946), compared with 0.937 for total energy and 0.503 for the MSE output term; lower energy is scored as more compatible. (ii) **Conditional compatibility:** while on-diagonal energies are consistently minimized, low cross-condition argmin accuracy (0.242) indicates that modulated energies optimize fixed-condition sample compatibility rather than serving as a globally calibrated classifier.

5 Sampling Ablations and Analysis

The quality of a finite-step sample depends on more than the energy learned by the model. GEB-PCN updates the image while its hidden state is still moving, and a chain can begin either near the data distribution in replay memory or far from it in noise. A good replay FID could therefore reflect an advantageous starting population; a poor noise FID could reflect an insufficient sampling horizon. Likewise, a faster update rule is useful only if it preserves sample quality. To separate these effects, we ask whether the PC force actively guides noise-start images, when replay changes the outcome of a finite chain, and what computational benefit joint state–image updates provide. We compare sampling interventions on frozen models so that changes in the resulting trajectories can be attributed to the sampling procedure. The model choices, matched-start construction, search criteria and complete numerical results are in Appendix F.

5.1 The PC force in noise-start generation

The output potential acts through the hidden state, while the implemented image gradient in the single-state sampler contains the PC mismatch term $\lambda_{\mathrm{pc}} J_u(x)^\top [u_{\boldsymbol{\theta}}(x) - h]$. This raises a mechanistic question: does that coupling shape image synthesis when the starting image carries no replay information? We vary the PC coefficient only at sampling time and compare the same frozen models from noise and replay starts. In Figure 6**a**, a moderate increase reduces noise-start FID from 8.52 to 2.76, but raises replay FID from 0.671 to 1.520; larger increases impair both starts. The opposing responses show that the PC pathway participates in image updates and that a force useful for transport from noise need not be best for replay refinement. This sampling-time intervention does not establish the effect of training with or without internal PC.

5.2 Replay and the finite sampling horizon

Replay particles already lie near the training distribution, so short-chain quality alone cannot show how well the energy transports arbitrary noise. We follow continuous chains from both starts across sampling horizons and compare noise chains with an analytic hidden-state profile. The latter asks whether the replay–noise gap is simply caused by incomplete hidden inference at each image. At the original short horizons, replay has much lower FID than noise on both datasets (Figure 6**b**). Longer noise chains close that gap, whereas replay chains eventually drift; Fashion-MNIST noise quality also degrades after its best intermediate horizon. Exact hidden-state profiling does not remove the

noise disadvantage at the short horizons. Replay therefore supplies a useful *finite-step starting point*, and the carried-state sampler need not behave like its instantaneously profiled counterpart.

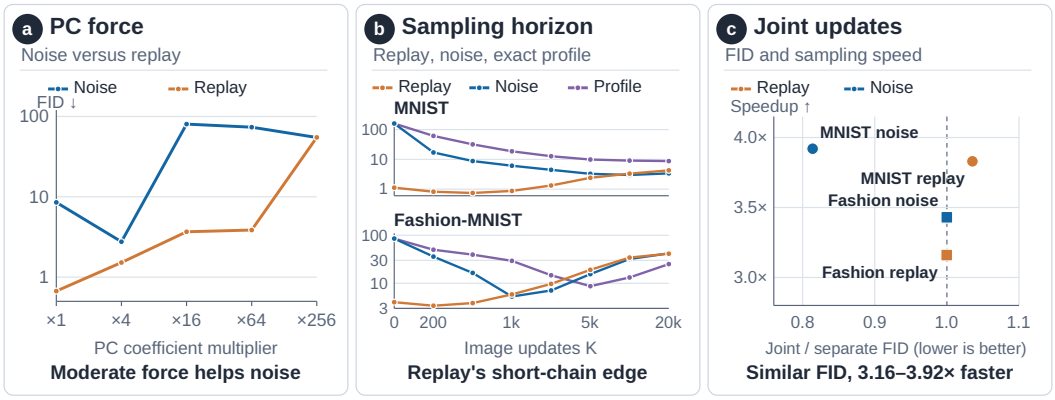

Figure 6: **Finite-step sampling analysis. (a)** MNIST FID under PC-force scaling from noise and replay. **(b)** FID trajectories from replay, noise and exact-state-profile noise on MNIST and Fashion-MNIST; the profile minimises h at each image step. **(c)** Joint/separate FID ratio versus paired speedup at matched image-update budgets; ratios below one favour joint sampling.

5.3 Sampling efficiency of joint updates

Joint state–image updates avoid repeatedly solving the hidden state before each image move. The relevant comparison is therefore whether this reduced inner computation retains the quality obtained by a separate update rule. Across replay and noise starts on both datasets, joint and state-preserving separate sampling produce similar FID at the same image-update budget: MNIST replay incurs a small FID increase under joint updates, MNIST noise improves, and Fashion-MNIST is nearly unchanged (Figure 6**c**). Joint sampling is 3.16–$3.92\times$ faster than the tested multi-step separate rule. This supports a sampling-cost benefit for coupled updates, subject to that specific comparator and fixed step budget. It does not measure training throughput or establish an equal-time quality advantage over all samplers.

6 Conclusion

GEB-PCN combines the state energy of a forward PCN with a learned output energy, then uses contrastive learning and coupled PC–Langevin updates to generate images without a top-down decoder. The framework extends to class-conditional generation through FiLM. In the evaluated configurations, GEB-PCN achieves the lowest FID among the reported baselines on MNIST, Fashion-MNIST and CIFAR-10, while PRDC reveals a quality–coverage trade-off relative to MCPC and VAE. Sampling interventions show that the PC force shapes noise-start trajectories, replay supplies strong finite-horizon starts, and joint updates attain similar FID to an eight-step separate sampler with lower sampling latency.

More broadly, GEB-PCN places forward predictive coding within a generative energy-based framework. Hidden-state relaxation provides a structured way to evaluate input energy, while contrastive learning and stochastic image updates make that energy usable for synthesis. This extends the role of PCNs from inferring representations of observed inputs to generating and refining sensory inputs, and offers a concrete setting for studying how finite-step inference, image-space sampling, and replay interact.

REPRODUCIBILITY STATEMENT

GEB-PCN and MCPC use JAX and the `pcx` predictive-coding library. Dataset splits and generation references are in Appendix A; state inference and joint sampling in Appendix B; architectures, training settings and compute in Appendix C; and evaluator features, checkpoint readouts, reference sets and statistical units in Appendix D. The main grayscale benchmarks use five training seeds and $10{,}000$ generated images per seed; CIFAR-10 main-table results use $50{,}000$ generated images per model. CIFAR-10 training, capacity controls and retrospective endpoint selection are documented in Appendices C.4, C.6 and D.6. Class-conditional training and selection are in Appendix E.

The sampling analyses in Appendix F use separate five-seed frozen cohorts. The PC-coefficient sweep compares five multipliers with $10{,}000$ images per seed and endpoint; the long-chain trajectories use $2{,}000$ per seed. Joint/separate FID comparisons use $10{,}000$ images per seed, and timing uses three interleaved batch-128 repeats per method and seed. The post-training head-removal control uses $2{,}000$ images per endpoint (Appendix D.2). The theory probes use a separate MNIST model, with their sample populations specified in Appendix G. Source code, frozen configurations, checkpoints, evaluator/reference hashes, generated arrays, per-seed metrics and figure/table scripts are retained in the experiment archive. The added summaries were independently recomputed from archived evaluation records.

AI USE STATEMENT

We used generative AI tools to identify related work; assist with experimental code implementation, analysis and figure-code development; draft selected passages; and revise and polish the manuscript. Codex was the primary tool across these tasks, while Kimi additionally assisted with experimental code. AI tools also supported consistency checks among the paper's claims, implementation and experimental records. The authors reviewed all AI-assisted work and take full responsibility for the final content, including its text, code, figures and results.

REFERENCES

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

André M. Bastos, William M. Usrey, Rick A. Adams, George R. Mangun, Pascal Fries, and Karl J. Friston. Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711, 2012.

Mikołaj Bińkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=r1lUOzWCW`.

Mathieu Blondel, Quentin Berthet, Marco Cuturi, Roy Frostig, Stephan Hoyer, Felipe Llinares-López, Fabian Pedregosa, and Jean-Philippe Vert. Efficient and modular implicit differentiation. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-0378.

Min Jin Chong and David Forsyth. Effectively unbiased FID and inception score and where to find them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6070–6079, 2020. doi: 10.1109/CVPR42600.2020.00611.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL `https://proceedings.neurips.cc/paper/2019/hash/378a063b8fdb1db941e34f4bde584c7d-Abstract.html`.

Yilun Du, Shuang Li, Joshua Tenenbaum, and Igor Mordatch. Improved contrastive divergence training of energy-based models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2837–2848, 2021. URL `https://proceedings.mlr.press/v139/du21b.html`.

Karl Friston, Jean Mattout, Nelson Trujillo-Barreto, John Ashburner, and Will Penny. Variational free energy and the Laplace approximation. *NeuroImage*, 34(1):220–234, 2007. doi: 10.1016/j.neuroimage.2006.08.035.

Will Grathwohl, Kuan-Chieh Wang, Jørn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hkxzx0NtDB.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002. doi: 10.1162/089976602760128018.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. *International Conference on Learning Representations (ICLR)*, 2014.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/0234c510bc6d908b28c70ff313743079-Abstract.html.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Joseph Marino. Predictive coding, variational autoencoders, and biological connections. *Neural Computation*, 34(1):1–44, 2022. doi: 10.1162/neco_a_01458.

Beren Millidge, Tommaso Salvatori, Yuhang Song, Rafał Bogacz, and Thomas Lukasiewicz. Predictive coding: Towards a future of deep learning beyond backpropagation? *arXiv preprint arXiv:2202.09467*, 2022.

Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7176–7185, 2020. URL https://proceedings.mlr.press/v119/naeem20a.html.

Weili Nie, Arash Vahdat, and Anima Anandkumar. Controllable and compositional generation with latent-space energy-based models. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/701d804549a4a23d3cae801dac6c2c75-Abstract.html.

Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run MCMC toward energy-based model. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/2bc8ae25856bc2a6a1333d1331a3b7a6-Abstract.html.

Gaspard Oliviers, Rafał Bogacz, and Alexander Meulemans. Learning probability distributions of sensory inputs with Monte Carlo predictive coding. *PLOS Computational Biology*, 20(10): e1012532, 2024. doi: 10.1371/journal.pcbi.1012532.

Gaspard Oliviers, Mufeng Tang, and Rafał Bogacz. Bidirectional predictive coding. In *International Conference on Learning Representations*, 2026.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 3942–3951, 2018. doi: 10.1609/aaai.v32i1.11671.

Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, 1999. doi: 10.1038/4580.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Tommaso Salvatori, Yuhang Song, Yordan Yordanov, Beren Millidge, Lei Sha, Cornelius Emde, Zhenghua Xu, Rafal Bogacz, and Thomas Lukasiewicz. A stable, fast, and fully automatic learning algorithm for predictive coding networks. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/554414e570a85eb3118e988c5d77986f-Abstract-Conference.html.

Eli Sennesh, Hao Wu, and Tommaso Salvatori. Divide-and-conquer predictive coding: a structured Bayesian inference algorithm. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Tijmen Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 1064–1071. ACM, 2008. doi: 10.1145/1390156.1390290.

Alexander Tscshantz, Beren Millidge, Anil K. Seth, and Christopher L. Buckley. Hybrid predictive coding: Inferring, fast and slow. *PLOS Computational Biology*, 19(8):e1011280, 2023. doi: 10.1371/journal.pcbi.1011280.

James C. R. Whittington and Rafał Bogacz. An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural Computation*, 29(5): 1229–1262, 2017. doi: 10.1162/NECO_a_00949.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Chhavi Yadav and Léon Bottou. Cold case: The lost MNIST digits. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Umais Zahid, Qinghai Guo, and Zafeirios Fountas. Sample as you infer: Predictive coding with Langevin dynamics. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 58105–58121, 2024. URL https://proceedings.mlr.press/v235/zahid24a.html.

The appendix groups datasets and splits (Appendix A), sampler implementation (Appendix B), architectures and training (Appendix C), evaluation and CIFAR-10 selection (Appendix D), conditional inference (Appendix E), sampling analyses (Appendix F), and theory diagnostics (Appendix G).

A Datasets

We evaluate on MNIST (LeCun et al., 1998), Fashion-MNIST (Xiao et al., 2017) and CIFAR-10 (Krizhevsky, 2009), using the $50{,}000$-image QMNIST `test50k` subset (Yadav & Bottou, 2019) as an external MNIST generation reference. Table 2 summarises dataset properties and partitions. Pixels are scaled to $[0, 1]$ for GEB-PCN.

Table 2: Summary of dataset characteristics and final-generator data partitions. Test counts refer to the full official test sets, except for QMNIST, which lists the external reference subset. A dash denotes no separate partition for that use.

Dataset / use	Image size	Channels	Classes	Train	Validation	Test
MNIST	28×28	1	10	60,000	—	10,000
Fashion-MNIST	28×28	1	10	60,000	—	10,000
CIFAR-10 / GEB-PCN, EBM	32×32	3	10	50,000	—	10,000
CIFAR-10 / other baselines	32×32	3	10	45,000	5,000	10,000
QMNIST / reference	28×28	1	10	—	—	50,000

The grayscale feature evaluators use stratified $55{,}000/5{,}000$ train/validation splits of the official training sets. The CIFAR-10 baselines other than GEB-PCN and EBM use a $45{,}000/5{,}000$ train/validation split of the official training set. The Fashion-MNIST MSE64 configuration was selected on a separate stratified $55{,}000/5{,}000$ split; final models were then refitted on all $60{,}000$ training images.

QMNIST `test50k` excludes the first $10{,}000$ examples of the QMNIST test archive; the official MNIST test set was used during model and sampler development. Fashion-MNIST test images serve as both the generation reference and the out-of-distribution inputs for MNIST evaluation. CIFAR-10 scoring subsets and shared-reference controls are specified in Appendix D.6.

B Predictive-coding inference and joint Langevin sampling

This section expands the two inner computations of Algorithm 1 with weights frozen: PC inference updates hidden activities with the image fixed; joint sampling updates the image and hidden activities together. A separate contrastive parameter update learns the energy afterwards.

B.1 Predictive-coding network: prediction, inference and learning

Predictions and independent activities. At each PC layer $\ell \in \mathcal{P}$ the preceding block predicts u_ℓ, but the next block receives an independent activity h_ℓ; the local mismatch is $\varepsilon_\ell = h_\ell - u_\ell$ (Equation (1)). Layers without a PC state compute feedforward. State inference minimises the total energy $G_{\boldsymbol{\theta}}$ (Equation (7)), so the output potential participates in inference even when its gradient cannot pass through a PC state to the image.

Fixed-image state inference. Clamp the image, freeze $\boldsymbol{\theta}$, initialise the active hidden arrays, and take $\mathcal{T}$ SGD steps on the free activities. In the single-state backbones used here, $u_{\boldsymbol{\theta}}(x)$ is the feedforward prefix and the per-image update is

$$h_{t+1} = h_t - \frac{\gamma}{d_B} \left\{ \lambda_{\mathrm{pc}} [h_t - u_{\boldsymbol{\theta}}(x)] + \partial_h E_{\mathrm{out},\boldsymbol{\theta}}(h_t) \right\}. \tag{11}$$

d_B is the batch reduction: $d_B = B$ in the main training recipes, $d_B = 1$ in the per-sample variant and in the joint sampler below, so identical configured rates need not give identical per-image update magnitudes.

The reported models use $\mathcal{T} = 32$, configured rate 0.01 and plain SGD without momentum or activity noise; random initialisation is distinct from per-step Langevin noise, and the finite budget does not certify convergence to $h^\star(x)$.

Algorithm 2 PC state inference for a clamped image batch

Require: images $x_{1:B}$; frozen weights $\boldsymbol{\theta}$; PC layers $\mathcal{P}$; budget $\mathcal{T}$; rate γ; divisor d_B.
1: Run an initialisation pass to obtain each active state's shape; draw $h_{0,i,\ell} \sim \mathcal{N}(0, I)$ for the reported models.
2: **for** $t = 0, \ldots, \mathcal{T} - 1$ **do**
3: Recompute feedforward predictions $u_{t,i,\ell}$ using the current independent states $h_{t,i,\ell}$.
4: Compute $\varepsilon_{t,i,\ell} = h_{t,i,\ell} - u_{t,i,\ell}$ and the total objective $\mathcal{G}_t = d_B^{-1} \sum_{i=1}^{B} G_{\boldsymbol{\theta}}(x_i, h_{t,i})$.
5: Compute $g_t^h = \partial_{h_t} \mathcal{G}_t$, with all images and weights fixed.
6: Update all free activities simultaneously: $h_{t+1} \leftarrow h_t - \gamma g_t^h$.
7: **end for**
8: Return $\mathrm{stopgrad}(h_{\mathcal{T}})$.

From inferred states to a parameter update. Generate a negative batch (Algorithm 3), infer positive and negative states separately (Algorithm 2), then take one Adam update on the regularised contrastive loss (Equation (9)) with images and inferred states held fixed. No gradient propagates through the inference iterations or the negative-image trajectory; the 32-step inference runs once per positive/negative batch, not inside each joint sampling step.

B.2 Joint PC–Langevin sampling

Initialisation and chain state. Replay and uniform-noise starts follow Du & Mordatch (2019): main-table grayscale evaluations draw x_0 from the frozen training replay buffer; noise-start evaluations draw each pixel from $\mathrm{Uniform}(0, 1)$. Active states are initialised from $\mathcal{N}(0, I)$ and carried through the chain without resets. The image-only FIFO buffer (capacity $10{,}000$; replay ratio 0.80 on MNIST, 0.95 on Fashion) stores no states, so each sampling call re-initialises h even for replay inputs; persistence of paired (x, h) states is a separate experimental setting.

Shared-state updates. Both gradients are evaluated at the same pre-update pair (x_t, h_t). Hidden states take plain SGD steps without momentum, clipping or Langevin noise, unlike latent-noise PC samplers (Zahid et al., 2024; Oliviers et al., 2024); hidden refinement is deterministic given (x_t, h_t), and the Gaussian perturbation applies only to the image after gradient evaluation.

Algorithm 3 Joint PC–Langevin sampling for training negatives or generation

Require: frozen $\boldsymbol{\theta}$; initial inputs x_0; chain length K_s; rates γ, δ; noise σ; clip g_c; smoothness λ_s.
1: Initialise h_0 from $\mathcal{N}(0, I)$ with the shapes defined by the active PC layers; freeze $\boldsymbol{\theta}$.
2: **for** $t = 0, \ldots, K_s - 1$ **do**
3: Recompute predictions and errors at (x_t, h_t); set $\mathcal{S}_t = \sum_i [G_{\boldsymbol{\theta}}(x_{t,i}, h_{t,i}) + \lambda_s E_{\mathrm{sm}}(x_{t,i})]$.
4: Compute $(g_t^x, g_t^h) = (\partial_{x_t} \mathcal{S}_t, \partial_{h_t} \mathcal{S}_t)$ in one differentiation pass.
5: Set $h_{t+1} \leftarrow h_t - \gamma g_t^h$; draw fresh $\boldsymbol{\omega}_t \sim \mathcal{N}(0, I)$.
6: Set $x_{t+1} \leftarrow \Pi_{[0,1]}[x_t - \delta\, \mathrm{clip}(g_t^x, -g_c, g_c) + \sigma \boldsymbol{\omega}_t]$.
7: **end for**
8: Return $\mathrm{stopgrad}(x_{K_s})$; evaluation leaves the replay buffer and weights unchanged.

The optional image-smoothness term is $E_{\mathrm{sm}}(x) = \frac{1}{2}[\mathrm{mean}((\Delta_i x)^2) + \mathrm{mean}((\Delta_j x)^2)]$; its coefficient is zero for main-table GEB-PCN generation and training negatives.

For a single-state model with an output potential that depends on h but not directly on x, let $J_u(x) = \partial u_{\boldsymbol{\theta}}(x)/\partial x$. The per-image gradients of the joint sampling objective are

$$\begin{aligned} g_t^h &= \lambda_{\mathrm{pc}}[h_t - u_{\boldsymbol{\theta}}(x_t)] + \partial_h E_{\mathrm{out},\boldsymbol{\theta}}(h_t), \\ g_t^x &= \lambda_{\mathrm{pc}} J_u(x_t)^\top [u_{\boldsymbol{\theta}}(x_t) - h_t] + \lambda_s \nabla_x E_{\mathrm{sm}}(x_t). \end{aligned} \tag{12}$$

The image gradient is not recomputed at h_{t+1}: the output potential first changes the hidden state, and the resulting prediction error changes subsequent image gradients (Equation (12)).

Gradient scaling and computation. The batched objective is a *sum* of per-image energies (no $1/B$ factor), unlike the batch-mean objective of fixed-image inference in the main recipes. For the

single-state models, the configured PC coefficient is divided by the number of state elements per image: final coefficients are $1/256$ on MNIST and $0.003/192$ on Fashion; CIFAR-10 uses configured weight 0.1, i.e. $0.1/(256 \times 8 \times 8) = 6.103515625 \times 10^{-6}$, retained without further division at evaluation. Both x and h derivatives use that same energy.

The implementation clears cached predictions before each energy evaluation and carries $(x_t, h_t, \text{random key})$ through a compiled JAX loop whose single differentiation call returns both gradients. Weights and chain initial values are detached within the loop; sampled inputs and inferred states are also detached before the parameter update, following the conventional contrastive update (Du & Mordatch, 2019) and omitting derivatives through the finite-chain sampling distribution (Du et al., 2021). Batches of 128 bound active-state memory; each batch starts a fresh chain with its own random subkey.

C Architectures and training configurations

We specify the three backbones, training and baseline configurations, and compute records. CIFAR-10 evaluation and endpoint selection are in Appendix D.6.

C.1 Architectures across datasets

GEB-PCN uses a ReLU MLP on MNIST, a SiLU hybrid conv–MLP on Fashion-MNIST, and the released IGEBM large residual architecture on CIFAR-10, with a single PC vode at the deepest MNIST hidden layer, Fashion trunk position 2, or CIFAR-10 slot 1 after residual stage 2 (Tables 3 to 5; Figure 7). The CIFAR-10 model has $18{,}591{,}105$ parameters, three residual blocks per stage, 2×2 average-pool downsampling in each stage's first block, spectral normalisation with one power iteration, leaky-ReLU 0.2 and dropout 0.5 (disabled for generation); the final stage, ReLU, global sum pooling and energy head act on the inferred second-stage feature map.

Output-head configuration. Each backbone supports a direct linear head of width 1 or an MSE head of configurable width d with $E_{\text{out},\boldsymbol{\theta}} = \|f_{\boldsymbol{\theta}}\|_2^2/d$; both give a scalar potential. Main-table results use MSE $d = 128$ on MNIST and $d = 64$ on Fashion-MNIST; CIFAR-10 retains the direct head. The MSE head uses a fixed zero target, so its squared output discrepancy requires no inferred output activity; removing internal PC retains the same output energy.

Across all three backbones, the prediction-error term enters the energy and the backward path is truncated at the vode, so the *earliest* PC vode determines the input gradient $\partial E/\partial \mathrm{x}$ (Section 3.3); parameter gradients within the preceding feed-forward prefix still use automatic differentiation.

Grayscale GEB-PCN, MCPC and VAE use backbones of approximately 400k parameters (Table 1); bPC retains its native $549{,}678$-parameter MLP and DCPC its $319{,}540$-parameter architecture. CIFAR-10 baselines use an 18.59M-parameter MCPC MLP, 18.59M convolutional bPC, 18.61M DCPC, a 0.72M VAE and an 18.59M widened VAE control (Appendix C.6).

Table 3: MLP backbone (MNIST): a 4-hidden-layer feed-forward network with a single PC vode at the deepest hidden layer; $431{,}232$ parameters with the main-table MSE128 head.

Stage	Layers	Width	PC vode
Input	flattened 784	—	—
Hidden	$4\times$ Linear + ReLU	256	—
PC vode	position 3 (deepest hidden)	256	✓
Head	Linear (direct or MSE)	1 (direct) / d (MSE)	—

C.2 Grayscale GEB-PCN training

Main-table GEB-PCN models use batch size 128, Adam with cosine decay to 10^{-6} and 100 epochs ($46{,}800$ updates). Initial weight rates are 10^{-4} (MNIST) and 2×10^{-4} (Fashion). The image-only FIFO replay buffer holds $10{,}000$ samples; the replay ratio is 0.80 on MNIST (with contrastive regularisation $\beta = 0.75$, centre $c = 1$) and 0.95 on Fashion ($\beta = 0$), with zero weight decay and

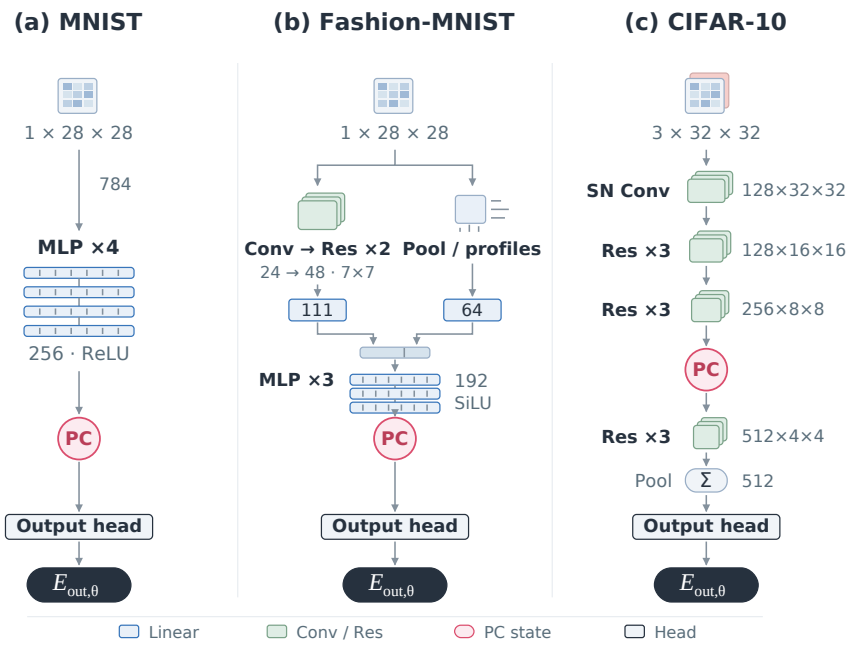

Figure 7: Main-table GEB-PCN architectures. **(a)** MNIST MLP. **(b)** Fashion-MNIST hybrid with local convolutional and global image-profile branches concatenated before the MLP trunk. **(c)** CIFAR-10 residual backbone, with three blocks per stage. Red circles mark the inferred PC states. All three outputs denote the scalar potential $E_{\text{out},\boldsymbol{\theta}}$. The output head is configurable as a direct or MSE readout, with its type and width specified in the accompanying text. Feature-map shapes use channels $\times$ height $\times$ width; glyphs are schematic. Exact configurations are listed in Tables 3 to 5.

Table 4: Hybrid conv–MLP backbone (Fashion-MNIST); 410,591 parameters with the main-table MSE64 head (398,432 with a direct scalar head). Local convolutional features and pooled image profiles are concatenated before the MLP trunk, with one PC state at position 2.

Stage	Configuration	Role
Input	$28\times28\times1$	pixels
Conv stem	Conv + SiLU, channels $[24, 48]$; 2×2 average pools	downsample to 7×7
Local branch	2 separable res-blocks at 48 channels; project to 111	local code
Global branch	pooled image and row/column profiles; project to 64	global code
Trunk (MLP)	concatenate; $3\times$ Linear + SiLU, width 192	hidden
PC vode	position $\{2\}$ in trunk	✓
Head	Linear (direct or MSE)	energy

Table 5: CIFAR-10 GEB-PCN backbone. Feature-map shapes use channels $\times$ height $\times$ width. The PC state follows stage 2.

Stage	Operation	Output shape
Input	RGB image	$3 \times 32 \times 32$
Stem	Spectral convolution + leaky-ReLU	$128 \times 32 \times 32$
Stage 1	3 residual blocks, downsample	$128 \times 16 \times 16$
Stage 2	3 residual blocks, downsample	$256 \times 8 \times 8$
PC vode	Slot 1, free hidden state	$256 \times 8 \times 8$
Stage 3	3 residual blocks, downsample	$512 \times 4 \times 4$
Pool	ReLU + global sum	512
Head	Linear (direct or MSE)	1 (direct) / d (MSE)

no input smoothness. Fashion's MSE64 configuration was selected by validation FID before the

five-seed final refits. The active PC state uses 32 fixed-image SGD updates at rate 0.01 for the contrastive loss.

C.3 PC-ENERGY WEIGHTING

The configured PC weight is 1 on MNIST and $(0.1, 0.03, 0.003)$ over three equal training phases on Fashion, divided by the state widths (256/192): effective coefficients $1/256$ and $0.003/192$. This schedule changes the weight during training; sampling uses the weight stored with the evaluated checkpoint.

C.4 CIFAR-10 GEB-PCN TRAINING AND REPLAY CONTINUATION

The $18{,}591{,}105$-parameter backbone in Table 5 uses one $256 \times 8 \times 8$ PC state with configured weight 0.1, mean reduction and 32 fixed-image state updates at rate 0.01. Its direct scalar head has per-sample contrastive regularisation (coefficient 1, centre 0). Training seed 42 uses all $50{,}000$ CIFAR-10 training images without augmentation, batch size 128, dequantisation and Adam at 3×10^{-4} with moments $(0, 0.999)$. Negative chains take 60 joint steps (image rate 10, noise 0.005, gradient clip 0.01).

Training used a $10{,}000$-image quantised FIFO replay buffer with ratio 0.95 and writeback. After monitored FID rose, we restored the preceding epoch-80 weights, optimiser, buffer and random states, then continued for ten epochs to step $35{,}100$. This continuation preserved particle identities by writing updates to sampled buffer indices, used replay ratio 1.0, and refreshed 13 of each 128 particles with noisy training images (noise scales $0.05, 0.1, 0.2, 0.4$). Hidden states were reinitialised. Identity-preserving writeback, replay ratio and refresh changed together, so the continuation does not isolate any one effect.

C.5 BASELINE ARCHITECTURES AND PARAMETER SETTINGS

The grayscale baselines use training seeds 0–4. Table 6 collects training budgets and optimiser settings for EBM, MCPC, bPC, VAE, and DCPC; grayscale genPC and hybridPC follow the released recipe described below. Generation readouts, evaluation seeds and reference sets are in Appendix D.2.

Energy-based model (EBM). The main-table EBM on MNIST and Fashion-MNIST is a ReLU MLP $784 \to 256 \to 256 \to 256 \to 256 \to 1$ (398,593 parameters, $[0, 1]$ inputs), trained for 100 epochs with batch size 128 and Adam (rate 10^{-4}, moments $(0.9, 0.999)$, cosine decay to 10^{-6}). Its image-only FIFO buffer holds $10{,}000$ samples at replay ratio 0.95; training negatives use 64 image updates at rate 10, noise 0.005 and gradient clip 0.01, with contrastive regularisation coefficient 1 and centre 0. Generation uses 500 updates with the same rate, noise and clipping, and no smoothing. All five seeds use epoch 100: MNIST FID/IS 1.204 ± 0.165/8.792 ± 0.078 and Fashion 14.614 ± 1.080/6.015 ± 0.081. This scalar MLP uses the shared IGEBM-style training code, not the original convolutional architecture.

On CIFAR-10, the EBM uses the convolutional IGEBM backbone without PC states (Appendix C.6).

Monte Carlo predictive coding (MCPC). MCPC (Oliviers et al., 2024) uses a fixed zero pseudo-input of dimension 30, four width-256 ReLU hidden layers and a 784-dimensional tanh image layer, with biases in all five affine maps ($406{,}800$ parameters). Inputs lie in $[-1, 1]$; hidden and observation noise variances are 1 and 0.16517, activity momentum is 0.9, and AdamW weight decay is 0.01.

Bidirectional predictive coding (bPC). bPC (Oliviers et al., 2026) uses the authors' unsupervised implementation: image-to-latent dimensions 784–256–256–30, $549{,}678$ parameters, leaky-ReLU/linear/tanh activations for intermediate/latent/image predictions on $[-1, 1]$ images, downward/upward energy coefficients $1/0.01$, activity momentum 0.5, noise variance 1 and AdamW weight decay 9.438×10^{-5}. We retain the source energy scaling and transfer the MNIST recipe to Fashion-MNIST without retuning.

Generative and hybrid predictive coding (genPC/hybridPC). The genPC (Rao & Ballard, 1999) and hybridPC (Tscshantz et al., 2023) comparisons use the released 25-epoch recipe of Oliviers et al. (2026) on MNIST and Fashion-MNIST, with ancestral decoder-mean readouts. Their CIFAR-10

entries are seed-0, epoch-100 transfers of the selected CIFAR bPC recipe, not CIFAR results reported by the original authors; their configuration is given below.

Variational autoencoder (VAE). The VAE (Kingma & Welling, 2014) has two width-200 ReLU layers in each of its encoder and decoder, a 16-dimensional diagonal-Gaussian latent and 784 sigmoid outputs (404,816 parameters), with $[0, 1]$ images, a standard-normal prior, a Bernoulli reconstruction objective and cosine-decayed Adam.

Divide-and-conquer predictive coding (DCPC). DCPC (Sennesh et al., 2024) uses Gaussian latent dimensions $20 \to 128 \to 256$ and a Bernoulli image likelihood (319,540 parameters), with four persistent particles per image, one inference sweep per batch and Adam with AMSGrad. A validation-loss scheduler reduces the rate by 0.9 (patience 100, cooldown 50); the inference rate follows the weight rate.

Table 6: Fixed grayscale baseline training settings, shared by MNIST and Fashion-MNIST. T_h counts activity updates per training batch (sweeps for DCPC); a dash means no PC activity inference. Rates are rounded for display; the accompanying configurations preserve their full precision.

Model	Epochs	Batch	Optimiser	Weight rate	T_h	State rate
EBM	100	128	Adam	10^{-4}	—	—
MCPC	100	900	AdamW	1.7592×10^{-3}	1,000	0.018447
bPC	100	256	AdamW	4.6606×10^{-3}	250	0.008907
VAE	100	128	Adam	10^{-3}	—	—
DCPC	2,000	4,096	AMSGrad	10^{-3}	1	10^{-3}

C.6 CIFAR-10 BASELINE CONFIGURATIONS

The EBM uses the full 50,000-image training set and seed 42. The MCPC, bPC, VAE, DCPC, genPC and hybridPC comparisons use seed 0, 100 epochs and a 45,000/5,000 train/validation partition of the official training set, without augmentation. Their main-table scores use the common 50,000-image evaluation in Appendix D.6.

EBM baseline. The CIFAR-10 EBM uses the same 18,591,105-parameter IGEBM convolutional residual backbone as GEB-PCN, with a scalar direct output and no PC states, prediction-error term or hidden-state inference (stage widths $[128, 256, 512]$, three residual blocks per stage, spectral normalisation, training dropout 0.5). It uses all 50,000 training images, $[0, 1]$ pixels with dequantisation, no augmentation and batches of 128. The IGEBM Adam variant uses rate 3×10^{-4} and moments $(0, 0.999)$; per-sample contrastive regularisation has coefficient 1 and centre 0. Training negatives use 60 image-only Langevin updates at rate 10, noise 0.005 and clip 0.01; the quantised FIFO replay buffer holds 10,000 images at replay ratio 0.95.

The epoch-150 checkpoint (step 58,500) was selected retrospectively; its sampling and metrics are given in Appendix D.6. Training duration and replay policy differ from the selected GEB-PCN branch, so their comparison does not isolate the effect of PC.

Configuration selection and capacity. MCPC, bPC, VAE and DCPC retain their best recorded completed epoch-100 configurations for the declared mean readouts. For bPC, VAE and DCPC, six small-network configurations per family were screened at epoch 30 with 1,024 generated images and 5,000 validation references; two per family advanced by validation FID. Both small VAE finalists completed 100 epochs, while small bPC/DCPC continuations were stopped for capacity alignment. The selected 716,723-parameter VAE ($\sigma = 0.2$) had validation FID 182.33 on 5,000 generated images, versus 228.19 for an 18,588,021-parameter widened VAE control. Selection and capacity checks were exploratory, and the main table is not fully parameter matched.

Predictive-coding baselines. MCPC uses a width-2025 five-layer ReLU MLP (18,594,597 parameters), 1,000 training activity updates and weight/activity rates $0.0017591587/0.0184465089$. bPC uses the author implementation with convolutional channels $[77, 154, 308, 616, 1215]$, latent width 256 (18,585,481 parameters), 128 training activity updates, upward and downward weights $0.01/1$, state rate 0.0089065652 and AdamW base rate 0.0004208161. MCPC and joint bPC take 10,000

activity updates at generation and return decoder means; ancestral bPC uses no iterative refinement. The genPC and hybridPC transfers use the selected bPC architecture, split and non-variant settings at a fixed epoch-100 endpoint, not released CIFAR-10 author results. Their upward/downward weights are $0/1$ and $1/1$, respectively; genPC uses zero-state initialisation and downward energy, while hybridPC uses bottom-up initialisation, downward energy for inference and joint energy for learning. Both use ancestral decoder-mean readouts.

Latent-variable baselines. The VAE and DCPC use the released convolutional code with Gaussian likelihoods, standard-normal priors and latent widths $40/64$. The selected VAE uses observation scale 0.2 and weight rate 10^{-3}; DCPC (18,605,133 parameters) uses scale 0.1, three inference sweeps and weight/state rates 10^{-3}. Both use ten particles, Adam with AMSGrad and the source validation-loss scheduler. They return independent prior-sampled decoder means without observation noise or averaging.

C.7 Computational resources

GEB-PCN and MCPC train in JAX on one RTX 5090 GPU per trial; the 431k-parameter MNIST MSE128 model takes 598–642s for 100 epochs across its five confirmation seeds. Sampling and inference budgets are in Appendix B; baseline configurations in Appendices C.5 and C.6.

The 100-epoch capacity-study VAE, DCPC, bPC and MCPC training calls took 4.20, 19.70, 3.17 and 4.17 hours, respectively, on RTX 5090 workers. Concurrent trials shared devices; these are execution costs, not controlled speed comparisons. Final sampling and scoring are separate. The selected small VAE's additional test score reuses saved features.

D Evaluation protocols and sensitivity

This section fixes the grayscale evaluators and main-table endpoints, then examines metrics, training trajectories and sampler initialisation, with each training-seed cohort stated separately. The final subsection specifies CIFAR-10 scoring and endpoint selection.

D.1 Domain ResNet-18 evaluators

Architecture and feature readout. MNIST and Fashion-MNIST use separately trained ResNet-18 classifiers (He et al., 2016) (Table 7): a stride-1 3×3 stem without initial max-pooling, bias-free 3×3 convolutions with batch normalisation, and stride-2 downsampling with 1×1 BN shortcuts in the first block of stages 2–4; 11,172,810 parameters including the 512-to-10 classifier. Global-average-pooled 512-dimensional features (not unit-normalised) feed FID, KID and PRDC; softmax probabilities feed IS. All generator families on a given dataset share the same frozen evaluator with eval-mode batch normalisation.

Table 7: ResNet-18 architecture shared by the two grayscale evaluators. Shapes are channels $\times$ height $\times$ width. A basic block has two 3×3 convolutions; stage stride applies to its first block.

Component	Operation	Stride	Output shape
Input	Standardised grayscale image	—	$1 \times 28 \times 28$
Stem	3×3 convolution, BN, ReLU	1	$64 \times 28 \times 28$
Residual stage 1	2 basic blocks, 64 channels	1	$64 \times 28 \times 28$
Residual stage 2	2 basic blocks, 128 channels	2	$128 \times 14 \times 14$
Residual stage 3	2 basic blocks, 256 channels	2	$256 \times 7 \times 7$
Residual stage 4	2 basic blocks, 512 channels	2	$512 \times 4 \times 4$
FID feature	Adaptive global average pooling	—	512
Classifier	Linear layer	—	10 logits

Training and checkpoint selection. Each evaluator trains from random initialisation on 55,000 official training images for 30 epochs (batch 256, SGD at rate 0.1, momentum 0.9, weight decay 5×10^{-4}, cosine schedule, no augmentation), with a stratified 500-per-class validation split

(split and training seeds 20260903). Images are standardised with the training-partition statistics ($(0.130621, 0.308077)$ for MNIST, $(0.286086, 0.353066)$ for Fashion-MNIST), shared by real and generated images. The minimum-validation-cross-entropy checkpoint is retained (epoch 24 for MNIST, 12 for Fashion-MNIST). Reference and generated images are excluded from evaluator training and selection.

Classification performance and reference calibration. Table 8 reports validation and reference accuracy (QMNIST `test50k` for MNIST; the official test set for Fashion). Real–real FID between disjoint reference halves (first/last 25,000 QMNIST images; a fixed stratified 5,000-image Fashion subset against its complement) is small. One evaluator is trained per dataset; the five-seed deviations in the main results concern generator training only. FID comparisons remain specific to this fixed representation. CIFAR-10 uses the separate Inception-v3 protocol in Appendix D.6.

Table 8: Selected evaluator checkpoints and reference-only checks. Validation CE is mean cross-entropy per image. Reference accuracy uses all reference images; real–real FID uses the disjoint halves described above. Each column represents one frozen evaluator.

Quantity	MNIST evaluator	Fashion-MNIST evaluator
Selected epoch / training epochs	24/30	12/30
Validation images	5,000	5,000
Validation CE ↓	0.02162	0.20905
Validation accuracy ↑	99.42%	93.00%
Reference population	QMNIST `test50k`	Fashion-MNIST test
Reference images	50,000	10,000
Reference accuracy ↑	99.44%	92.36%
Real–real FID ↓	0.01748	0.04996

D.2 Grayscale metrics and baseline endpoints

Table 9 summarises the frozen checkpoints and generation readouts. Each seed generates 10,000 images, scored against QMNIST `test50k` (50,000 images) on MNIST or the official Fashion-MNIST test set (10,000 images). The fixed evaluators of Appendix D.1 provide 512-dimensional features for FID (Heusel et al., 2017), KID (Bińkowski et al., 2018) and precision/recall/density/coverage (PRDC; Kynkäänniemi et al., 2019; Naeem et al., 2020); IS uses class probabilities (Salimans et al., 2016). The genPC/hybridPC checkpoints follow the released recipe of Oliviers et al. (2026). Architectures, parameter counts and detailed EBM settings are in Appendices C and C.5.

Metric and uncertainty conventions. KID uses 100 subsets of 1,000; PRDC uses fixed 5,000-image subsets per distribution with $k = 3$. Tables report mean $\pm$ sample SD across generator-training seeds, including for IS. Finite-sample FID and IS have generator-dependent bias (Chong & Forsyth, 2020). Paired intervals use 100,000 seed-bootstrap draws for matched interventions; MNIST GEB-PCN comparisons with other models instead resample the two training-seed cohorts independently. These small-cohort intervals are descriptive and unadjusted for multiple comparisons. Reported paired sign-flip tests assume a symmetric null; with five pairs their minimum two-sided p value is 0.0625.

Table 9: Grayscale checkpoints and generation readouts. All rows use training seeds 0–4. K counts image or activity updates; the GEB-PCN $500/200$ entry is MNIST/Fashion. Joint bPC is an additional readout of the same weights; the bPC, genPC and hybridPC main-table rows use ancestral means. A dash denotes sampling without iterative updates.

Model / readout	Epoch	Start and returned sample	K
GEB-PCN	100	Frozen replay; joint PC–Langevin	$500/200$
EBM	100	Frozen replay; image-only Langevin	500
MCPC	100	Zero activities and fixed zero pseudo-input; decoder mean	10,000
bPC (joint)	100	Zero dummy prior, downward initialisation; conditional mean	10,000
bPC (ancestral)	100	Sampled prior latents; conditional mean	—
genPC / hybridPC	25	Ancestral decoder mean	—
VAE	100	Independent prior path; decoder mean	—
DCPC	2,000	Independent prior path; decoder mean	—

The post-training head-removal control uses 2,000 generated images per endpoint and matched 2,000-feature PRDC subsets, retaining the same evaluator and FID references; these scores are not main-table endpoints.

Readout and display details. MCPC uses a fixed zero pseudo-input and returns decoder means mapped to $[0, 1]$. Both bPC readouts return conditional image means without observation noise in batches of 256: evaluation seed $100000 + s$ for training seed s, plus 1000000 for the ancestral readout. Their $[-1, 1]$ outputs are mapped by $(x + 1)/2$, clipped and rounded to uint8. VAE and DCPC use independent prior paths without particle averaging.

Displayed samples and PRDC cohorts. The MNIST GEB-PCN strip in Figure 3a shows archived indices $0, 1, 5, 6, 13$, selected for digit diversity by scanning archive order, omitting predicted digit 1 and taking the first occurrence of five other evaluator-predicted classes $(0, 2, 5, 7, 3)$, without confidence or quality ranking; all other strips show the first five archived images. These choices do not alter the full populations used for quantitative metrics. All strips use seed 0 except CIFAR-10 GEB-PCN and EBM (seed 42); CIFAR-10 GEB-PCN uses the selected epoch-85 checkpoint, whose stored-particle sample batch is separate from its FID population. Panel (c) reuses the epoch-100 GEB-PCN, EBM, MCPC and VAE evaluations for seeds 0–4 on both grayscale datasets without further checkpoint selection. The bPC, genPC and hybridPC strips in panel (a) use ancestral decoder-mean readouts. Fashion panels (a,c) use the hybrid MSE64 GEB-PCN and scalar MLP EBM; its panel (b) uses a distinct direct-head cohort $7, 42, 123$. MNIST panels (a,c) use the selected MSE128 replay population; its panel (b) uses MSE16 training trajectories.

D.3 MNIST NOISE-START SENSITIVITY

The main MNIST model (431,232-parameter MSE128 MLP, recipe in Appendix C) generates with 500 joint steps at image/state rates $3/0.03$, Gaussian hidden initialisation, noise 0.005, gradient clip 10^6 and no input smoothing. The sampler was selected by replay FID on three other training seeds against the official MNIST test reference, before QMNIST evaluation of seeds 0–4. This test-informed selection makes the noise-start comparison exploratory.

On the same five frozen MSE128 checkpoints, changing generation rates from $10/0.01$ to the selected $3/0.03$ improves replay FID from 1.389 ± 0.206 to 0.870 ± 0.024, but worsens noise-start FID from 1.077 ± 0.072 to 4.302 ± 0.251 (10,000 images per start, QMNIST `test50k` reference, mean $\pm$ sample SD over seeds 0–4). The scalar MLP EBM has noise-start FID 2.887 ± 0.599 and IS 7.962 ± 0.171 under this protocol. Noise-start quality is therefore sampler-sensitive: a sampler selected for replay quality need not perform well from noise. The training curves use MSE16 checkpoints rather than the main-table MSE128 models.

Table 10: MNIST frozen-checkpoint robustness audit: 10,000 samples per checkpoint, external QMNIST `test50k` reference, five training seeds per model. GEB-PCN uses the selected MSE128 replay sampler. PRDC uses fixed 5,000-sample subsets per distribution, $k = 3$. Values are mean $\pm$ sample SD across seeds 0–4. KID is multiplied by 10^3 for display. Bold denotes the best mean per metric, without a significance claim. †DCPC uses its 2,000-epoch endpoint; the other models use epoch 100.

Model	FID ↓	KID $\times 10^3$ ↓	Precision ↑	Recall ↑	Density ↑	Coverage ↑
GEB-PCN (ours)	$\mathbf{0.870 \pm 0.024}$	$\mathbf{1.236 \pm 0.167}$	0.850 ± 0.009	0.663 ± 0.012	$\mathbf{0.816 \pm 0.010}$	$\mathbf{0.630 \pm 0.006}$
EBM	1.204 ± 0.165	1.981 ± 0.566	$\mathbf{0.851 \pm 0.014}$	0.422 ± 0.026	0.793 ± 0.022	0.441 ± 0.009
MCPC	3.039 ± 0.291	3.587 ± 0.632	0.770 ± 0.007	0.864 ± 0.018	0.606 ± 0.016	0.497 ± 0.016
bPC (joint mean)	11.575 ± 0.767	24.467 ± 2.983	0.684 ± 0.009	0.842 ± 0.025	0.421 ± 0.013	0.219 ± 0.011
bPC (ancestral mean)	8.695 ± 0.629	18.246 ± 2.319	0.674 ± 0.008	$\mathbf{0.879 \pm 0.016}$	0.405 ± 0.011	0.225 ± 0.013
VAE	2.997 ± 0.258	5.573 ± 0.629	0.785 ± 0.007	0.815 ± 0.009	0.643 ± 0.008	0.505 ± 0.008
DCPC†	47.121 ± 0.163	121.106 ± 0.630	0.222 ± 0.009	0.041 ± 0.038	0.078 ± 0.003	0.008 ± 0.001

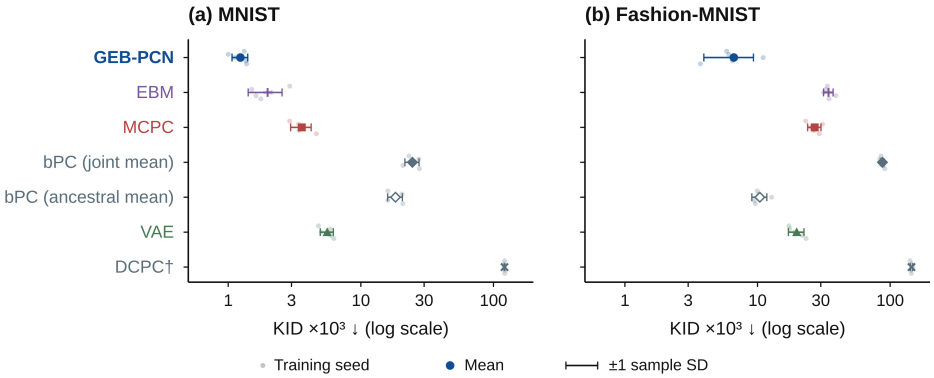

Figure 8: KID across all grayscale baselines and both bPC readouts, on **(a)** MNIST and **(b)** Fashion-MNIST. Faint dots show all five training seeds; large markers and horizontal bars show the arithmetic mean and sample SD. Each row uses the same 10,000 generated images per seed as Table 1. KID uses 100 subsets of 1,000, is multiplied by 10^3 for display, and is plotted on a shared logarithmic axis; lower is better. †DCPC uses epoch 2,000; the other models use epoch 100. Comparisons retain the main experiment's differing training and generation recipes.

D.4 Metric sensitivity at fixed grayscale endpoints

These checks reuse the main-table checkpoints and generated populations under the fixed evaluators of Appendix D.2: Table 10 compares all grayscale baselines and both bPC readouts across six MNIST metrics; Figure 8 visualises KID on both datasets. Differences across metrics therefore do not arise from selecting new checkpoints or changing the generator.

On MNIST, GEB-PCN has the lowest mean FID and KID and the highest mean density and coverage; the EBM has slightly higher mean precision and ancestral bPC the highest mean recall. GEB-PCN is also lowest in KID on Fashion-MNIST (Figure 8), while its lower MNIST KID coexists with lower recall than MCPC, both bPC readouts and VAE (Table 10). These are metric trade-offs within the chosen feature spaces; they do not remove the test-informed model selection discussed in Section 4.2.

D.5 FID trajectories across retained checkpoints

Figure 3b uses the frozen domain ResNet-18 evaluators and reference sets of the grayscale main table, with 10,000 generated images per checkpoint. MNIST uses training seeds 0–4 for GEB-PCN, MCPC and VAE at epochs $20, 40, 60, 80, 100$; the MNIST GEB-PCN curve uses an MSE16 head and rates $10/0.01$, not the selected MSE128 endpoint in Table 1. Fashion MCPC and VAE use seeds 0–4 at epochs $40, 60, 80, 100$; Fashion GEB-PCN uses seeds $7, 42, 123$ at epochs $45, 60, 80, 100$, a distinct cohort not paired with the other methods and without checkpoints before epoch 45. These trajectories compare training progress within each cohort, not endpoints across methods.

GEB-PCN uses each checkpoint's frozen replay buffer and stored PC coefficient, with $500/200$ joint updates on MNIST/Fashion and Fashion smoothness weight 0.06; MCPC uses its $10{,}000$-update decoder-mean generator; evaluation seeds are fixed across epochs. VAE trajectories come from reruns with intermediate checkpoints saved (architecture, loss, optimiser, schedule and data cycle preserved); every rerun's epoch-100 weights and FID reproduce the archived endpoint exactly. No checkpoint is selected using these scores.

Markers show mean FID across training seeds; shaded bands show sample SD and faint lines individual seeds.

D.6 CIFAR-10 METRICS

CIFAR-10 main-table entries use $50{,}000$ generated images and all $50{,}000$ official training images as the FID reference, using TorchMetrics Inception-v3 2048-dimensional features for FID and Inception logits for IS. For models selected before the $50{,}000$-image evaluation, checkpoints, sampling starts and readouts match the $5{,}000$-image selection runs. The genPC and hybridPC CIFAR-10 transfers use fixed epoch-100 checkpoints, evaluation seed $1{,}100{,}000$ and ancestral-mean readouts. MCPC, bPC, genPC, hybridPC, VAE and DCPC were trained on a $45{,}000$-image subset, but the full official training set is used for their $50{,}000$-image reference statistics. Images in $[-1, 1]$ (MCPC, bPC, genPC, hybridPC, VAE, DCPC) are converted by $\lfloor 255\ \mathrm{clip}((x+1)/2, 0, 1)\rfloor$; $[0, 1]$ images (GEB-PCN, EBM) by $\lfloor 255\ \mathrm{clip}(x, 0, 1)\rfloor$ directly. IS uses ten splits of the generated population; split SD is not training-seed uncertainty. The $5{,}000$-generated-image selection evaluations used $5{,}120$ reference images: GEB-PCN and EBM FID references contain $5{,}120$ training images from a seed-42 shuffled loader, whereas MCPC, bPC, VAE and DCPC use the first $5{,}120$ official test images in train/test-reference checks. Those scores remain selection and sensitivity records, not the main-table estimates. The frozen-checkpoint controls below use fixed $5{,}120$-image train/test references; the selected noise horizon is reevaluated on fresh images. Absolute scores are not compared across datasets or feature extractors.

CIFAR-10 endpoint selection and sensitivity. The GEB-PCN continuation in Appendix C.4 was assessed at epochs $81, 82, 85, 90$ from both stored particles and uniform noise, using training seed 42, evaluation seed 42, 200 joint steps, $5{,}000$ generated images and $5{,}120$ training references per endpoint (Table 11). Epoch 85 had the lowest recorded particle-start FID (72.187; IS 4.976) and supplies the frozen checkpoint, but not the numerical score, for the $50{,}000$-image main table (FID 62.932, IS 5.153). This is retrospective selection within one training branch, not an independent confirmation seed.

Table 11: CIFAR-10 checkpoint selection on $5{,}000$ generated images and $5{,}120$ training references. Both starts use the same weights; epoch 90 is the final completed checkpoint.

Epoch	Particle FID ↓	Noise FID ↓	Particle IS ↑	Noise IS ↑
81	89.122	243.058	3.843	1.608
82	83.228	276.544	4.419	1.376
85	72.187	259.067	4.976	1.456
90	88.046	460.475	4.830	1.165

The EBM baseline uses its retrospectively selected epoch-150 checkpoint (step $58{,}500$), replay start and 200 image updates at the training rate, noise and clipping (training/evaluation seed 42). Its selection-stage $5{,}000$-image FID/IS was $87.918/3.701$; the $50{,}000$-image reevaluation gives $78.746/3.795$. Both methods have one training seed, and their training durations and replay policies differ.

Frozen-checkpoint controls. At epoch 85, a fixed-reference $5{,}000$-image control gives test FID $82.53 \to 71.88$ for stored particles and $431.98 \to 256.80$ for uniform noise before and after 200 updates. The initialisation gap therefore remains large after refinement. A separate common-start screen on $2{,}000$ noise images at $K = 200, 500, 1000, 2000$ selects $K = 500$ by training-reference FID (180.95; test-reference FID 181.65); a fresh $5{,}000$-image population at that horizon gives test FID 176.50. These are different reference and sampling populations, not paired improvements over the checkpoint-selection scores. At least $9{,}207$ of the $10{,}000$ epoch-85 buffer entries had a

noisy-training-image refresh since epoch 80, so particle-start performance measures refinement of a data-refreshed population rather than synthesis independent of training images. Neither these single-seed controls nor the continuation isolate a PC-layer-specific causal advantage.

E CONDITIONAL GENERATION AND INFERENCE

E.1 CLASS-CONDITIONAL MNIST GENERATION

FiLM parameterisation. The one-hot label $y \in \mathbb{R}^{10}$ is mapped to $z_y = W_e y + b_e \in \mathbb{R}^{32}$; each of the four width-256 hidden layers has its own affine map from z_y to $[\gamma_\ell; \beta_\ell] \in \mathbb{R}^{512}$, applied to the linear preactivation before ReLU and before any PC state (Perez et al., 2018). A selected layer contributes the condition-dependent error

$$\varepsilon_\ell(x, h; y) = h_\ell - \phi_\ell(\gamma_\ell(z_y) \odot (W_\ell s_{\ell-1} + b_\ell) + \beta_\ell(z_y)) \,. \tag{13}$$

The condition therefore changes both the hidden prediction target and the input gradient induced by its squared error; it is neither concatenated to the image nor restricted to the output head.

Conditional energy and training. When the partition function is finite, substituting Equation (13) into the joint energy induces a conditional EBM,

$$p_\theta(x, h \mid y) = \frac{\exp[-G_\theta(x, h; y)]}{Z_\theta(y)}, \qquad Z_\theta(y) = \int \exp[-G_\theta(x', h'; y)]\, dx'\, dh'. \tag{14}$$

Ignoring the optional energy regulariser, the implemented finite-state loss $G_\theta(x^+, h^+_{\mathcal{T}}; y) - G_\theta(x^-, h^-_{\mathcal{T}}; y)$ contrasts a real image with a class-conditioned Langevin negative under the same label. The reported cohort has no mismatched-pair loss, and we do not claim that the finite practical kernel exactly samples Equation (14). At generation, y and z_y are clamped and only x and the enabled PC states are updated, making z_y a deterministic control code without prior, posterior inference or latent Langevin chain — closer to feature-conditioned input-space EBMs (Du & Mordatch, 2019; Grathwohl et al., 2020) than to latent-update methods (Nie et al., 2021).

Continuous condition paths. For $y_\alpha = (1-\alpha)e_a + \alpha e_b$, $\alpha \in [0, 1]$, affine linearity gives

$$z_{y_\alpha} = (1-\alpha) z_a + \alpha z_b, \quad (\gamma_\ell, \beta_\ell)_{y_\alpha} = (1-\alpha)(\gamma_\ell, \beta_\ell)_a + \alpha(\gamma_\ell, \beta_\ell)_b. \tag{15}$$

The energy and generated image need not vary linearly (the network, state relaxation and sampler are nonlinear): equation 15 is a reproducible interpolation probe, not a guarantee of semantic smoothness, and the reported quantitative experiment evaluates only the ten discrete endpoints.

Conditional generation results. The selected model uses an MSE8 fixed-zero output head, one internal state after hidden layer 3, 32 positive-phase state updates and 64 joint negative updates. Training runs 100 epochs with replay ratio $r = 0$ (Nijkamp et al., 2019), weight rate 7×10^{-4}, no weight decay and no wrong-label or global auxiliary loss, retaining the centred energy-magnitude penalty ($\alpha = 1$, centre 1) applied after batch-mean energy reduction. At evaluation, each checkpoint generates 10,000 balanced requests from noise in 500 joint updates; an independent digit ResNet-18 measures label adherence and provides features for comparison with QMNIST `test50k`.

Prospective selection. A fixed class-balanced 5,000-image subset of the MNIST training set serves as generator validation. Twelve configurations (FiLM width $\{8, 16, 32\}$ crossed with wrong-label and global-loss weights in $\{0, 0.05\}$) are screened at seed 20; the best three by macro conditional FID add seeds 21, 22. The selected FiLM32 model uses zero auxiliary weights and is retrained on all 60,000 training images at seeds 30–34; QMNIST is not used during selection. Validation FIDs use a PCA-128 feature projection, whereas the frozen QMNIST evaluation uses all 512 ResNet features, so their absolute values are not compared.

The untuned sampler uses image step 10, noise 0.005, hidden step 0.01 and 500 joint updates. A validation-only search over 14 settings (number of steps, image/hidden step sizes, noise scale, alternation; four candidates repeated on two refinement seeds) minimises $0.6\,\overline{\mathrm{FID}}_{\{2,8,9\}} + 0.4\,\mathrm{macro\ cFID}$ and selects image step 5, with every other setting unchanged, frozen before the paired five-checkpoint QMNIST evaluation. The complete search space is listed in Table 12.

Table 12: Conditional MNIST search protocol. A fixed 5,000-image generator-validation split is used throughout; QMNIST is reserved for the frozen five-seed confirmation.

Stage	Candidates	Advancement rule
FiLM model	FiLM width $\{8, 16, 32\}$; wrong-label weight $\{0, 0.05\}$; global-loss weight $\{0, 0.05\}$ (12 total)	Seed 20 screen; top 3 macro cFID candidates repeated at seeds 21, 22
Sampler	Steps $\{500, 750, 1000\}$; image step $\{5, 7.5, 10, 12.5, 15\}$; noise $\{0, 0.0025, 0.005, 0.01\}$; hidden step $\{0.005, 0.01, 0.02\}$; joint or $1/2$ alternating hidden updates (14 one-factor settings)	Seed 20 screen; top 4 weak-class scores repeated at seeds 21, 22
Training follow-up	Targeted changes among FiLM width $\{32, 48, 64\}$; MSE width $\{8, 16, 32\}$; learning rate $\{5, 7, 10\} \times 10^{-4}$; small wrong-label/global weights (12 targeted settings)	Seed 23 screen; top 4 repeated at seeds 24, 25; winner confirmed at five fresh seeds

Frozen five-checkpoint comparison. Table 13 reports mean ± sample SD over five training seeds. The two FiLM rows reuse the same checkpoints, so their difference isolates the evaluation sampler: reducing the image step $10 \to 5$ improves every reported aggregate metric, with paired checkpoint bootstrap intervals (step-5 minus step-10) of $[0.0130, 0.0226]$ for requested-label accuracy, $[-1.631, -1.058]$ for macro conditional FID, $[-0.545, -0.338]$ for global FID and $[0.0477, 0.0853]$ for recall.

Table 13: Frozen conditional MNIST evaluation (mean ± sample SD over five seeds; 10,000 balanced noise-initialised samples per seed). Weak FID averages digits 2, 8 and 9. The two FiLM GEB-PCN rows use the same trained checkpoints and differ only in image Langevin step size. Bold marks the best mean in each column.

Model / sampler	Label acc. ↑	Worst acc. ↑	Weak FID ↓	Macro cFID ↓	Global FID ↓
FiLM GEB-PCN, step 10	$.921 \pm .009$	$.766 \pm .022$	8.704 ± 3.303	$4.477 \pm .726$	$1.180 \pm .192$
FiLM GEB-PCN, step 5	$.939 \pm .009$	$.834 \pm .027$	5.983 ± 2.769	$3.121 \pm .592$	$.743 \pm .123$
High-LR FiLM, step 5	$.944 \pm .017$	$.785 \pm .121$	8.097 ± 3.848	3.588 ± 1.118	$1.227 \pm .367$
Conditional VAE	$\mathbf{.964 \pm .003}$	$\mathbf{.921 \pm .021}$	$\mathbf{1.419 \pm .308}$	$\mathbf{1.277 \pm .117}$	$\mathbf{.386 \pm .023}$

Model / sampler	KID ↓	Precision ↑	Recall ↑	Density ↑	Coverage ↑	IS ↑
FiLM GEB-PCN, step 10	$.00279 \pm .00110$	$.803 \pm .013$	$.662 \pm .036$	$.720 \pm .025$	$.565 \pm .026$	$8.960 \pm .091$
FiLM GEB-PCN, step 5	$.00161 \pm .00067$	$.820 \pm .011$	$\mathbf{.729 \pm .026}$	$.749 \pm .026$	$.619 \pm .034$	$9.154 \pm .029$
High-LR FiLM, step 5	$.00215 \pm .00113$	$.778 \pm .048$	$.555 \pm .042$	$.694 \pm .069$	$.472 \pm .068$	$9.132 \pm .179$
Conditional VAE	$\mathbf{.00100 \pm .00011}$	$\mathbf{.878 \pm .006}$	$.714 \pm .016$	$\mathbf{.989 \pm .014}$	$\mathbf{.740 \pm .013}$	$\mathbf{9.504 \pm .031}$

Fixed-label energy audit. The five selected checkpoints are evaluated on all 10,000 MNIST test images after fresh standard-normal state initialisation and the training protocol's 32 hidden-state updates (state step .01, $1/128$ batch-mean scaling). With $y = 1$ clamped, true ones have total energy .531 (95% t interval $[.524, .538]$ over checkpoints) versus .695 averaged over the other nine class means; negative total energy separates ones from the remaining digits at AUC .937 ($[.919, .956]$). The internal PC term gives AUC .946 versus .503 for the MSE output term, locating the condition signal in the FiLM-dependent prediction error. On the 10,000 step-5 samples per checkpoint, requested ones similarly have energy .530 versus .680 for other requested classes (AUC .916).

For a paired ten-condition audit, each initial state is reused across labels on a deterministic balanced subset of 100 test images per class. Correct-label energy averages 1.020 versus 1.067 over wrong labels, but $\arg\min_y G_\theta(x, h_\mathcal{T}; y)$ reaches only .242 accuracy because the ten conditions have different energy offsets: the audit supports lower energy for matching images within a fixed condition, not calibrated energies across labels or normalised conditional likelihoods. Figure 5**(c)** visualises the column-centred contrast $G(c, y) - \mathrm{mean}_{c' \neq y} G(c', y)$ (all ten diagonal means negative); beneath it are the first archived seed-30 step-5 samples for requested labels 0–9, without image or energy selection.

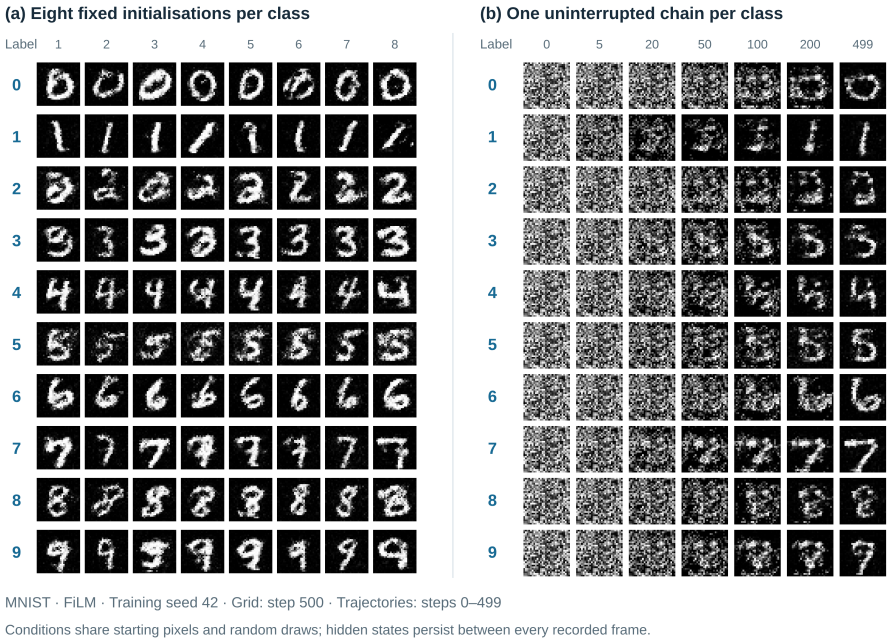

Figure 9: Qualitative class-conditional generation from a separate FiLM16 cohort, $r = 0$ and 100 epochs. **(a)** Eight samples per class ($y = 0, \ldots, 9$). **(b)** Each row starts from the same noise under a different class condition; columns show steps $0, 5, 20, 50, 100, 200, 499$. This grid uses a separately recorded, uninterrupted production chain at seed 42, with shared initial pixels, hidden-state randomness and input noise across labels. The first eight images per class are retained, without quality selection. It is separate from the quantitative population; recorded endpoints were checked against uninterrupted production calls.

Digits $2, 8, 9$ were the weakest classes under the step-10 sampler. The step-5 sampler reduces their mean class FIDs $7.003/11.622/7.486$ to $4.405/8.041/5.504$ ($-37.1\%/-30.8\%/-26.5\%$) and raises label accuracies $0.867/0.826/0.858$ to $0.909/0.875/0.885$. Increasing the training rate to 10^{-3} appeared stronger on the generator-validation seeds but did not reproduce the weak-class or recall gains on five fresh QMNIST-evaluated seeds (Table 13), so the reported comparison keeps the training weights fixed and changes only the evaluation sampler.

The conditional VAE (same five seeds, 100 epochs, width-200 encoder/decoder, 16-dimensional latent, one-hot label concatenated to the encoder input and decoder latent) has 408,816 trainable parameters versus 468,328 for FiLM GEB-PCN. It is stronger on label adherence and both conditional and global FID, although the tuned FiLM sampler has slightly higher recall: a clamped FiLM code controls the GEB-PCN energy from noise, while leaving a clear gap to a purpose-built conditional latent-variable baseline.

Matched conditional ablation. This separately trained cohort tests internal PC and is independent of the prospectively selected FiLM32 model. Five training-seed pairs differ only in the deepest PC state and its 32 inference updates; both retain the same fixed-target MSE output energy. Target accuracy is 0.9348 ± 0.0401 with internal PC off and 0.8986 ± 0.0266 with it on (paired on-minus-off difference -0.0362, 95% bootstrap interval $[-0.0650, -0.0048]$, four of five pairs favouring off; exact two-sided sign-flip $p = 0.125$). Macro FID is 3.782 off versus 6.371 on; internal PC's recall/coverage gains are small and uncertain. Conditional control in this FiLM/MSE recipe does not require internal state inference; the five-pair sample and test-informed recipe limit generalisation of that contrast.

F SAMPLING-ANALYSIS PROTOCOL AND COMPLETE RESULTS

The sampling controls in Section 5 use unregularised MSE128 MNIST MLP and MSE64 Fashion-MNIST hybrid PC-on checkpoints at epoch 100, with five training seeds 30–34. These are separate refits from the seeds 0–4 and model-specific replay starts of Table 1. Comparisons use paired image starts and image-noise streams where stated; replay buffers are not refreshed during evaluation. The coefficient sweep and joint/separate FID comparisons use 10,000 generated images per seed, whereas the continuous-chain profile diagnostic uses 2,000. FIDs from these population sizes are not combined as a single ranking.

Sampling-time PC-coefficient intervention. The saved, sum-normalised MNIST coefficient is $\lambda_{\mathrm{pc}} = 1/256$. On frozen MSE128 checkpoints, we multiply it by $\{1, 4, 16, 64, 256\}$ only while sampling. We do not change weights, output head, image step size, noise scale, gradient clip, state rate or image starts. Each seed receives paired replay-mixture and uniform-noise starts and image-noise streams across multipliers. Continuous chains are recorded at $K \in \{0, 50, 100, 200, 500, 1000\}$ with 10,000 images per endpoint; Table 14 shows the diagnostic horizons used in Figure 6a. At $K = 0$, the image populations agree across all multipliers. On the first diagnostic batch, the head/PC hidden-state force ratio decreases from 1.37 at $\times 1$ to 0.34, 0.086, 0.021 and 0.005 at $\times 4, 16, 64, 256$. The output head has zero direct pixel derivative under the implemented gradient truncation, while the PC pixel force scales with the multiplier. These measurements verify the post-training intervention, not a causal effect of PC during learning.

Table 14: **Sampling-time PC-coefficient sweep on frozen MNIST models.** Domain test FID (mean $\pm$ sample SD over five training seeds; 10,000 images per seed). The multiplier changes only λ_{pc} during joint sampling; starts and image-noise streams are paired. All arms use the same nominal image-update settings.

PC multiplier	Noise, $K = 200$	Noise, $K = 500$	Replay, $K = 500$
$\times 1$	16.9 ± 1.1	8.52 ± 1.47	$\mathbf{0.671 \pm 0.146}$
$\times 4$	$\mathbf{2.69 \pm 0.25}$	$\mathbf{2.76 \pm 0.45}$	1.520 ± 0.331
$\times 16$	78.8 ± 20.8	80.3 ± 20.4	3.669 ± 0.935
$\times 64$	87.1 ± 19.9	73.5 ± 18.0	3.858 ± 0.915
$\times 256$	45.4 ± 6.0	54.7 ± 3.6	54.85 ± 3.61

Continuous chains and exact hidden-state profile. The joint arm carries one hidden state through image updates. The profile arm analytically minimises the single-state quadratic energy at each image under the same frozen weights, image-update rate and noise scale; it does not carry a state. Neither arm is retuned for this comparison. Chains continue across all listed horizons without resetting images, states or random streams. The profile arm is a mechanism control, not an independently optimised model. Table 15 gives the test-reference FID and uncertainty for the milestones displayed in Figure 6b.

Paired quality and sampling time. At the nominal image-step budget, joint updates are compared with a state-preserving separate sampler that performs eight hidden-state updates per image update. Table 16 uses independent, interleaved, three-repeat batch-128 timing measurements on the same worker for every dataset and seed. Timing includes initialization, transfers and synchronized sampling, but excludes compilation and metric evaluation. The 10,000-image FIDs are from a separate evaluation, and batch-128 latency is not extrapolated to its production time. At the same image-step count, joint sampling has 1.45–$1.70\times$ the batch latency of corresponding models without internal PC.

Table 15: **Replay, noise and exact-state-profile trajectories.** Domain test FID (mean ± sample SD over five frozen PC-on checkpoints; 2,000 common-start images per seed and endpoint). The joint sampler carries h across image updates; the profile sampler minimises over h at every update. Same-K columns share weights and image-update settings.

Dataset	K	Joint replay	Joint noise	Profile noise
MNIST	0	1.121 ± 0.096	160.000 ± 0.104	160.000 ± 0.104
	500	0.743 ± 0.161	8.790 ± 0.913	31.928 ± 2.880
	10,000	3.309 ± 0.756	2.989 ± 0.904	9.086 ± 3.207
	20,000	4.274 ± 1.105	3.385 ± 0.982	8.750 ± 3.538
Fashion-MNIST	0	4.037 ± 0.182	85.324 ± 0.501	85.324 ± 0.501
	200	3.408 ± 0.147	35.909 ± 1.263	49.965 ± 3.377
	1,000	5.865 ± 1.402	5.295 ± 2.647	29.329 ± 5.393
	5,000	19.071 ± 6.899	15.519 ± 3.708	8.706 ± 2.287
	20,000	41.333 ± 11.067	42.018 ± 7.764	25.076 ± 6.390

Table 16: **Joint versus state-preserving separate sampling.** Domain test FID is mean ± sample SD across five seeds with 10,000 images per seed. Speedup is the five-seed geometric mean of paired separate-8/joint latency ratios for a compiled, synchronised batch of 128 at the same image-step count; compilation and metric evaluation are excluded. Both samplers use the same frozen PC-on weights and starts.

Dataset	Start	K	Joint FID	Separate-8 FID	Speedup
MNIST	Replay	500	0.673 ± 0.141	0.650 ± 0.122	$3.83\times$
	Noise	500	8.648 ± 1.439	10.626 ± 1.442	$3.92\times$
Fashion-MNIST	Replay	200	3.663 ± 0.258	3.664 ± 0.259	$3.16\times$
	Noise	200	36.261 ± 1.535	36.265 ± 1.534	$3.43\times$

Limits of component attribution. In a separate post-training probe, freezing the remaining random hidden state after head removal changes Fashion-MNIST nominal FID by less than 0.001 and lowers MNIST noise FID from 8.806 to 8.524; all weights were trained with the head present. A separately selected four-cell retraining comparison has MNIST MSE/on replay FID 54.033 ± 59.059 versus 1.464 ± 0.322 for MSE/off across all five seeds, but the selected recipes differ. These observations limit claims of component necessity or a universal internal-PC gain; they do not negate the post-training force response or the specified sampling cost comparison. Full per-seed results and protocols are retained in the experiment archive.

G THEORY-VERIFICATION DETAILS

These diagnostics probe finite-step gradient tracking and energy ordering in an unconditional MNIST model with an eight-output quadratic head and training seed 42. They describe the implemented energy and sampler; they do not establish global state optimality or exact stationary sampling.

Finite-reference gradient tracking. The fixed-input gradient probe uses 250 training images (25 per class). Latent states are inferred from a fresh initialisation for $\mathcal{T}$ plain-SGD steps at the training rate γ (reference $\mathcal{T}_{\mathrm{ref}}$=256); gradients $\nabla_{\boldsymbol{x}^{(0)}} E$ hold the vode states fixed, and the reported cosine is averaged per sample over the batch. Cosines between the $\mathcal{T}$-step gradient and the reference: $\mathcal{T}$=1:0.9984, 2:0.9984, 4:0.9984, 8:0.9985, 16:0.9986, 32:0.9988, 64:0.9992, 128:0.9997; the gradient at the joint sampler's terminal $(\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1:L)})$ agrees with a re-inferred reference at 0.9992.

Energy ordering and decomposition. Figure 10b compares real images, generated images, corrupted inputs and uniform noise (200 inputs per group), evaluating the total energy and output-head contribution at the same inferred states. Total energy orders real ≈ generated < corrupted < noise, while the output head is nearly flat across groups, identifying the measured contribution of prediction errors.

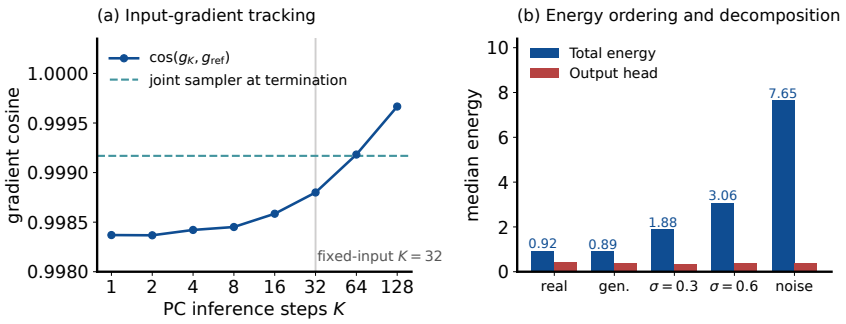

Figure 10: **Finite-state energy diagnostics** on one unconditional MNIST model (seed 42). **(a)** Input-gradient cosine after $\mathcal{T}$ PC updates against a $\mathcal{T} = 256$ reference; the dashed line marks the terminal sampler gradient. **(b)** Median total and output-head energies for 200 inputs per group. These probes assess finite-step behaviour.

Interpretation and scope. The cosine in Figure 10a measures directional agreement, not gradient-magnitude accuracy, global state optimality or chain stationarity. With one PC state, gradient truncation makes the PC term carry the implemented input gradient, but automatic differentiation still traverses the feed-forward prefix. These diagnostics therefore do not establish synapse-local learning or canonical top-down PC (Rao & Ballard, 1999; Bastos et al., 2012; Whittington & Bogacz, 2017).