Fast and Functional structured data generator

Alessandra Carbone, Aurélien Decelle, Lorenzo Rosset, Beatriz Seoane

Fast and Functional structured data generator

Alessandra Carbone [1] **Aurélien Decelle** [2 3] **Lorenzo Rosset** [2 3] **Beatriz Seoane** [2 3]

Abstract

In this study, we address the challenge of using energy-based models to produce high-quality, label-specific data in complex structured datasets. Traditional training methods encounter difficulties due to inefficient Markov chain Monte Carlo mixing, which affects the diversity of synthetic data and increases generation times. To address these issues, we use a novel training algorithm that exploits non-equilibrium MCMC effects. This approach improves the model's ability to correctly classify samples and generate high-quality samples in only a few sampling steps. The effectiveness of this method is demonstrated learning three datasets with Restricted Boltzmann Machines: handwritten digits for visualization, a human mutation genome dataset classified by continental origin, and sequences of an enzyme protein family categorized by experimental biological function.

1. Introduction

Energy-based models (EBMs) (Ackley et al., 1985; Smolensky, 1987; LeCun et al., 2006; Xie et al., 2016) are powerful generative models that encode the complex data set distribution into the Boltzmann distribution of a given energy function. Their simplest versions, the Boltzmann (Ackley et al., 1985) and the Restricted Boltzmann machine (Smolensky, 1987), have recently got renewed attention in the scientific world, not only because they can generate high-quality synthetic samples in datasets for which convolutional layers offer no appreciable advantage (Cocco et al., 2018; Yelmen et al., 2021; 2023), but also because they offer appealing modelling and interpretation capabilities for applications while requiring relatively small training sets. Indeed, the trained model can be understood and studied as a physical interaction system to model many-body distributions (Carleo & Troyer, 2017; Melko et al., 2019), infer physical interactions (Weigt et al., 2009; Morcos et al., 2011), extract patterns (Tubiana et al., 2019), or cluster (Decelle et al., 2023). The process of feature coding can also be analytically rationalized to some extent (Decelle & Furtlehner, 2021; Decelle et al., 2017).

EBMs, however, pose a major difficulty in training because the goodness of the trained models depends entirely on the quality of convergence to equilibrium of the Markov Chain Monte Carlo (MCMC) used to estimate the log-likelihood gradient during training (Decelle et al., 2021; Nijkamp et al., 2022). These concerns are especially critical when dealing with highly structured datasets, as sampling multimodal distributions is particularly costly. This is because mixing times increase rapidly during training, which is dominated by barriers between metastable states. Non-ergodic MCMC sampling often leads to models that overrepresent certain modes at the equilibrium distribution level (Nijkamp et al., 2020; Decelle et al., 2021). Moreover, even perfectly trained models can be very poor generators because they are unable to display all of the diversity encoded in the probability measure due to the inability of the chains to mix in a reasonable amount of time.

Recent work has shown that linking the Boltzmann distribution to the empirical distribution is indeed a nuisance that should be avoided if the ultimate goal is to generate samples. Instead, it is more efficient to train the model to fit the statistics of the dataset, not at the convergence of the MCMC process (as is common when training EBMs), but after a short and predetermined sampling process (Nijkamp et al., 2019; Decelle et al., 2021; Agoritsas et al., 2023). This means that EBMs can be trained to function as stable diffusion models (Sohl-Dickstein et al., 2015), i.e., fast and accurate generators that perform a set of decoding tasks impressed on the model during training. For structured datasets, this strategy offers two obvious improvements: The generated samples better reflect the diversity of the dataset, and one does not need an excessive number of MCMC steps to generate good-quality samples. Moreover, training out-of-equilibrium EBMs is not only faster than the standard procedure, but also more stable and easier to

Lorenzo Rosset is the main author contributing to this work. [1]Sorbonne Université, CNRS, IBPS, Laboratoire de Biologie Computationnelle et Quantitative - UMR 7238, 75005 Paris, France. [2]Departamento de Física Teórica, Universidad Complutense de Madrid, 28040 Madrid, Spain. [3]Université Paris-Saclay, CNRS, INRIA Tau team, LISN, 91190 Gif-sur-Yvette, France.. Correspondence to: Aurélien Decelle <adecelle@ucm.es>, Beatriz Seoane <beseoane@ucm.es>.

Accepted to ICML workshop on Structured Probabilistic Inference & Generative Modeling, 2023, Honolulu, Hawaii, USA.

control (Decelle et al., 2021).

In this paper, we show that Restricted Boltzmann Machines (RBMs) can be simultaneously trained to perform two different tasks after a few MCMC sweeps. First, they are able to generate samples conditioned on a particular label when initialized with random conditions. The samples generated by the model satisfy well the individual label statistics with high accuracy and cover the entire data space (Fig. 1). Second, they can accurately predict the label associated with a given sample. We validate our method on three different datasets: MNIST, primarily to illustrate the method, and two highly structured datasets - one listing human DNA mutations in individuals, and the other featuring sequences of a protein family. For these two complex cases, a high-quality generation is usually challenging, if not impossible.

The structure of this paper is as follows: We begin by introducing our EBM. This is followed by an explanation of the out-of-equilibrium training. We then discuss our results in detail, coupled with an analysis of the tests performed to assess the quality of the generated samples. The paper concludes with a summary of our results and conclusions.

2. Restricted Boltzmann Machine

Although RBMs have been around for a long time, they are largely used to describe aligned DNA/RNA or homologous protein sequence datasets (Tubiana et al., 2019; Bravi et al., 2021; Yelmen et al., 2021; 2023). There are two reasons for this. First, convolutional layers are unlikely to provide much advantage in this case, and most importantly, they do not require many training examples to provide reliable results. The latter is especially important when dealing with semi-supervised tasks, since the number of manually curated entries is usually very small compared to the number of sequences available in public databases. We will devote all our work to this type of tasks and machines.

2.1. Definition of the model

The RBM is a Markov random field with pairwise interactions defined on a bipartite graph of two noninteracting layers of variables: The visible variables $\boldsymbol{v} = \{v_i\}_{i=1,\dots,N_v}$ represent the data, while the hidden variables $\boldsymbol{h} = \{h_\mu\}_{\mu=1,\dots,N_h}$ form a latent representation of the data that models the effective interactions between the visible variables. The joint probability distribution of the visible and hidden variables is given by the Boltzmann distribution

$$p_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h}) = \tfrac{1}{Z_{\boldsymbol{\theta}}} e^{-E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h})} \text{ with } Z_{\boldsymbol{\theta}} = \textstyle\sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h})}. \quad (1)$$

In the previous expressions, the normalization factor $Z_{\boldsymbol{\theta}}$ is called the *partition function*, $\boldsymbol{\theta}$ refers to the parameters of the model and E is the energy function or Hamiltonian. In the simplest case, both the visible and the hidden units are binary variables, $v_i, h_\mu \in \{0,1\}$, but we will also consider categorical (namely Potts) variables for v_i in the case of the protein sequence dataset, see e.g. (Tubiana et al., 2019; Decelle et al., 2023) for a Potts version of the model. For the semi-supervised setting, we introduce an additional categorical variable in the visible layer, $\ell \in \{1,\dots,N_\ell\}$, that represents the label associated with the data point. That is, we follow the same scheme as in Ref. (Larochelle et al., 2012), but use a categorical variable for the label instead. The associated Hamiltonian is

$$E_{\boldsymbol{\theta}}(\boldsymbol{v},\boldsymbol{h},\ell) = -\sum_i a_i v_i - \sum_\mu b_\mu h_\mu - \sum_{i\mu} v_i w_{i\mu} h_\mu \\ - \textstyle\sum_m c_m \delta_{\ell,m} - \sum_{m\mu} \delta_{\ell,m} d_{m\mu} h_\mu, \quad (2)$$

where $\delta_{\ell,m}$ is the Kronecker symbol that returns 1 if the label has the value m and 0 otherwise, $\boldsymbol{a} = \{a_i\}$, $\boldsymbol{b} = \{b_\mu\}$ and $\boldsymbol{c} = \{c_m\}$ are three sets of local fields acting respectively on the visible and hidden layers and on the label variable. $\boldsymbol{w} = \{w_{i\mu}\}$ is the *weight matrix* that models the interactions between visible and hidden layers, and $\boldsymbol{d} = \{d_{m\mu}\}$ is the *label matrix* that represents the interactions between the label and the hidden layer. The structure of the semi-supervised RBM is sketched in Fig. 2-A.

2.2. Out-of-equilibrium training

EBMs are generally trained by maximizing the Log-Likelihood (LL) function of the model computed on the dataset $\mathcal{D} = \{(\boldsymbol{v}^{(1)},\ell^{(1)}),\dots,(\boldsymbol{v}^{(M)},\ell^{(M)})\}$

$$\mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) = \tfrac{1}{M}\textstyle\sum_{m=1}^{M} \log p_{\boldsymbol{\theta}}\left(\boldsymbol{v}=\boldsymbol{v}^{(m)}, \ell=\ell^{(m)}\right) \\ = \frac{1}{M}\sum_{m=1}^{M} \log \sum_{\boldsymbol{h}} e^{-E_{\boldsymbol{\theta}}\left(\boldsymbol{v}^{(m)},\boldsymbol{h},\ell^{(m)}\right)} - \log Z_{\boldsymbol{\theta}}, \quad (3)$$

via (stochastic) gradient ascent. As usual, the gradient of $\mathcal{L}$ is obtained by deriving it with respect to all parameters of the model (i.e., $\boldsymbol{\theta} = \{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\boldsymbol{d}\}$ in our RBMs), which can be written as a subtraction of two terms:

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \left\langle -\frac{\partial E_{\boldsymbol{\theta}}}{\partial \theta_i} \right\rangle_{\mathcal{D}} - \left\langle -\frac{\partial E_{\boldsymbol{\theta}}}{\partial \theta_i} \right\rangle_{E}. \quad (4)$$

The symbols $\langle\cdot\rangle_{\mathcal{D}}$, and $\langle\cdot\rangle_E$ represent the average over the dataset and the model's Boltzmann measure (1), respectively. One of the main challenges in training Energy-Based Models (EBMs) is computing a term on the right-hand side, usually estimated via MCMC simulations. This term requires the Markov chains to reach equilibrium—reflecting the Boltzmann measure—before statistical averages can be computed. This process can be very time-consuming, especially with complex datasets. The same issue comes up when generating new data samples according to the Boltzmann distribution. However, as mentioned in the introduction, there is a simple way around this problem (Nijkamp et al., 2019; Decelle et al., 2021; Agoritsas et al., 2023).

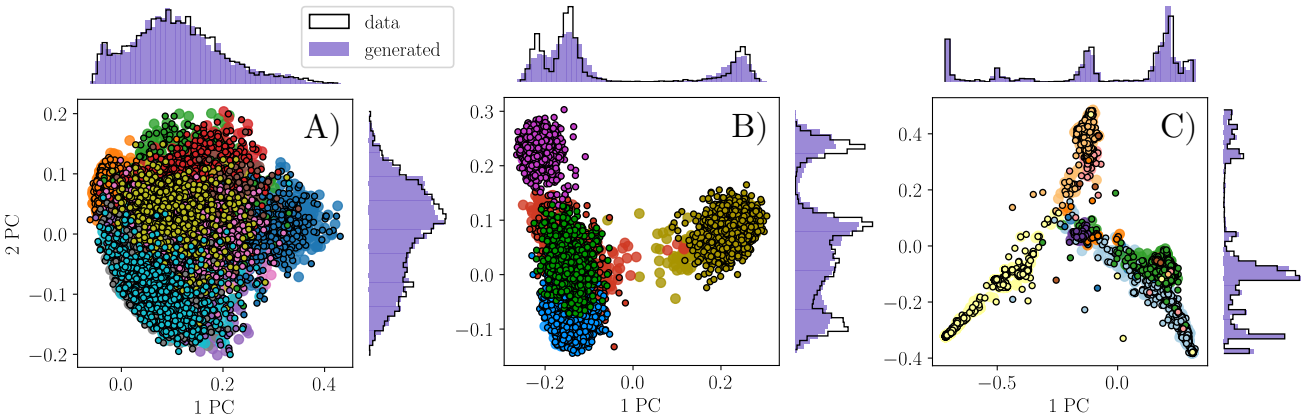

Figure 1. Conditioned generation F&F for the MNIST (A), HGD (B), and GH30 (C) datasets after 10 MCMC steps from a random initialization. The data are projected along the first two principal components of the dataset's PCA. The big dots correspond to true data and the small contoured dots are the generated samples, with different colors corresponding to the different labels.The synthetic dataset has the same structure as the real dataset, i.e. each category contains the same number of entries as the real dataset. In the outer panels, the histograms represent the distributions of the dataset (black outline) and the generated samples (violet-shaded area) when projected along each of the two principal directions used for the central scatter plot. The corresponding figure obtained with a standard PCD training is shown in Fig. 8 in Appendix D.

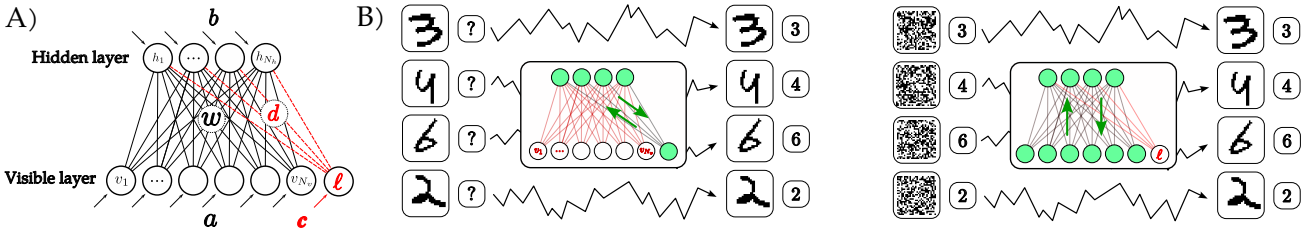

Figure 2. A): Scheme of the semi-supervised RBM. B): Sketch of the sampling procedures used to calculate the two gradients during training. Left): label prediction. The visible layer is clamped to the data, while the labels are initialized randomly. The hidden layer and labels are sampled alternately using block-Gibbs sampling and, after k MCMC steps, the model must provide the correct labels. Right): Conditional Sampling. The labels are fixed and the visible layer is initialized randomly. The model must generate a sample corresponding to the label in k MCMC steps.

The learning dynamics ruled by the gradient in Eq. 4 have a fixed point where the moments of the distribution match those of the dataset, signified by $\langle -\partial E_{\boldsymbol{\theta}}/\partial\theta_i\rangle_{\mathcal{D}} = \langle -\partial E_{\boldsymbol{\theta}}/\partial\theta_i\rangle_{E}$. This indicates that even with accurate gradient computation during training—which is often not achievable—generation with these models is costly. It involves equilibrating the MCMC chains prior to generating good quality samples. This becomes more challenging as the mixing times increase during training (Decelle et al., 2021; Dabelow & Ueda, 2022). An alternative approach suggests training the model to replicate the dataset's moments not at equilibrium, but after a few sampling steps k from an initial distribution $\boldsymbol{p}_0$. This can be achieved by adjusting the gradient as

$$\frac{\partial \mathcal{L}^{\mathrm{OOE}}}{\partial\theta_i} = \left\langle -\frac{\partial E_{\boldsymbol{\theta}}}{\partial\theta_i}\right\rangle_{\mathcal{D}} - \left\langle -\frac{\partial E_{\boldsymbol{\theta}}}{\partial\theta_i}\right\rangle_{p(k,\boldsymbol{p}_0)} . \tag{5}$$

Here, $p(k, \boldsymbol{p}_0)$ represents the non-stationary distribution of samples generated through an MCMC process that hasn't reached equilibrium. The model trained this way is optimized to generate quality samples when sampled following the exact same procedure (at the fixed point): same update rules, initialization distribution and number of steps. This possibility has been recently proven rigorously (Agoritsas et al., 2023), and experimentally validated in several studies across different EBMs (Nijkamp et al., 2019; 2020; Muntoni et al., 2021), including RBMs (Decelle et al., 2021).

In this paper we will go one step further. We want to train the RBM to perform not one but two different generative

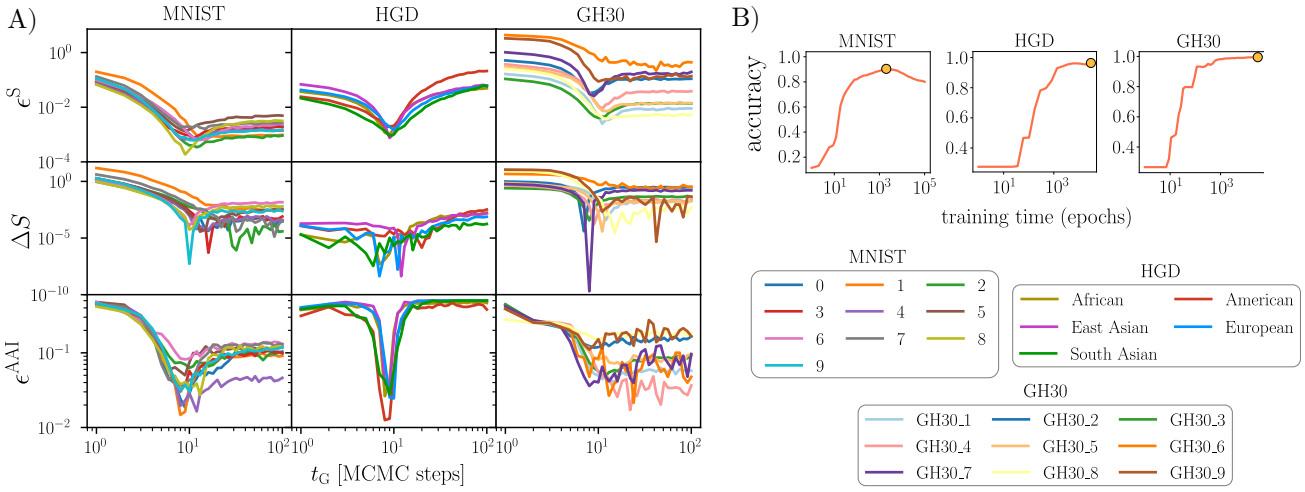

Figure 3. A): For each of the datasets considered, we show the evolution of three different quality scores as a function of sampling generation time, t_G, for each label separately. The first row shows the error on the eigenvalue spectra, the second row shows the error on the entropy, and the third row shows the Adversarial Accuracy Indicator. For the GH30 dataset only, we used the training set to generate the error curves because there was a too limited data in the testset to compare certain categories. The definition of the scores can be found in the Appendix C. B): Accuracy of the F&F RBM in the label prediction task as a function of training time.

tasks after only $k = 10$ MCMC sweeps by manipulating the chain initializations $\boldsymbol{p}_0$. Specifically, we want to train the model to both synthesize (from random) samples of a given a label, and to infer the correct label when given a dataset sample as chain initialization. To this end, we use two different out-of-equilibrium gradients in training, each designed for one of these tasks. The difference between the two is how we compute $\langle\cdot\rangle_{p(k,\boldsymbol{p}_0)}$ in (5). For label prediction, this term is computed by clamping the visible layer onto the images/sequences in the minibatch and letting evolve the label configurations. For conditional generation, this term is computed using chains where the visible layer is randomly initialized and the labels are kept fixed to the labels in the minibatch. A sketch of the sampling procedures used to compute the two gradients can be found in Fig. 2-B. We refer to the models trained in this way as F&F RBMs. The model and hyperparameters used for the training are listed in Tab. 2 of the Appendix B.

3. Results

We applied the F&F RBM to three labeled datasets. First, MNIST (LeCun et al., 1998), which comprises images of handwritten digits along with their respective labels, enabling us to visually evaluate conditional generation quality. Second, the Human Genome Dataset (HGD) (Consortium et al., 2015), comprising binary vectors representing a human individual with 1s or 0s indicating gene mutation relative to a reference sequence. Labels here signify the individual's continental origin. Lastly, the GH30 enzyme protein family dataset, a benchmark for the model's capability to generate artificial protein sequences having a particular biological function trained using natural sequences classified in the CAZy database (Lombard et al., 2014). Detailed explanations of these datasets are available in Appendix A.

Classification task – Fig.3-B illustrates the label prediction accuracy for the testset over training time. For MNIST, accuracy peaks at 0.9 after about 2000 epochs, then declines. In contrast, accuracy continually rises for both HGD and GH30, achieving 0.96 and 0.99, respectively, in the most trained models. The confusion matrices from label prediction for all datasets are gathered in Appendix D, Fig. 5.

Conditioned Generation task –We show in Fig. 1 a projection of the samples generated with a given label onto the first two principal directions of each dataset. The F&F model effectively generates data within a few MCMC steps (10 sweeps in our case) that satisfy the target labels and cover the entire data space following a very similar distribution to the original data, as can be seen from the comparison of the histograms in the figure. To further assess the generated data's quality, we used several error scores comparing synthetic and real data properties over the sampling time. These scores examine error in the covariance matrix spectrum, ϵ^{S}, diversity via an entropy measure, ΔS, and mode collapse and overfitting using the Adversarial Accuracy Indicator (Yale et al., 2020), ϵ^{AAI}. In all three cases the score is always positive and the perfect generation corresponds to an error of zero. Detailed definitions are found in Appendix C. As shown in Fig. 3-A, the best quality samples of each category are generated at about 10 steps, the same number of steps used for gradient estimation during training.

For comparison, we show in Fig. 7 of the Appendix D the scores obtained by the F&F RBM and the traditional RBM trained in semi-supervised mode with PCD on the three datasets. Interestingly, based on previous experience with models trained on these datasets without label monitoring, we found some unexpected results when we applied this analysis to the semi-supervised PCD-RBM. On the MNIST dataset, which normally yields well-trained PCD-RBM models (Decelle et al., 2021), we obtained machines with enormous thermalization times after only a few epochs of training. Conversely, even though the HGD is typically a very difficult benchmark dataset for classical equilibrium RBM models (Béreux et al., 2023), we found that semi-supervised training yielded very high-quality models with thermalization times of only a few hundred MCMC steps. Finally, we found that PCD-RBM completely fails in generating samples from the GH30 dataset, as the Markov chains immediately get stuck in wrong regions of the data space. In Fig. 8 in the Appendix D, we show a visualization of the results obtained by generating data using PCD-RBM models after no less than 10^5 MCMC steps for the three different datasets. These results show that classical training of RBMs with PCD is unreliable for conditional generation. In contrast, the F&F model proved to be robust and reliable for all the tested datasets and provided high-quality artificial samples after only a few MCMC steps.

For a more biologically relevant measure of generated protein sequences' quality, we extensively assessed their predicted three-dimensional structure using the `esm` tool (Lin et al., 2023), comparing these predictions with the test set. Specifically, we created histograms for both the generated sequences and the test set based on the frequency of predicted pLDDT scores from `esm`, indicating the average confidence in the folding. The generated set consists of 150 samples per each of the 9 categories. These distributions are displayed in Fig. 4 showing a remarkable agreement.

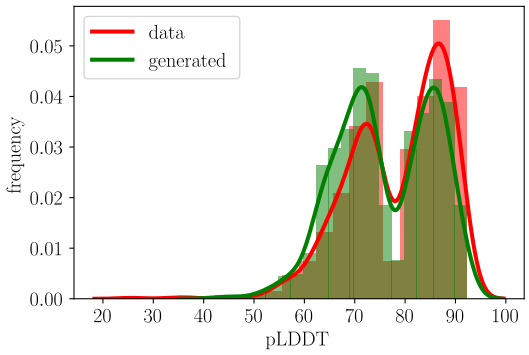

Figure 4. Histograms of frequencies of the pLDDT score for generated data (green) and real data (red). Given a reference protein structure and a structure predicted by a model, the LDDT (Local Distance Difference Test) score assesses how well local atomic interactions in the reference protein structure are reproduced in the prediction. The pLDDT (predicted LDDT) score is returned by the `esm` model, and it allows us to evaluate the degree of confidence of a folding even without having the reference structure.

4. Conclusions

In this study, we used a unique method for training RBMs to embed the statistics of the datasets into the nonstationary distributions of a Markov chain process (Nijkamp et al., 2019; Decelle et al., 2021; Agoritsas et al., 2023), in contrast to conventional methods that encode information only at the equilibrium measure level. This strategy allows us to use RBMs as efficient generators, similar to stable diffusion models, with the added benefit that various generative tasks can be easily encoded into the model. In particular, we trained RBMs to generate label-conditioned samples in a minimal number of sampling steps– a process that is typically tedious and slow in conventional methods (Larochelle et al., 2012)– and derive the good label when Markov chains are randomly initialized. We have shown that our approach successfully generates high-quality synthetic samples that accurately reflect the full diversity of the dataset even from highly structured data, overcoming the limitations of standard (equilibrium) training methods. Last but not least, the two-gradient method presented here can be easily implemented in more powerful EBMs to model other complex datasets.

Acknowledgements

We acknowledge financial support by the Comunidad de Madrid and the Complutense University of Madrid (UCM) through the Atracción de Talento programs (Refs. 2019-T1/TIC-13298 and 2019-T1/TIC-12776), the Banco Santander and the UCM (grant PR44/21-29937), and Ministerio de Economía y Competitividad, Agencia Estatal de Investigación and Fondo Europeo de Desarrollo Regional (Ref. PID2021-125506NA-I00).

References

Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. A learning algorithm for boltzmann machines. Cognitive science, 9 (1):147–169, 1985.

Agoritsas, E., Catania, G., Decelle, A., and Seoane, B. Explaining the effects of non-convergent sampling in the training of energy-based models. arXiv preprint arXiv:2301.09428, 2023.

Baronchelli, A., Caglioti, E., and Loreto, V. Measuring complexity with zippers. European journal of physics, 26(5):S69, 2005.

Bravi, B., Tubiana, J., Cocco, S., Monasson, R., Mora,

T., and Walczak, A. M. Rbm-mhc: a semi-supervised machine-learning method for sample-specific prediction of antigen presentation by hla-i alleles. Cell systems, 12 (2):195–202, 2021.

Béreux, N., Decelle, A., Furtlehner, C., and Seoane, B. Learning a restricted Boltzmann machine using biased Monte Carlo sampling. SciPost Phys., 14:032, 2023. doi: 10.21468/SciPostPhys.14.3.032. URL https://scipost.org/10.21468/SciPostPhys.14.3.032.

Cantarel, B. L., Coutinho, P. M., Rancurel, C., Bernard, T., Lombard, V., and Henrissat, B. The carbohydrate-active enzymes database (cazy): an expert resource for glycogenomics. Nucleic acids research, 37(suppl_1):D233–D238, 2009.

Carleo, G. and Troyer, M. Solving the quantum many-body problem with artificial neural networks. Science, 355 (6325):602–606, 2017.

Cocco, S., Feinauer, C., Figliuzzi, M., Monasson, R., and Weigt, M. Inverse statistical physics of protein sequences: a key issues review. Reports on Progress in Physics, 81 (3):032601, 2018. doi: 10.1088/1361-6633/aa9965.

Consortium, . G. P. et al. A global reference for human genetic variation. Nature, 526(7571):68, 2015. doi: 10.1038/nature15393.

Dabelow, L. and Ueda, M. Three learning stages and accuracy–efficiency tradeoff of restricted boltzmann machines. Nature communications, 13(1):5474, 2022.

Decelle, A. and Furtlehner, C. Restricted boltzmann machine: Recent advances and mean-field theory. Chinese Physics B, 30(4):040202, 2021. doi: 10.1088/1674-1056/abd160.

Decelle, A., Fissore, G., and Furtlehner, C. Spectral dynamics of learning in restricted boltzmann machines. EPL (Europhysics Letters), 119(6):60001, 2017. doi: 10.1209/0295-5075/119/60001.

Decelle, A., Furtlehner, C., and Seoane, B. Equilibrium and non-equilibrium regimes in the learning of restricted boltzmann machines. Advances in Neural Information Processing Systems, 34:5345–5359, 2021.

Decelle, A., Rosset, L., and Seoane, B. Unsupervised hierarchical clustering using the learning dynamics of rbms. arXiv preprint arXiv:2302.01851, 2023.

Edgar, R. C. Muscle: a multiple sequence alignment method with reduced time and space complexity. BMC bioinformatics, 5(1):1–19, 2004.

Larochelle, H., Mandel, M., Pascanu, R., and Bengio, Y. Learning algorithms for the classification restricted boltzmann machine. The Journal of Machine Learning Research, 13(1):643–669, 2012.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. A tutorial on energy-based learning. Predicting structured data, 1(0), 2006.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. Science, 379(6637): 1123–1130, 2023.

Lombard, V., Golaconda Ramulu, H., Drula, E., Coutinho, P. M., and Henrissat, B. The carbohydrate-active enzymes database (cazy) in 2013. Nucleic acids research, 42(D1): D490–D495, 2014.

Melko, R. G., Carleo, G., Carrasquilla, J., and Cirac, J. I. Restricted boltzmann machines in quantum physics. Nature Physics, 15(9):887–892, 2019.

Morcos, F., Pagnani, A., Lunt, B., Bertolino, A., Marks, D. S., Sander, C., Zecchina, R., Onuchic, J. N., Hwa, T., and Weigt, M. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. Proceedings of the National Academy of Sciences, 108(49):E1293–E1301, 2011. doi: 10.1073/pnas.1111471108.

Muntoni, A. P., Pagnani, A., Weigt, M., and Zamponi, F. adabmdca: adaptive boltzmann machine learning for biological sequences. BMC bioinformatics, 22(1):1–19, 2021.

Nijkamp, E., Hill, M., Zhu, S.-C., and Wu, Y. N. Learning non-convergent non-persistent short-run mcmc toward energy-based model. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/2bc8ae25856bc2a6a1333d1331a3b7a6-Paper.pdf.

Nijkamp, E., Hill, M., Han, T., Zhu, S.-C., and Wu, Y. N. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. Proceedings of the AAAI Conference on Artificial Intelligence, 34

(04):5272–5280, Apr. 2020. doi: 10.1609/aaai.v34i04.5973. URL https://ojs.aaai.org/index.php/AAAI/article/view/5973.

Nijkamp, E., Gao, R., Sountsov, P., Vasudevan, S., Pang, B., Zhu, S.-C., and Wu, Y. N. Mcmc should mix: learning energy-based model with neural transport latent space mcmc. In International Conference on Learning Representations (ICLR 2022)., 2022.

Smolensky, P. Information Processing in Dynamical Systems: Foundations of Harmony Theory, volume 6. 1987. ISBN 9780262291408.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In International Conference on Machine Learning, pp. 2256–2265. PMLR, 2015.

Tubiana, J., Cocco, S., and Monasson, R. Learning protein constitutive motifs from sequence data. Elife, 8:e39397, 2019. doi: 0.7554/eLife.39397.

Weigt, M., White, R. A., Szurmant, H., Hoch, J. A., and Hwa, T. Identification of direct residue contacts in protein–protein interaction by message passing. Proceedings of the National Academy of Sciences, 106(1):67–72, 2009. doi: 10.1073/pnas.0805923106.

Xie, J., Lu, Y., Zhu, S.-C., and Wu, Y. A theory of generative convnet. In International Conference on Machine Learning, pp. 2635–2644. PMLR, 2016.

Yale, A., Dash, S., Dutta, R., Guyon, I., Pavao, A., and Bennett, K. P. Generation and evaluation of privacy preserving synthetic health data. Neurocomputing, 416:244–255, 2020.

Yelmen, B., Decelle, A., Ongaro, L., Marnetto, D., Tallec, C., Montinaro, F., Furtlehner, C., Pagani, L., and Jay, F. Creating artificial human genomes using generative neural networks. PLoS genetics, 17(2):e1009303, 2021.

Yelmen, B., Decelle, A., Boulos, L. L., Szatkownik, A., Furtlehner, C., Charpiat, G., and Jay, F. Deep convolutional and conditional neural networks for large-scale genomic data generation. bioRxiv, pp. 2023–03, 2023.

A. Dataset description

A.1. MNIST dataset

The MNIST dataset (LeCun et al., 1998) consists of 28×28 grayscale images of handwritten digits tagged with a label indicating the digit represented, from 0 to 9. We first extracted a training set and a test set of respectively 10000 and 2000 images, and we then binarized the data by setting each pixel to 1 if the normalized value was above 0.3, and to 0 otherwise. To be fed to the RBM, the images have to be flattened into 784-dimensional binary vectors.

A.2. Human Genome Dataset (HGD)

The Human Genome dataset (HGD) (Consortium et al., 2015) represents the human genetic variations of a population of 5008 individuals sampled from 26 populations in Africa, East Asia, South Asia, Europe, and the Americas. Each sample is a sequence of 805 binary variables, $v_i \in \{0, 1\}$, representing the change alteration or not of a gene relative to a reference genetic sequence. Sequences are classified based on the continental origin of individuals. We trained the RBM on 4507 samples and retained 501 samples for the test set.

A.3. GH30 family

The glycoside hydrolases (EC 3.2.1.-), GH for short, are a family of enzymes that hydrolyze the glycosidic bond between two or more carbohydrates or between a carbohydrate and a non-carbohydrate moiety. GH30 is one of the GH families that has been divided into subfamilies in CAZy. It includes nine different subfamilies (GH30-1,..., GH30-9) corresponding to 11 different enzymatic chemical reactions. We created a training and test set of respectively 3922 and 975 annotated sequences from CAZy (Lombard et al., 2014; Cantarel et al., 2009), having care of reproducing the same samples-per-label proportion between training and test sets. The sequences were previously aligned in an MSA matrix using the MUSCLE algorithm (Edgar, 2004) with default parameters. We then cleaned all MSA columns in which the proportion of gaps was above 70% of the entries. The resulting sequences have a length of $N_v = 430$.

The details about the composition of the training/testing sets used for each dataset can be found in Table 1.

MNIST

Label	0	1	2	3	4	5	6	7	8	9
Train set count	1022	1078	1046	1031	965	916	972	1042	977	951
Test set count	188	224	218	191	220	174	208	178	197	202

HGD

Label	African	American	East Asian	European	South Asian
Train set count	1184	622	912	910	879
Test set count	138	72	96	96	99

GH30

Label	GH30_1	GH30_2	GH30_3	GH30_4	GH30_5	GH30_6	GH30_7	GH30_8	GH30_9
Train set count	886	287	1044	270	435	39	89	810	62
Test set count	221	71	260	67	108	9	22	202	15

Table 1. Number of data samples for each category in the train and test sets for the used datasets.

B. RBM training details

The hyperparameters used for the training processes discussed in this paper are given in Table 2.

dataset	epochs	minibatch size	total gradient updates	k	learning rate	N_{h}
MNIST (PCD)	30000	500	$6 \cdot 10^5$	100	10^{-2}	1024
HGD (PCD)	30000	4507	$3 \cdot 10^4$	100	10^{-2}	1024
GH30 (PCD)	30000	1961	$6 \cdot 10^4$	100	10^{-2}	1024
MNIST (F&F)	30000	500	$6 \cdot 10^5$	10	10^{-2}	1024
HGD (F&F)	30000	4507	$3 \cdot 10^4$	10	10^{-2}	1024
GH30 (F&F)	30000	1961	$6 \cdot 10^4$	10	10^{-2}	1024

Table 2. Hyper-parameters of the RBMs used in this work.

C. Quality scores

To assess the generation capabilities of the RBM, one can compute a set of observables on the generated dataset and the actual data and compare them (Decelle et al., 2021). In the plots of Figs. 3-A and 7 we have considered the following scores:

- **Error on the spectrum ($\epsilon^{\mathbf{S}}$)**: Given a data matrix $X \in \mathbb{R}^{M \times N_v}$, its singular value decomposition (SVD) consists in writing X as the matrix product

$$X = USV^T,$$

where $U \in \mathbb{R}^{M \times M}$, S is an $M \times N_v$ matrix with the singular values of X in the diagonal, and $V \in \mathbb{R}^{N_v \times N_v}$. Let us call $N_s = \min(M, N_v)$. Once we sort the singular values $\{s_i\}$ such that $s_1 > s_2 > \cdots > s_{N_s}$, we can define the error of the spectrum as

$$\epsilon^S = \frac{1}{N_s} \sum_{i=1}^{N_s} \left(s_i^{\mathrm{data}} - s_i^{\mathrm{gen}}\right)^2, \tag{6}$$

where $\{s_i^{\mathrm{data}}\}$ are the singular values of the true data and $\{s_i^{\mathrm{gen}}\}$ are the singular values of the generated dataset.

- **Error on the entropy ($\mathbf{\Delta S}$)**: We approximate the entropy of a given dataset by its byte size when compressed with gzip (Baronchelli et al., 2005). In particular, if S^{data} is the estimated entropy of the true data and S^{gen} is the estimated entropy of the generated data, we define the error of entropy as

$$\Delta S = \left(\frac{S^{\mathrm{gen}}}{S^{\mathrm{data}}} - 1\right)^2. \tag{7}$$

A large ΔS indicates that the generated set lacks diversity or that the generated samples are less "ordered" than the dataset.

- **Error on the Adversarial Accuracy Indicator (ϵ^{AAI})**: This score was introduced in Ref. (Yale et al., 2020) to quantify the similarity and "privacy" of data drawn from a generative model with respect to the training set. We first construct a dataset obtained by joining the real dataset with the generated dataset, and then compute the matrix of distances between each pair of data points. We denote by P_{GG} the probability that a generated datapoint has as the nearest neighbour a generated data and by P_{DD} the probability that a true datapoint has as the nearest neighbour a true data. In the best case, when the generated data are statistically indistinguishable from the true ones, we have $P_{\mathrm{GG}} = P_{\mathrm{DD}} = 0.5$. Therefore, we can define the error of the Adversarial Accuracy Indicator as follows:

$$\epsilon^{\mathrm{AAI}} = \frac{1}{2}\left[(P_{\mathrm{GG}} - 0.5)^2 + (P_{\mathrm{DD}} - 0.5)^2\right]. \tag{8}$$

D. Supplementary figures

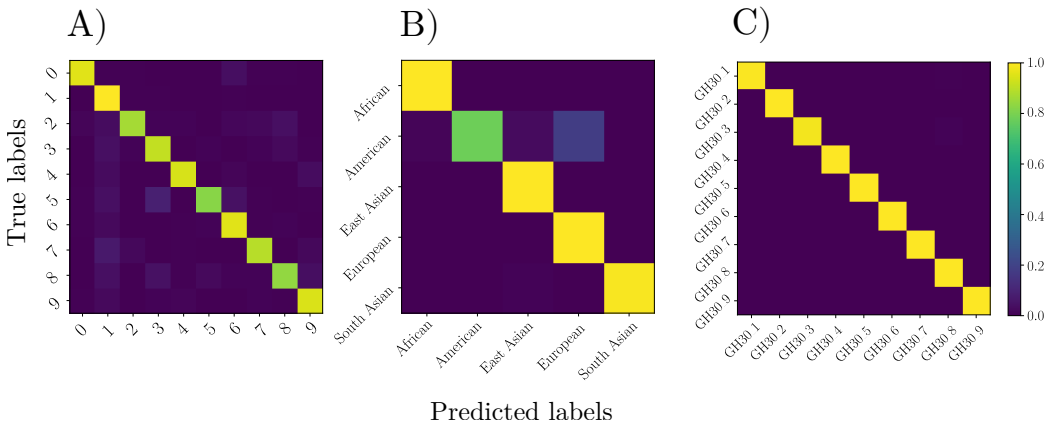

Figure 5. Confusion matrices for the label classification using F&F on the test sets of A) MNIST, B) HGD and C) GH30.

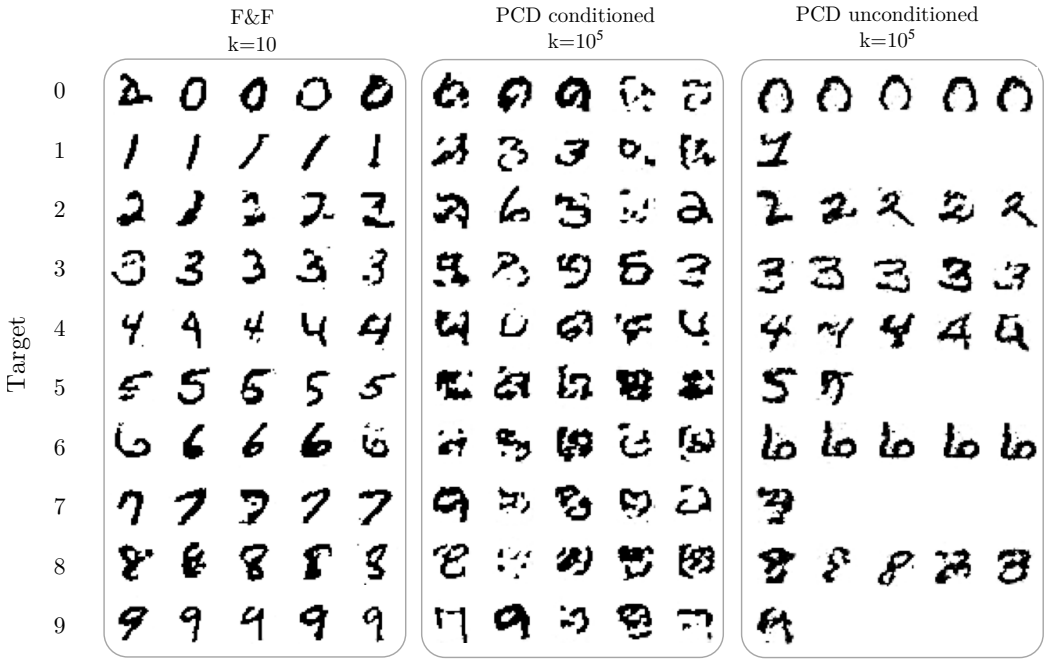

Figure 6. MNIST images created using different methods for specific labels. From left to right, the first box shows the output of F&F for $k = 10$ MCMC steps. The images in the second box are generated using a PCD-RBM after 10^5 MCMC steps when the Markov chains are clamped to a specific label value. The third box shows the result of sampling with a PCD-RBM, where we also sample the labels when running the Markov chains. An empty slot means that the RBM never provided the appropriate sample in our tests after 10^5 MCMC steps.

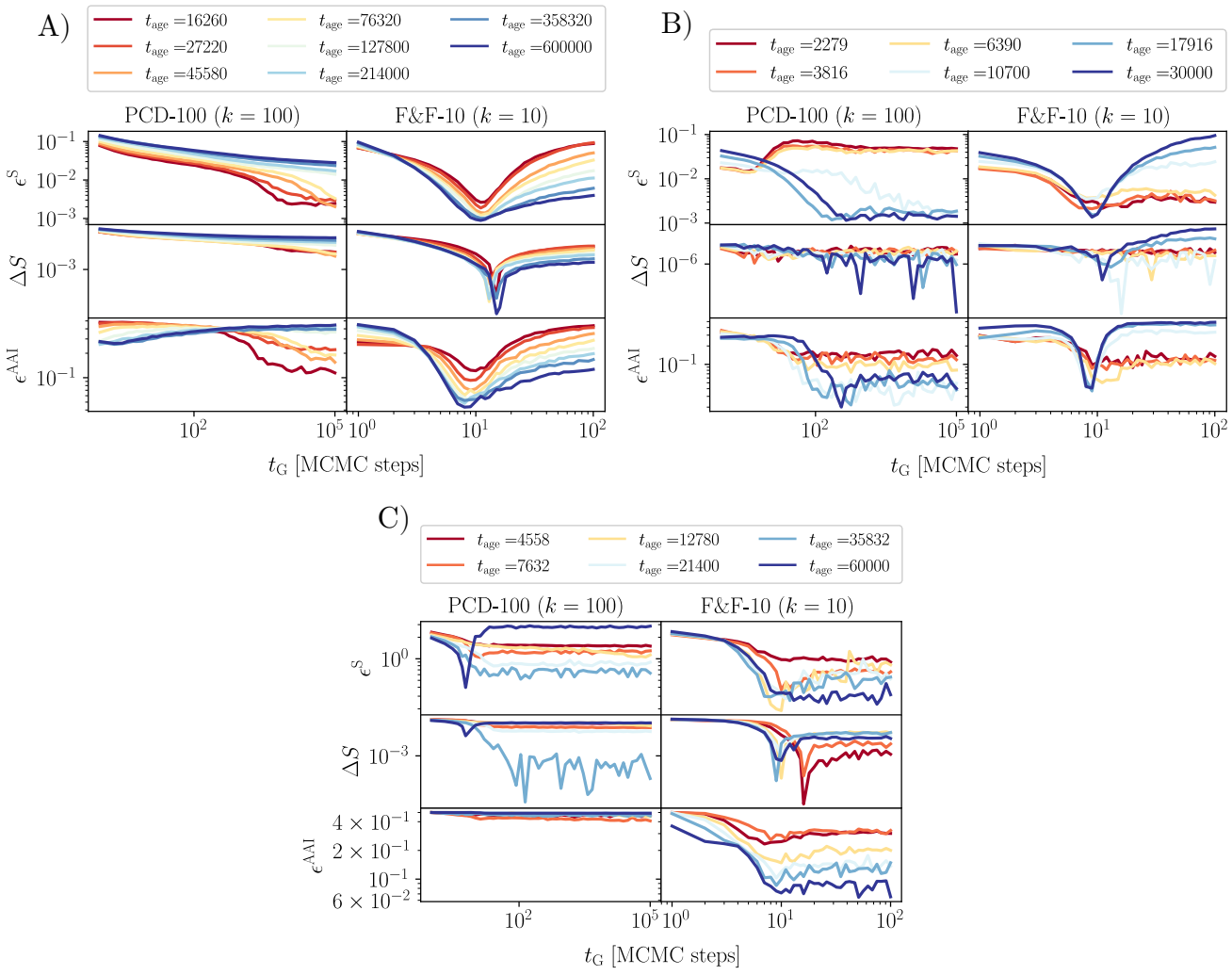

Figure 7. Comparison of the scores on the generated data between PCD and F&F RBMs as a function of the generation time for A) MNIST, B) HGD and C) GH30. All the scores are computed by comparing the test set with an identical (in terms of samples for each category) generated dataset. The samples of each category of the dataset have been compared with the corresponding samples of the synthetic data, and the curves shown in the figure represent the average scores across the different categories. The different colours of the curves represent different training times (t_{age}), expressed in terms of gradient updates. Notice that for the PCD-RBM the generation time ranges up to 10^5 MCMC updates, while for the F&F-RBM it only reaches 10^2 MCMC updates.

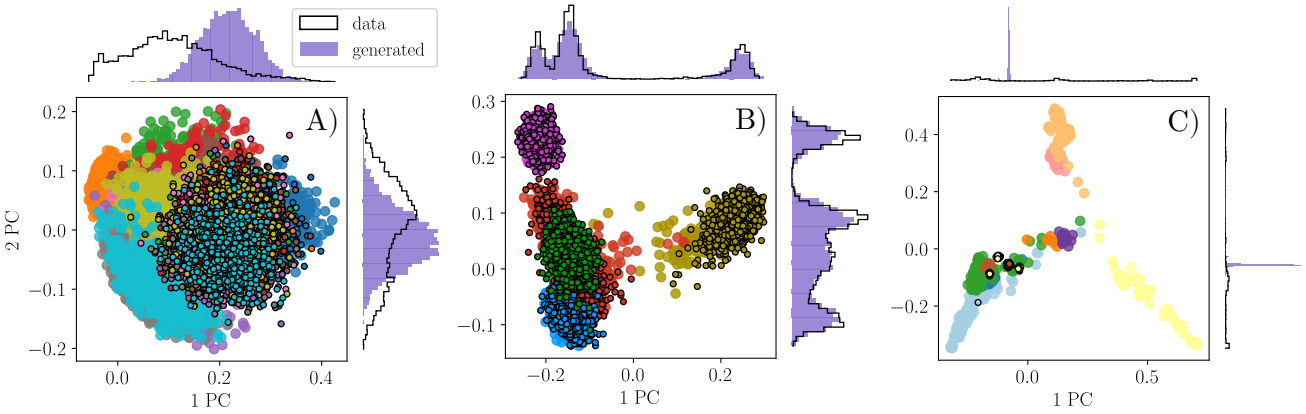

Figure 8. Conditioned generation using PCD for MNIST (A), HGD (B) and GH30 (C) datasets after 10^5 MCMC steps. The data are projected along the first two principal components of the dataset's PCA. The big dots correspond to true data and the small contoured dots are the generated samples, where different colours correspond to different labels. The synthetic dataset has the same structure as the true one, meaning that each category contains the same number of data as the true dataset. On the sides of the PCA, the histograms represent the distributions of the data (black contour) and the generated samples (violet-shaded area) when projected along the first two principal directions.