Semester project: On the difficulty of training MCMC-Based Maximum Likelihood Learning of Energy-Based Models

Matteo Bunino
EURECOM
matteo.bunino@eurecom.fr

Abstract

Energy based models are a novel method for parametric unnormalized density estimation, able to overcome the structural limitations of state of the art models, such as VAEs, and giving stronger theoretical guarantees compared to GANs. This topic is currently an active field of research and the literature is scarce, making it hard for a newbie to explore this word without falling in pitfalls or taking exaggeratedly long paths to reach a good solution. The aim of this semester project is to examine these models, with the aim of discovering general principles and drafting a recipe on how to properly train an EBM with Maximum Likelihood estimation, thoroughly sticking to the mathematical background. In this report, I will at first present the intrinsic complexity of training such models, then I will compare my findings to the current state of the art literature and eventually provide some intuitions that helped me to train an EBM with ML estimation.

1. Introduction

Energy based models are a novel method for parametric unnormalized density estimation, able to overcome the structural limitations of state of the art models, such as VAEs, and giving stronger theoretical guarantees compared to GANs. However, EBM with Maximum Likelihood estimation are particularly difficult to train due to the intractable normalizing constant Z_θ. For this reason the more promising and more studied alternative of Score Matching, [7], is usually preferred.
This semester project has the aim to explore the less popular EBM with ML estimation, trying to find a way to improve the current state of the art result while keeping strong theoretical bases.
An important survey on this family of EBM is [8], while further works are [4], [5] and [2].
In this work I explore and compare the performances of SGLD and SGHMC sampling methods, trying to improve the sampling performances, which represent a bottleneck of the entire training process.
The main topics of this project are:

- A descriptive introduction of EBM training with ML estimation, giving a detailed explanation of the dynamics of training.
- An overview of the most critical pitfalls that may occur during trainig, with a suggested list of general principles to follow when trainig an EBM.
- The implementation of a working model which is not employing informative initialization of the Markov chains and the exact Langevin sampling (SGLD).
- The proposal of an activation function able to improve generation performances and speedup convergence.

2. Energy Based Models

Training an Energy Based Model means fitting the unnormalized underlying negative log-density of data examples with a parametric function

$$E_\theta(\mathbf{x}) \propto -\log p_\theta(\mathbf{x}) \tag{1}$$

This may entail many advantages with respect to other methods, above all a very low bias in approximating the real distribution. This also removes the burden of the *a priori* selection of the density's structure, which usually has to be made by the experimenter.
The density given by an EBM is

$$p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z_\theta} \tag{2}$$

where $E_\theta(\mathbf{x})$ is a real valued function parametrized by θ and Z_θ is a normalizing constant:

$$Z_\theta = \int \exp(-E_\theta(\mathbf{x}))\, d\mathbf{x} \tag{3}$$

Classical alternatives to this approach are autoregressive models and variational inference.

Autoregressive models are tractable models in which it is assumed that an example depends on the previous ones. Given a dataset $\mathcal{X} = \{x_0, \dots, x_N\}$, the likelihood of this dataset can be written as:

$$p(\mathcal{X}) = \prod_{i=1}^{N} p_\theta(x_i | x_0, \dots, x_{i-1}) \tag{4}$$

where $p_\theta(x_i|x_0, \dots, x_{i-1})$ is a parametric function to learn. In variational inference, the target density $p(\mathcal{X})$ is approximated by a density function chosen from a predefined distributions family $q_\theta(\mathcal{X})$, which has to be *guessed* by the experimenter. Subsequently it is minimized the KL divergence

$$KL[q_\theta(\mathcal{X}) \,||\, p(\mathcal{X})] \tag{5}$$

In the setting of generative models, we assume the data to be generated from a latent variable $\mathbf{z}$, resulting in the conditioning $\mathbf{x}|\mathbf{z}$. This allows to define the latent variable model

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}) \tag{6}$$

We are interested in the posterior distribution

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z})\, p(\mathbf{z})}{\int p(\mathbf{x}, \mathbf{z})\, d\mathbf{z}} \tag{7}$$

which is intractable.

Variational inference allows to approximate the intractable posterior with another density $q_\theta(\mathbf{z}|\mathbf{x})$, where θ are called variational parameters that have to be optimized minimizing the KL divergence

$$\begin{aligned} &KL[q_\theta(\mathbf{z}|\mathbf{x}) \,||\, p(\mathbf{z}|\mathbf{x})] = \\ &= -\mathbb{E}_{z\sim q_\theta}[\log q_\theta(\mathbf{x}|\mathbf{z})] \\ &\quad + KL[q_\theta(\mathbf{z}|\mathbf{x}) \,||\, p(\mathbf{z})] \\ &\quad + \log p(\mathbf{x}) \end{aligned} \tag{8}$$

The term $p(\mathbf{x})$ is the marginal likelihood and sometimes is called *evidence*. It can be proven that

$$\log p(\mathbf{x}) \geq \mathbb{E}_{z\sim q_\theta}[\log q_\theta(\mathbf{x}|\mathbf{z})] - KL[q_\theta(\mathbf{z}|\mathbf{x}) \,||\, p(\mathbf{z})] \tag{9}$$

therefore defining a tractable lower bound (called ”Evidence Lower Bound”) to the log-likelihood. Maximizing this lower bound with respect to θ is equivalent to simultaneously maximizing the likelihood $p(\mathbf{x})$ and minimizing the KL divergence of Equation 8.

In variational inference, a partial solution to the potential lack of flexibility of the parametric distribution can be given by normalizing flows. In normalizing flows the parametric likelihood is built starting from a simple density $p_Z(\mathbf{z}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and transforming it into an arbitrary complex distribution. This is carried out by choosing a parametric and invertible function $\mathbf{f}_{\mathbf{i},\theta}(\mathbf{x})$ such that:

$$p_\theta(\mathbf{x}) = p_Z(\mathbf{f}_{\mathbf{i},\theta}{}^{-1}(\mathbf{x}))|\det(\nabla_{\mathbf{x}} \mathbf{f}_{\mathbf{i},\theta}{}^{-1}(\mathbf{x}))^T)| \tag{10}$$

The term ”flow” derives from the fact that Equation 10 can be extended by composing an arbitrary number of different transformations $\mathbf{f}_{\mathbf{i},\theta}$.

However this approach has the drawback of computing determinants that reduce the scalability potential.

2.1. Maximum Likelihood EBM

In the maximum likelihood framework, we are interested in maximizing

$$\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log p_\theta(\mathbf{x})] \tag{11}$$

From information theory we can extend this well known formula as:

$$\begin{aligned} -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log p_\theta(\mathbf{x})] &= H(p_{\text{data}}, p_\theta) && \text{(12a)} \\ &= KL[p_{\text{data}} \,||\, p_\theta] + H(p_{\text{data}}) && \text{(12b)} \end{aligned}$$

where $H(p_{\text{data}}, p_\theta)$ is the cross-entropy and $H(p_{\text{data}})$ is the entropy.

We cannot directly compute the likelihood of an EBM as in the maximum likelihood approach due to the intractable normalizing constant Z_θ. Nevertheless, we can still estimate the gradient of the log-likelihood with MCMC sampling, allowing for likelihood maximization with gradient ascent.

$$\nabla_\theta \log p_\theta(\mathbf{x}) = -\nabla_\theta E_\theta(\mathbf{x}) - \nabla_\theta \log Z_\theta \tag{13}$$

The second term is intractable to be computed exactly, but can be estimated through Monte Carlo sampling

$$-\nabla_\theta \log Z_\theta \simeq \nabla_\theta E_\theta(\tilde{\mathbf{x}}) \tag{14}$$

where $\tilde{\mathbf{x}} \sim p_\theta(\mathbf{x})$ is a random sample from the model.

Sampling from the model is not trivial and it is the most time consuming part of the training of this family of models. However, this can be achieved through SGLD and SGHMC methods that leverage the fact that $\nabla_{\mathbf{x}} \log Z_\theta = 0$, hence:

$$\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) = -\nabla_{\mathbf{x}} E_\theta(\mathbf{x}) \tag{15}$$

where $\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})$ is called *score*.

2.1.1 SGLD

Stochastic Gradient Langevin Dynamics, often referred to as *first-order Langevin dynamics* or simply Langevin dynamics, is a MCMC sampling method that brings superior convergence performances with respect to random walks like pure Metropolis-Hastings due to the additional gradient information. In particular, Langevin dynamics are a class of stochastic processes that can be used to sample from a large variety of distributions and are defined as:

$$d\mathbf{x}(t) = \mathbf{r}(\mathbf{x}(t))dt + \sqrt{2}d\mathbf{w}(t) \tag{16}$$

used to determine the evolution of the stochastic process $\mathbf{x}(t)$.

Algorithm 1 Learning of Energy Based model

```
procedure TRAINING($\epsilon, N$)
    $\theta_0 = p_{noise}(\theta)$                                        ▷ Initialize the model
    loop $n = 1, \dots N$                         ▷ External loop to update the parameters $\theta_n$
        $\mathbf{x}^+ \sim p_{\mathcal{D}}(\mathbf{x})$                              ▷ Sample data points from $\mathcal{D}$
        $\mathbf{x}^- \sim p_{\theta_{n-1}}(\mathbf{x})$                                 ▷ Sample from the model
        $\mathcal{C}^*_{ost} = E_{\theta_{n-1}}(\mathbf{x}^+) - E_{\theta_{n-1}}(\mathbf{x}_l^-)$   ▷ Compute the energy difference: Contrastive Divergence loss
        $\theta_n = \theta_{n-1} - \epsilon \nabla_\theta \mathcal{C}^*_{ost}$                     ▷ Update the model parameters
```

- $\mathbf{r}(\cdot) : \mathbb{R}^N \to \mathbb{R}^N$ is the driving force.

- $\mathbf{dw}(t) \sim \mathcal{N}(\mathbf{0}, dt\mathbf{I})$.

We can use the Fokker-Planck equation to study the stationary distribution $\rho_{ss}(\mathbf{x})$ of the stochastic process:

$$\mathrm{Tr}\{\nabla_{\mathbf{x}}[-\mathbf{r}(\mathbf{x})^T \rho_{ss}(\mathbf{x}) + \nabla_{\mathbf{x}}^T(\rho_{ss}(\mathbf{x}))]\} = 0 \qquad (17)$$

When choosing $\mathbf{r}(\mathbf{x}) = \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})$ (the *score*):

$$\rho_{ss}(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z_\theta} \qquad (18)$$

This method is employed to sample from the EBM and it is presented in details in Algorithm 2.

2.1.2 SGHMC

A valid alternative to Langevin dynamics presented in the previous paragraph is the Stochastic Gradient Hamiltonian Monte Carlo sampling method introduced by [1]. It is possible to show that SGLD and SGHMC are related in the same way as SGD and SGD with momentum are.

SGHMC comes from the necessity of making Hamiltonian Monte Carlo sampling scalable to large datasets, where it is not possible to compute the gradient on the entire dataset all at once. However, stochastic gradient is a source of unwanted stochasticity that hinders the convergence to the true stationary distribution. To solve this issue, [1] introduce a *friction* term that gives asymptotic guarantees of convergence.

Again, it is possible to find a link among SGHMC and Langevin dynamics considering the latter a special case of the former when the friction is large. For this reason, this method is also called *second-order Langevin dynamics*. However, as anticipated before, SGHMC has slightly better convergence properties due to the momentum term. A detailed description of this method can be found in Algorithm 3.

Usually matrix M is taken as the identity matrix. In my experiments I sought for more flexibility defining it as $M = m\mathbf{I}$, thus tuning the hyperparameter m.

2.2. Comparison EBM v. VAEs

Both Variational Autoencoders and EBM learn the parameters by maximizing the (marginal) log-likelihood, which can be interpreted also as the minimization of $KL[p_{\text{data}} \,||\, p_\theta]$.

VAEs are intrinsically latent variable models imposing an information bottleneck and approximating the posterior on the latent variables $p(\mathbf{z}|\mathbf{x})$ through variational inference, whereas EBMs generally are not. However, they can easily extended to latent variable models, like in the work of [9].

In VAEs, both encoder and decoder are implemented as DNN, modeling $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z})$ respectively, which are both Gaussian. Their parameters are optimized by maximizing the variational lower bound:

$$\mathbb{E}_{\mathbf{z}}[\log p_\theta(\mathbf{x}|\mathbf{z})] - KL[q_\phi(\mathbf{z}|\mathbf{x}) \,||\, p_\theta(\mathbf{z})] \qquad (19)$$

where $p_\theta(\mathbf{z})$ is a standard Gaussian prior on the latent space. The second term has a crucial regularization effect that makes it possible to generate new samples from random samples from the latent space. In particular, this term is forcing the distributions returned by the encoder to be close to the standard Gaussian, preventing them from being punctual or being too far apart from each other.

Effect of regularization:

- Continuity: two close points in the latent space should not give two completely different contents once decoded.

- Completeness: a point sampled from the latent space should give ”meaningful” content once decoded.

However, in the context of image generation, VAEs have the well-known drawback of producing noisy or blurred images, which usually make them fall behind GANs or other generative methods.

3. Two axes of difficulty

This specific family of Energy Based Models with Maximum Likelihood estimation presents a quite sensitive structure, presented in Algorithm 1, where we have an outer loop devoted to parameters update, optimizing the Contrastive

Algorithm 2 SGLD sampling from the model

```
procedure SAMPLE_SGLD($\eta, L$)
    $\mathbf{x}^- \sim p_{noise}(\mathbf{x})$    ▷ Initialize the generated data randomly. Usually standard Gaussian.
    loop $l = 1, \dots L$    ▷ Internal loop to collect samples $x^- \sim p_{\theta_{n-1}}(\mathbf{x})$
        $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$    ▷ Sample noise
        $\mathbf{x}_l^- = \mathbf{x}_{l-1}^- - \eta \nabla_{\mathbf{x}} E_{\theta_{n-1}}(\mathbf{x}_{l-1}^-) + \sqrt{2\eta}\mathbf{w}$    ▷ Perform one step of Langevin Dynamics
```

Algorithm 3 SGHMC sampling from the model

```
procedure SAMPLE_SGHMC($\eta, L, C, M$)
    $\mathbf{r} \sim \mathcal{N}(\mathbf{0}, M)$    ▷ Initialize momentum randomly
    $\mathbf{x}^- \sim p_{noise}(\mathbf{x})$    ▷ Initialize the generated data randomly. Usually standard Gaussian.
    loop $l = 1, \dots L$    ▷ Internal loop to collect samples $x^- \sim p_{\theta_{n-1}}(\mathbf{x})$
        $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$    ▷ Sample noise
        $\mathbf{x}_l^- = \mathbf{x}_{l-1}^- + \eta M^{-1}\mathbf{r}_{l-1}$    ▷ Update generated samples
        $\mathbf{r}_l = \mathbf{r}_{l-1} - \eta \nabla_{\mathbf{x}} E_{\theta_{n-1}}(\mathbf{x}_{l-1}^-) - \eta C M^{-1}\mathbf{r}_{l-1} + \sqrt{2\eta C}\mathbf{w}$    ▷ Update momentum
```

Divergence loss with respect to θ, and an inner loop represented by the sampling from the model.

Indeed these two loops are strongly correlated and this represent a pathological defect that makes it difficult to properly decorrelate the effect of each actor on the outcome of training. This structure can be the cause of some unstable behaviors in which the divergence of the sampling results in the divergence of the parameters optimization.

This architecture is unique because we employ the same function (which is a DNN) to work on both true and generated samples, differently from VAEs and GANs.

3.1. Model optimization

The effect of CD loss is to update the parameters θ in order to reduce the energy in the regions close to real data points, while increasing the energy around the generated points. This can be noticed at the beginning of training when the energy of generated images increases whereas we have an opposite trend for the energy of true images, which becomes negative.

After some iterations, if the training is well set, the energy of the generated images begins to decrease until reaching negative values. Regardless the sign of the energies, the model has converged when the energy of real and generated samples reach asymptotic values which are close each other.

Energy landscape

Recalling Algorithms 2 and 3, sampling is driven by the energy gradient and its effectiveness improves as the gradient increases, prevailing over the random noise. This means that upon convergence, the landscape of the energy can be imagined as steep valleys surrounded by high mountains. In this scenario sampling is intuitively represented by throwing a ball that bouncing on this landscape will be dragged by gravity to the bottom of a pit-shaped valley. Once at the bottom of this hole, the gradient will vanish leaving the stage to the random noise that will make the ball move around.

The need for sufficiently large gradients has the effect of strongly deforming the energy landscape around the true examples regions. At the beginning of training, the energy is pretty flat everywhere. After few backpropagations, the energy surface starts to reduce, bending in the regions of true data points. At this point the sampling is roughly a random walk: the energy is increased around randomly picked regions on the landscape. As these deformations become stronger, the gradients increase and the sampling is more and more attracted towards the edges of some energy pits, spending there most of the time. As a consequence, the energy further rises on the edges, making these valleys become more and more steep, until the gradient term of SGLD or SGHMC drives the sampling in the hole. This is the phenomenon that makes the energy of generated samples to reduce and reach the one of true data points.

The vanilla EBM does not impose any constraint on the energy landscape, therefore this tendency could generate energy landscapes very difficult to explore for a MCMC sampler.

A rich portion of literature about optimization of DNN[1] states that the feasibility and the quality of optimization are strongly related to the simplicity of the loss landscape. Too irregular landscapes or sharp minima can hinder from good generalization.

Since SGLD and SGHMC can be roughly imagined as noisy SGD, I believe that the quality of sampling can be improved in the same way optimization is improved, namely improving the traversability of the energy landscape during sampling.

[1] Deep Neural Networks

Batchnorm and skip connections

This paragraph underlines how EBM training is intrinsically different from other classic deep models, making some of the most popular training silver bullets completely useless in this setting.

Two well know strategies widely employed in literature to improve the loss landscape are batch normalization and skip connections. However, extensive experiments proved that these two techniques are not applicable in this setting.

Batch normalization has the effect of controlling the magnitude of information flow throughout the net during the forward pass but, as will be subsequently explained, we want the net to have large outputs magnitude in order to achieve good performances. It is not clear, but I believe that batchnorm also has a negative effect on sampling from the model, having an active role on the gradients. Furthermore, in the current state of the art literature there are no working models employing batch normalization in this setting.

Skip connections showed an outstanding positive effect in simplifying the loss landscape as showed in Figure 1. For this reason I carried out an extensive experimentation and tuning with DenseNet model. However, sampling is not identical to optimizing and the skip connections make the sampler generate samples of decreasing norm until nearly-zero image norms are produced. I have tried different numbers of Dense Blocks, Growth Rate, Compression Factor and whether use the bottleneck blocks or not. Regardless the model configuration, the tendency on the generated images norm was the same. Therefore I associate this unwanted behavior with the skip connections.

I have not tried Dropout, but this regularization technique may introduce an unwanted stochasticity that may hinder the convergence to a stationary distribution, which requires a quite deep analysis as already carried out for stochastic gradient in [1].

These techniques have been developed after long research and their non-applicability to EBM suggests that these models are still at the beginning and a lot of improvement has yet to come.

Activation function

In Section 2 it is explained that a key feature of EBMs is that they do not impose any constraint on the parametric function they are learning, which gives them the freedom to fit well any density. However, the bias-variance trade-off has to be considered: as previously presented in section 2.2, VAEs introduce a regularization of their latent space that make it possible to generate sensible samples.

I propose to introduce an activation function

$$\begin{aligned}\pi(f_\theta(\mathbf{x})) &= \frac{1}{2}f_\theta(\mathbf{x})^2 \\ \nabla_\mathbf{x}\pi(f_\theta(\mathbf{x})) &= f_\theta(\mathbf{x})\nabla_\mathbf{x}f_\theta(\mathbf{x})\end{aligned} \tag{20}$$

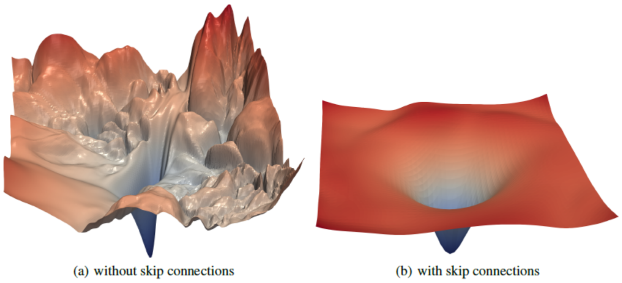


Figure 1: Effect of skip connection on the loss landscape, from [3].

on the last layer of the network, such that $E_\theta(\mathbf{x}) = \pi(f_\theta(\mathbf{x}))$, in order to regularize the energy landscape.

When optimizing the CD loss, the energy associated to real examples is reduced indefinitely. In theory, upon convergence the generated examples will be so close to the real ones that the contrastive effects of CD loss will counterbalance each other, keeping the energy in that region roughly constant. In practice, especially when the dimensionality of the problem is large, this is not guaranteed and a region around real samples may be difficult to traverse by the sampler: there are no theoretical constraints to prevent it from being highly irregular. Irregularity can be associated with discontinuity and locally large derivatives, which are a source of disturb for a good sampling.

With this activation function we introduce a lower bound to the energy of real examples and minimizing it means bringing that energy to zero. On the contrary, the energy of generated samples is still unbounded and can be arbitrarily maximized. Furthermore, with this activation it is now possible to have an absolute reference for the energies.

An alternative way to interpret this is that by optimizing the CD loss, we are asking the net to produce a fixed small output value (close to 0) when the samples are real and any large value (in magnitude) when the samples are generated. From a learning perspective, is like forcing the model to recognize something that *is* against all the other alternatives that *are not*.

Since I did not have enough computational power to study the effect of this activation on the hessian eigenvalues in the regions of true samples, I did not find the exact explanation for why this activation works. However, the results obtained on MNIST dataset suggest that this activation improved the quality of samples generation.

I also believe that this lower bound of the energy may be beneficial in the case of unbalanced datasets because there is not a ”preferred” class, having lower energy values associated. However this has to be proved in future experiments.

3.2. MCMC sampling

Chains initialization

There are basically two alternatives to initialize the Markov chains of the sampling methods. A popular strategy employed in literature is the informative initialization in which the chains are initialized once and subsequently are kept "alive" throughout the training, which is called Persistent Contrastive Divergence (PCD). This approach helps to convergence faster to the stationary distribution, but we believe that at the same time it introduces some bias and it could be also source of overfitting. Intuitively, if we want to generate animal pictures, when a chain is initialized form the image of a fish it will strongly biased towards generating fishes of differen species, but almost never it will have enough time to walk up to the rhinos "area" of the samples space.

Practical experience with Markov chains suggests that, in general, initializing the sampling around a "good" point is always a good idea, having the effect of reducing the burn-in time. The difficult task is to find a sufficiently good starting point. Furthermore, SGLD and SGHMC have very good convergence properties compared to a random walk, which reduces the burden of a "bad" initial point.

Alternatively, the chains can be initialized from random noise, which can be either uniformly distributed in $[0, 1]$ or $[-1, 1]$ or normally distributed, according to the standard Gaussian. This is called non-informative initialization.

In the experiments I will compare the performances of PCD versus Gaussian random initialization.

Relation among $\nabla_{\mathbf{x}} E_\theta(\mathbf{x}^-)$ and η

As presented in the previous section, the quality of sampling improves as the the gradient step $\eta \cdot \nabla_{\mathbf{x}} E_\theta(\mathbf{x}^-)$, after an initial transitory stage, reaches a similar order of magnitude of the noise step $\sim \sqrt{2\eta}$. At regime, the norm of the gradient will reach a stable value which guarantees a balanced contribution of gradient information and random noise.

This entails that the smaller η, the larger the norm of the gradient after the transitory stage. As a result, we have a trade-off among gradient norm and sampling precision. A larger η will require:

- Smaller gradient norms, thus more numerical stability.
- Faster exploration and maybe faster convergence of the Markov chains. This is an important feature when working in high dimensional spaces, as they are more sparse.
- Slower precision during sampling.

Since the training and the generation are highly correlated, the tuning of η is of key importance, being the behavior of the entire model depending on it. In fact, as it will be remarked in the results, different values of η made a considerable difference in the quality of generated samples.

There is a work in literature, [2], that performs a quite invasive correction of the presented *self-adaptation* of the gradient norm according to η. They propose to clamp the gradient and use $\eta = 10$. I do not agree with this choice because other than being invasive it seems also quite an arbitrary choice that is not justifiable from theory.

Related work

The paper of [4] states that:

1. It is not possible to converge to the stationary distribution without an informative initialization of the chains, unless employing MCMC chains of length of about 20000 steps.
2. If η is chosen too large the Langevin sampling degenerates to a stochastic gradient descent whereas if chosen too small it roughly behaves like a random walk. In fact, they propose a way to track the average gradient norm in order to detect unwanted behaviors.

Practical experience suggests that 500 steps in the sampling phase are usually enough, being in contrast with the first statement. Surprisingly enough, what influenced the most the quality of generation has been η.

I believe that the second statement is intrinsically wrong being the Langevin sampling self-consistent and reaching an equilibrium among gradient and noise contributions, upon convergence. Indeed at the beginning, the sampling will behave like a random walk being the gradient step negligible, but this has the good effect of visiting more the space surrounding the true samples regions.

4. Experiments

4.1. Toy 2D dataset

I hereby present the results obtained on a simple 2D dataset made from a Gaussian mixture model, following the example of [4]. Practical experience suggests that it is always a good idea to test these models on a dataset where the real density is known. This way, it is possible to compute some accurate distributional distance metrics that would not be available on a more classic dataset (e.g. MNIST), where the ground truth density is not known.

The following results are obtained with two different approaches:

- SGLD sampling (ML)
- SGHMC sampling (ML)

For each of these I compute the Kolmogorov-Sminrov distance, which measures the similarity of the samples obtained from two different distributions, hence indirectly the similarity of such distributions. Furthermore, to have a better contextualization of these results I compare them to the *self-distance* of the ground truth distribution, namely computing the K-S distance among two samples obtained from the target distribution. This reference metric represents a rough lower bound and it is referred as ”benchmark” in Table 1. K-S distance is a measure intended for 1D samples, which is not directly applicable in this case. However, it is possible to generalize it to two dimensions according to [6]. Additionally, I also compute the discretized KL divergence among the target and the learned density from the EBM. This is computed on a sufficiently dense grid.

4.1.1 Results on 2D datasets

With the results of Table 1 I prove that SGLD and SGHMC can achieve the stationary results, being the K-S distance below 0.1 and close to the benchmark. The discretized KL divergence give an additional confirmation of this.

However, it is also possible to notice that the performances achieved by SGLD and SGHMC are quite similar. Both models have also been trained with informative initialization, namely Persistent Contrastive Divergence, sticking to what state of the are theory suggests. In fact, in the works [4] and [5] it is strongly argued that an informative initialization of chains is necessary to converge to a stationary distribution.

The results here presented show that not only PCD is not necessary to converge to the true stationary distribution but informative initialization slightly worsened the performances.

The quadratic activation function on the last layer did not show any benefit to be applied in this setting. However, its results are consistent with the others.

In Figure 3 it is possible to notice the effect on the steepness of the learned energy landscape depending on the value of η, as previously stated in Section 3.1.

The optimal hyperparameters resulting from tuning are presented in Table 3.

4.2. MNIST dataset

The experiments carried out on this dataset have shown since the beginning many difficulties. This, together with the lack of literature concerning EBM training with Maximum Likelihood estimation using pure Langevin sampling required me to devise the regularization of the energy landscape through quadratic activation function on the last layer. Unfortunately the space is high dimensional and it is not possible to easily asses *why* this activation is actually helping the training. However I have three hypotheses:

- As suggested in Section 3.1, it may improve the energy smoothness around true samples regions.
- The activation function forces the previous layers to produce low output values for true samples and high output values for generated ones (regardless the sign). This may ease the task of the optimizer when updating the parameters θ of the network.
- It may also improve the Lipschitz constant of the energy, ameliorating the convergence properties of the sampling, being gradient driven. Also the optimizer on the network parameters could benefit from this.

To further justify this regularization I have trained a model with the following activation on the last layer $f_\theta(\mathbf{x})$ such that:

$$E_\theta(\mathbf{x}) = f_\theta(\mathbf{x}) + \frac{\beta}{2} f_\theta(\mathbf{x})^2$$

where β is a learnable parameter randomly initialized around 0. The training of the model increased its value until $\beta \approx 1$. To further constrain the value of the pixel to lie on the unit sphere, while sampling, I also tried the following

$$E_\theta(\mathbf{x}) = f_\theta(\mathbf{x}) + \gamma\sqrt{n}||\mathbf{x}||_2^2$$

where γ is a learnable parameter randomly initialized around zero and n is the size of $\mathbf{x}$.

The optimizer brought γ to zero, suggesting that this regularization is useless. I additionally tried a third model with both learnable β and γ and I noticed the same behavior. Further manual tuning on β gave additional evidence on its usefulness.

Further reasoning and experiments took me to define the quadratic activation as in Equation 20. Empirical experiments suggest that when a model is trained with this activation, the samples quality may be less prone to degrade as the sampling iterations increase.

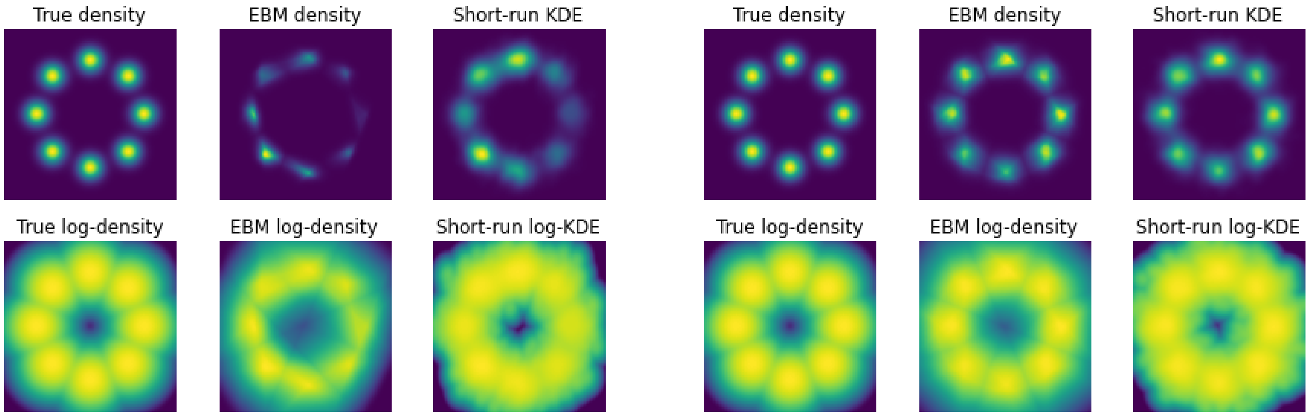
Effect of batch size

The batch size has an important effect on the nature of generated samples. As the batch size increases, the samples loose variability and more straight-shaped digits are generated (e.g. ”1” or ”7”) and vice versa. A good batch size value has been found to be 64.

4.2.1 Results on MNIST dataset

Since SGLD and SGHMC results on the toy datasets are quite similar and given the difficulty of tuning SGHMC when training on MNIST dataset, I decided to focus on SGLD.

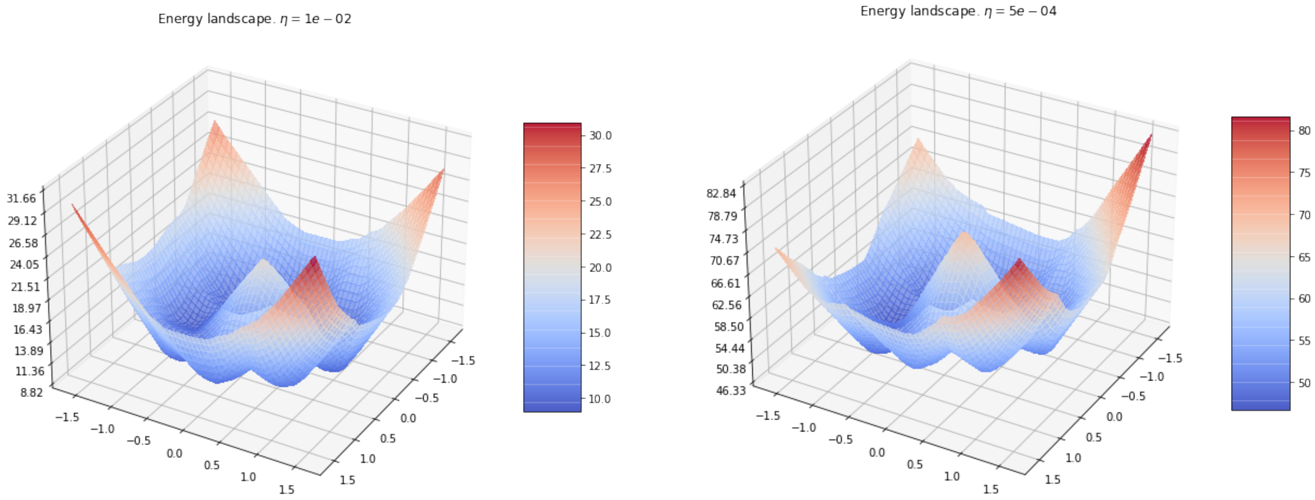
I trained and compared two CNN architectures. The first one I used is LeNet model. The vanilla implementation employs tanh activations, however these proved to strongly negative affect gradient propagation. LeakyReLU



(a) SGLD without informative initialization.

(b) SGHMC without informative initialization.

Figure 2: Best toy models fitted density comparison.



(a) $\eta = 0.01$

(b) $\eta = 0.0005$

Figure 3: Energy landscapes for 2D toy dataset when varying the value of η. Note the scale.

and Swish activation functions were more suitable. The second model I tried is a custom CNN with more capacity and stride instead of average pooling, with the goal of studying how network capacity influences generation. This model was slightly better than LeNet.
In Figure 4 are presented the results obtained on MNIST dataset for three configurations of SGLD:

(a) Vanilla: random initialization of Markov chains and linear activation function on the last layer.

(b) PCD initialization and linear activation.

(c) Random initialization and quadratic activation.

In figure 5 I show how increasing the number of sampling iterations on my best model does not causes samples degradation. This result is worth of note since this is not always guaranteed. The best model hyperparameters are reported in Table 4 and they result from an extensive tuning. The more critical hyperparameter on the quality of the outcome and on the stability of the model was η.

5. Conclusions

The quality of the generated images from the best model underline how much the training of these EBM may be hard, especially in an highly dimensional space. As presented in Section 2.2, there are some promising techniques like the latent dimension EBM that try to overcome the burden of sampling in an high dimensional space. However, the aim

Model	K-S (B*)	K-S	D_{KL}
SGLD (with PCD)	5.19e-02 ± 1.09e-02	1.59e-01 ± 1.22e-02	2.92e-01 ± 3.04e-02
SGLD (no PCD)	5.19e-02 ± 1.09e-02	1.54e-01 ± 1.37e-02	3.78e-01 ± 2.13e-02
SGLD (activation)	5.19e-02 ± 1.09e-02	2.01e-01 ± 1.18e-02	4.59e-01 ± 2.76e-02
SGHMC (with PCD)	5.19e-02 ± 1.09e-02	8.61e-02 ± 2.19e-03	8.88e-02 ± 5.10e-03
SGHMC (no PCD)	5.19e-02 ± 1.09e-02	5.25e-02 ± 3.06e-03	7.99e-02 ± 4.29e-03
SGHMC (activation)	5.19e-02 ± 1.09e-02	8.16e-02 ± 2.45e-03	7.74e-02 ± 4.66e-03

Table 1: Kolmogorov-Smirnov distance (sample size 1000) an KL divergence among true and fitted density for EBM with ML estimation on 2D GMM dataset. Best models. Format: mean ± std, on 5 different random seeds. B*: benchmark.

Model	Optimizer	ϵ	Weight decay	η	Batch size	epochs	L	C	m
SGLD (with PCD)	SGD	0.01	0	0.01	100	10	500	-	-
SGLD (no PCD)	SGD	0.01	0	0.01	100	10	500	-	-
SGLD (activation)	SGD	0.01	0	0.01	100	10	500	-	-
SGHMC (with PCD)	SGD	0.01	0	0.005	100	10	500	1	1
SGHMC (no PCD)	SGD	0.01	0	0.01	100	10	500	1	0.5
SGHMC (activation)	SGD	0.005	0	0.01	100	10	500	1	0.5

Table 2: Best hyperparameters. Notation according to Algorithms 1, 2 and 3.

Layer	Output size	Details
Input img	1 x 28 x 28	-
5 x 5 conv	16 x 16 x 16	stride 2, padding 4
3 x 3 conv	32 x 8 x 8	stride 2, padding 1
3 x 3 conv	64 x 4 x 4	stride 2, padding 1
3 x 3 conv	64 x 2 x 2	stride 2, padding 1
fc	64	-
fc	1	-

Table 3: Custom CNN architecture.

Hyperparameter	Values
CNN	custom
Optimizer	Adam
Learning rate ϵ	1e-04
Weight decay	1e-4
η	5e-06
L	1000
Batch size	64
Epochs	10

Table 4: Best model hyperparameters for MNIST dataset.

of this semester project was more theoretically oriented to explore the world of EBM trained with maximum likelihood estimation, which is often ovelooked. A proof of this is the lack of literature.

This work tried to approach the matter basing experiments and reasoning on strong mathematical bases, therefore this can be referred by a reader wanting to have a introduction on EBM and having a reference work as a guide at the same time.

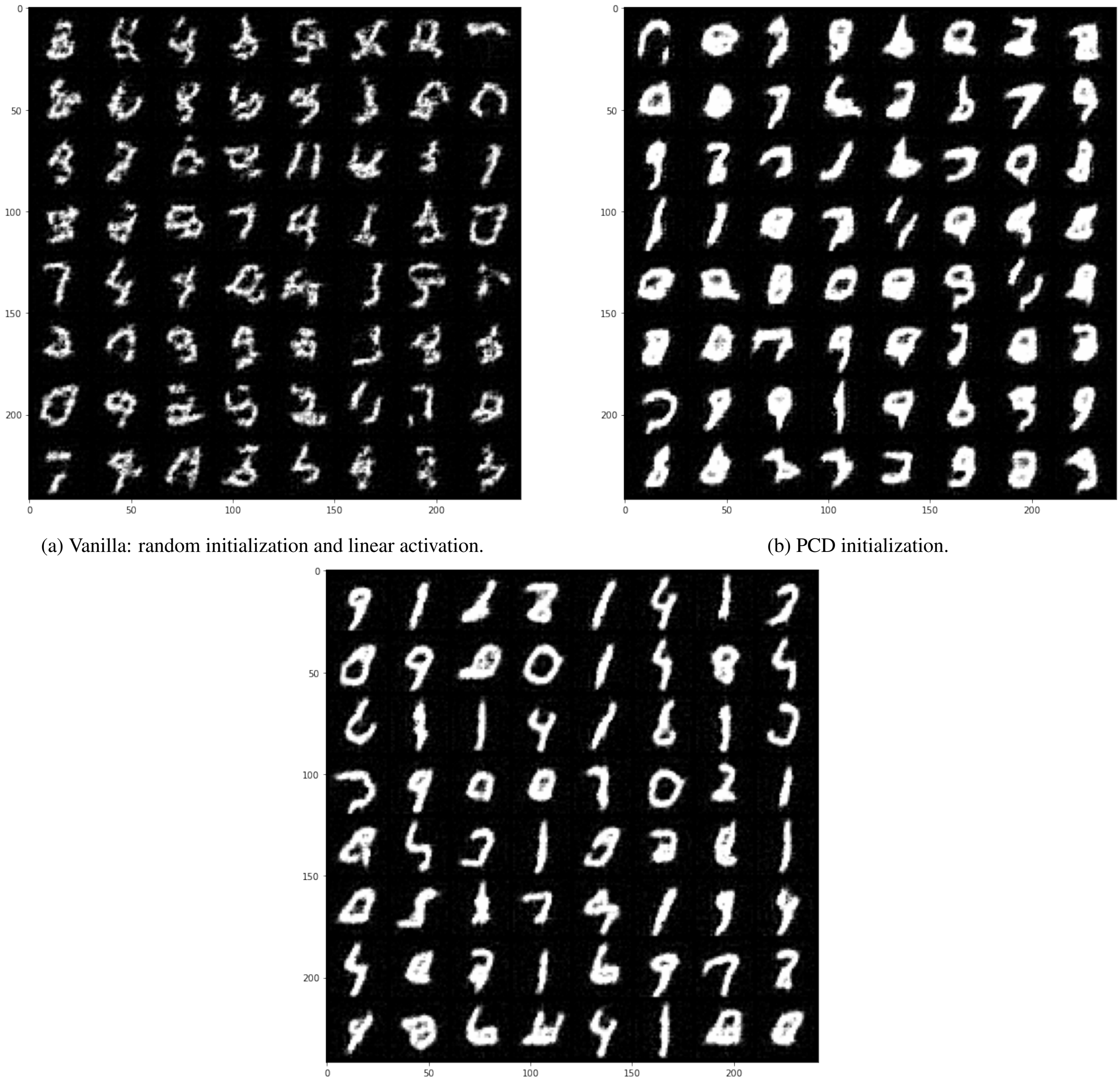
I have discovered by empirical observation that a quadratic activation function my be beneficial for training, especially in high dimensional spaces like MNIST dataset, where there are 784 dimensions. Nevertheless, I was not able to precisely determine the effect on the convergence of the Markov chains or on the local curvature of the energy landscape, which is let for future work. I strongly believe that this is a promising direction to further explore and it may result in being a valid alternative to PCD.

My results are not comparable to the ones of some works in the current literature, despite having been obtained after extensive experimentation based on strong mathematical bases. A suggestion to further improve my results is to resort to non-local convolutional layers, as done in [4] to solve a similar problem.

Despite being quite garbled models, I want to define some general principles I believe would help a newbie avoid some problems I faced:

- Test whether the model is converging to the stationary distribution running some experiments on a toy dataset like the one I used. This is always a good sanity check of the code as well.

- Focus on the tuning of η parameter, having understood the relation between η, $||\nabla_{\mathbf{x}} E_\theta(\mathbf{x}^-)||$ and the magni-



(a) Vanilla: random initialization and linear activation.

(b) PCD initialization.

(c) Random initialization and quadratic activation.

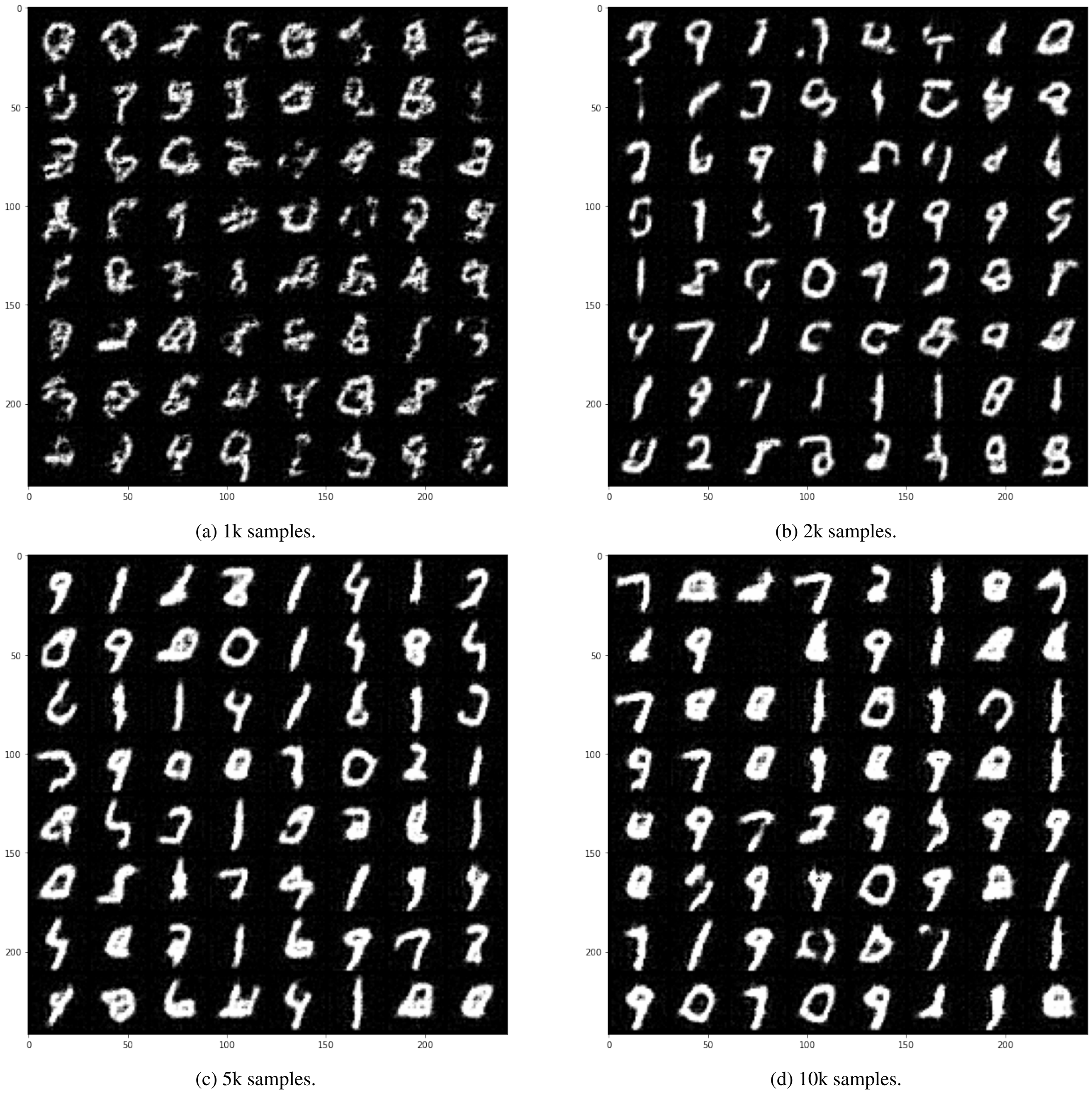
Figure 4: Comparison of the results of different configurations of the best model.

tude of $E_\theta(\mathbf{x})$.

- *Sufficiency*. Increasing L and the number of epochs improves the quality of generated samples. However, to have a glance of the correct learning of the model 3 epochs and $L = 200$ are usually enough.

- Avoid any practitioner alteration in the sampling algorithm, namely gradient clamping, gradient normalization, normalization of generated batch at each iteration. Being the sampling quite sensitive to alterations and deeply bonded to parameters optimization, any action in this sense may cause the training to diverge. Regardless whether the training diverges or not, the theoretical guarantees no more hold.

- Avoid batch normalization.



(a) 1k samples.

(b) 2k samples.

(c) 5k samples.

(d) 10k samples.

Figure 5: Effect of increasing number of sampling iterations on the best model. There is no degradation, but slight saturation.

- I would suggest to be cautious with skip connection since my experiments on DenseNet always failed, regardless the hyperparameters configurations or putting in place other stratiegies that worked on other models.

References

[1] T. Chen, E. B. Fox, and C. Guestrin. Stochastic gradient hamiltonian monte carlo, 2014.

[2] Y. Du and I. Mordatch. Implicit generation and generalization in energy-based models, 2020.

[3] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets, 2018.

[4] E. Nijkamp, M. Hill, T. Han, S.-C. Zhu, and Y. N. Wu. On the anatomy of mcmc-based maximum likelihood learning of energy-based models, 2019.

[5] E. Nijkamp, M. Hill, S.-C. Zhu, and Y. N. Wu. Learning non-convergent non-persistent short-run mcmc toward

energy-based model, 2019.

[6] W. H. Press and S. A. Teukolsky. Kolmogorov-smirnov test for two-dimensional data. *Computers in Physics*, 2(4):74–77, 1988.

[7] Y. Song, S. Garg, J. Shi, and S. Ermon. Sliced score matching: A scalable approach to density and score estimation, 2019.

[8] Y. Song and D. P. Kingma. How to train your energy-based models, 2021.

[9] Z. Xiao, Q. Yan, and Y. Amit. Exponential tilting of generative models: Improving sample quality by training and sampling from latent energy, 2020.