GAN as an EBM

Problem statement

How can one improve the quality of GAN samples ?

- Discriminator Rejection Sampling
- Metropolis-Hastings GAN
- Discriminator Optimal Transport

Drawbacks : inefficient or implying mode collapse

Idea : Consider GAN as Energy-Based Model and sample from latent space

GAN as an Energy-Based Model

GAN

Original data $(X_1, \ldots, X_N) \in \mathbb{R}^{N \times d}, \quad X_i \sim p_d$

Generator $G : \mathcal{Z} \rightarrow \mathcal{X}, \quad z \in \mathcal{Z}, \quad z \sim p_z$

Discriminator $D : \mathcal{X} \rightarrow \mathbb{R}$

EBM

State space χ

Energy $E : \chi \rightarrow \mathbb{R}$

Boltzmann distribution $p(x) = \dfrac{e^{-E(x)}}{Z}, \quad x \in \chi$

Normalizing constant $Z = \int e^{-E(x)} dx$

GAN as an Energy-Based Model

Trained GAN with generator distribution p_g, we assume $G(z)$ is imperfect

Discriminator is almost optimal: $D(x) \approx \dfrac{p_d(x)}{p_d(x) + p_g(x)}$

Logit of $D(x)$: $d(x)$, $D(x) = \sigma(d(x))$

$$D(x) \approx \frac{1}{1 + \exp(-d(x))}$$

Energy-Based Model: $p_d^* = p_g(x)e^{d(x)}/Z_0$

If $D = D^*$, D^* - optimal discriminator, then $p_d^* = p_d$

GAN as an Energy-Based Model

Main theorem

Theorem 1. *Assume p_d is the data generating distribution, and p_g is the generator distribution induced by the generator $G : \mathcal{Z} \to \mathcal{X}$, where $\mathcal{Z}$ is the latent space with prior distribution $p_0(z)$. Define Boltzmann distribution $p_d^* = e^{\log p_g(x) + d(x)} / Z_0$, where Z_0 is the normalization constant.*

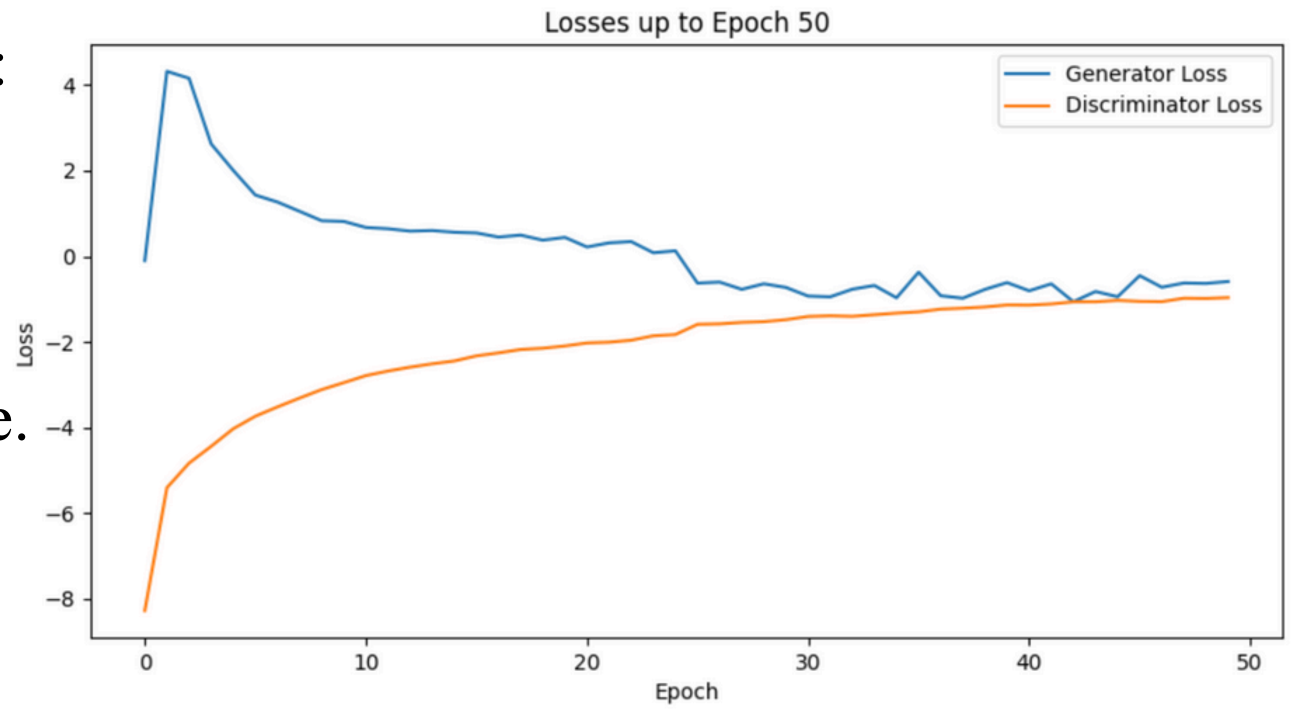
Assume p_g and p_d have the same support. We address the case when this assumption does not hold in Corollary 2. Further, let $D(x)$ be the discriminator, and $d(x)$ be the logit of D, namely $D(x) = \sigma(d(x))$. We define the energy function $E(z) = -\log p_0(z) - d(G(z))$, and its Boltzmann distribution $p_t(z) = e^{-E(z)} / Z$. Then we have:

1. *$p_d^* = p_d$ when D is the optimal discriminator.*
2. *If we sample $z \sim p_t$, and $x = G(z)$, then we have $x \sim p_d^*$. Namely, the induced probability measure $G \circ p_t = p_d^*$.*

WGAN-GP as an EBM

Compared to the original GAN algorithm, the WGAN undertakes the following changes:

- After every gradient update on the critic function, clamp the weights to a small fixed range, [-c, c].
- Use a new loss function derived from the Wasserstein distance, no logarithm anymore. The "discriminator" model does not play as a direct critic but a helper for estimating the Wasserstein metric between real and generated data distribution.



Wassersetein GANs (WGANs) target Kantorovich-Wassersetein distance so its objectives are :

$$L_D = \mathbb{E}_{p_g}[D(x)] - \mathbb{E}_{p_d}[D(x)], \qquad L_G = -\mathbb{E}_{p_0}[D(G(z))].$$

where D has to be K-Lipshitz function

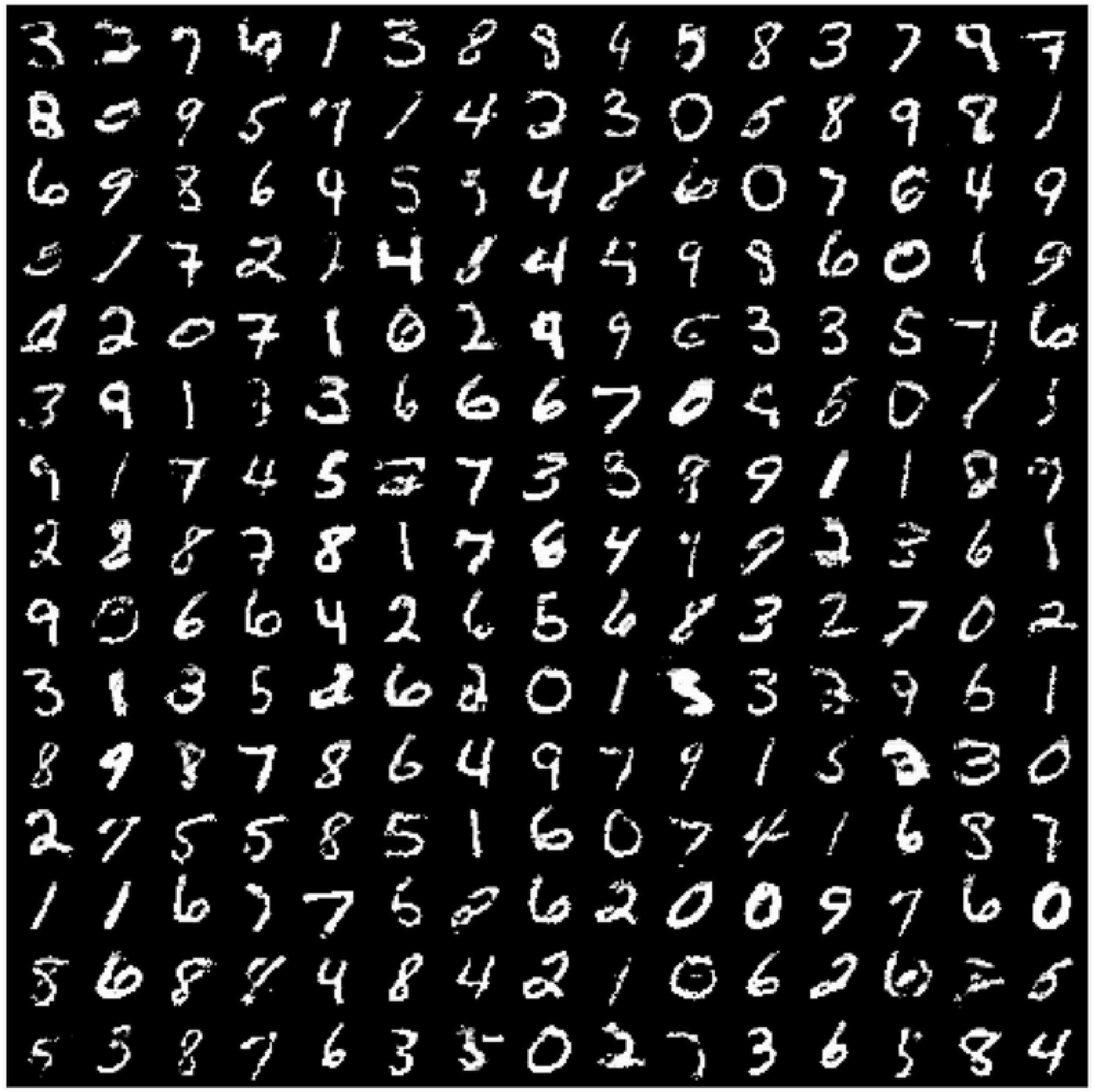
Therefore

$$E(z) = D_\phi(z) - \log p_g(z).$$

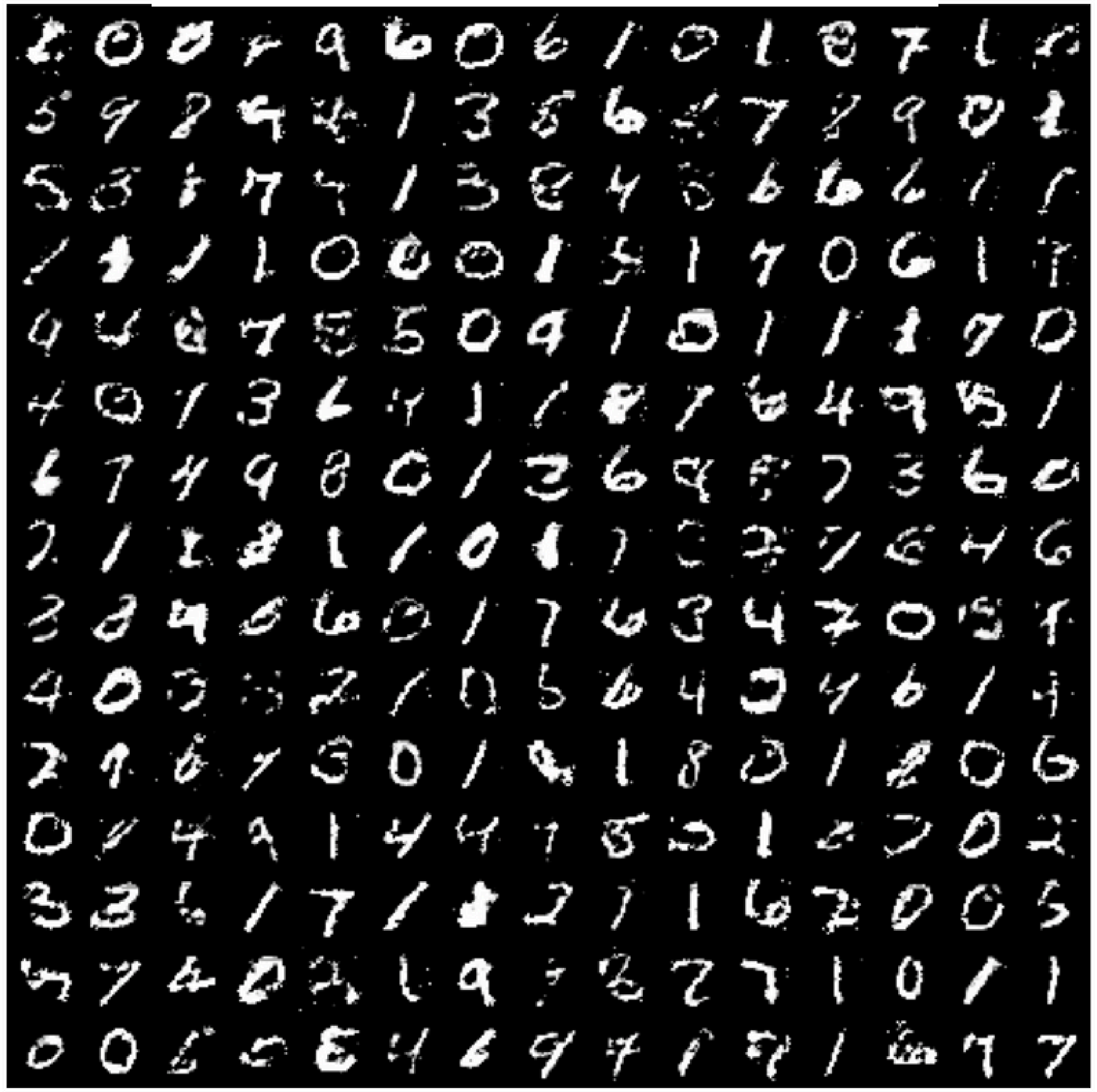
Comparison

- Trained over 100 epochs
- Adam Optimizer

WGAN-GP



GAN



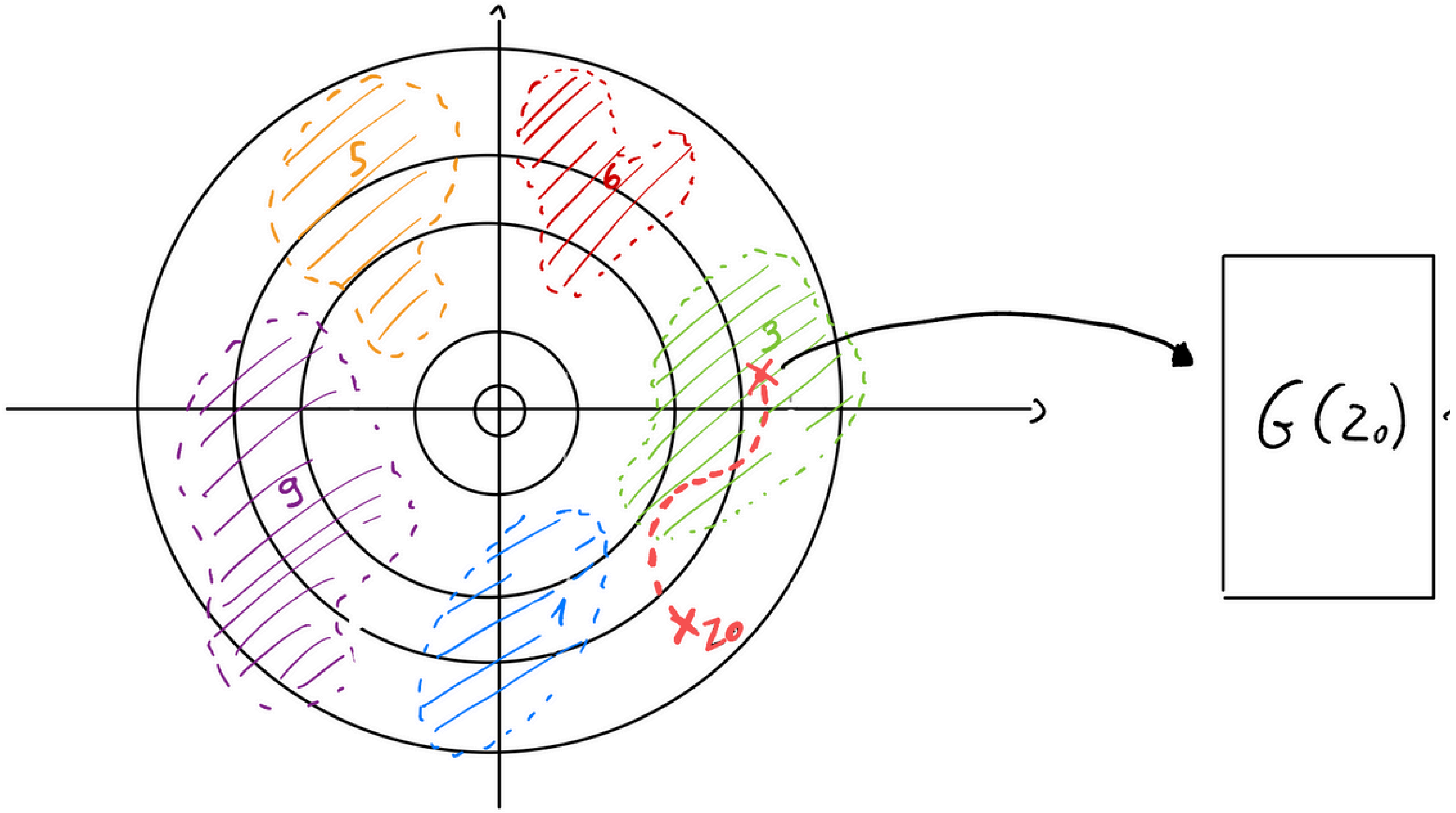
EBM and Langevin sampling

Pseudo Code

Algorithm 1 Discriminator Langevin Sampling

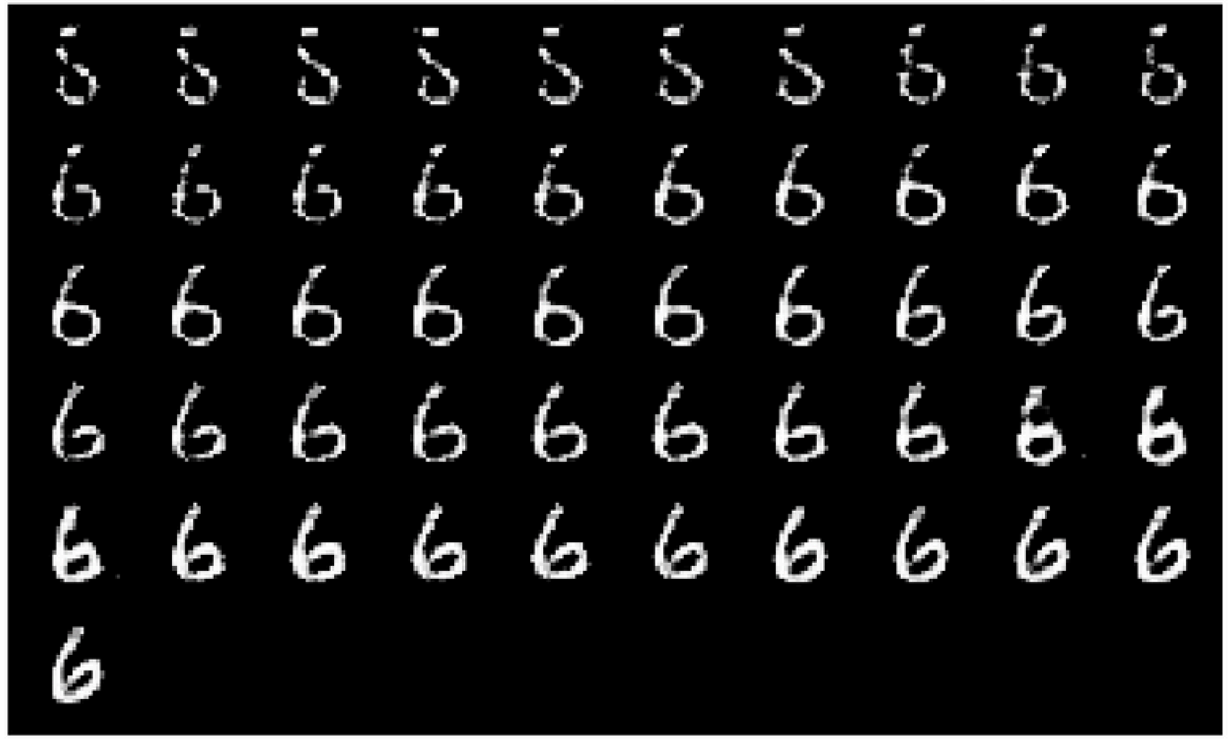
Input: $N \in \mathbb{N}_+, \epsilon > 0$
Output: Latent code $z_N \sim p_t(z)$
Sample $z_0 \sim p_0(z)$.
for $i < N$ **do**
  $n_i \sim N(0, 1)$
  $z_{i+1} = z_i - \epsilon/2\nabla_z E(z) + \sqrt{\epsilon} n_i$
  $i = i + 1$
end for

Intuition

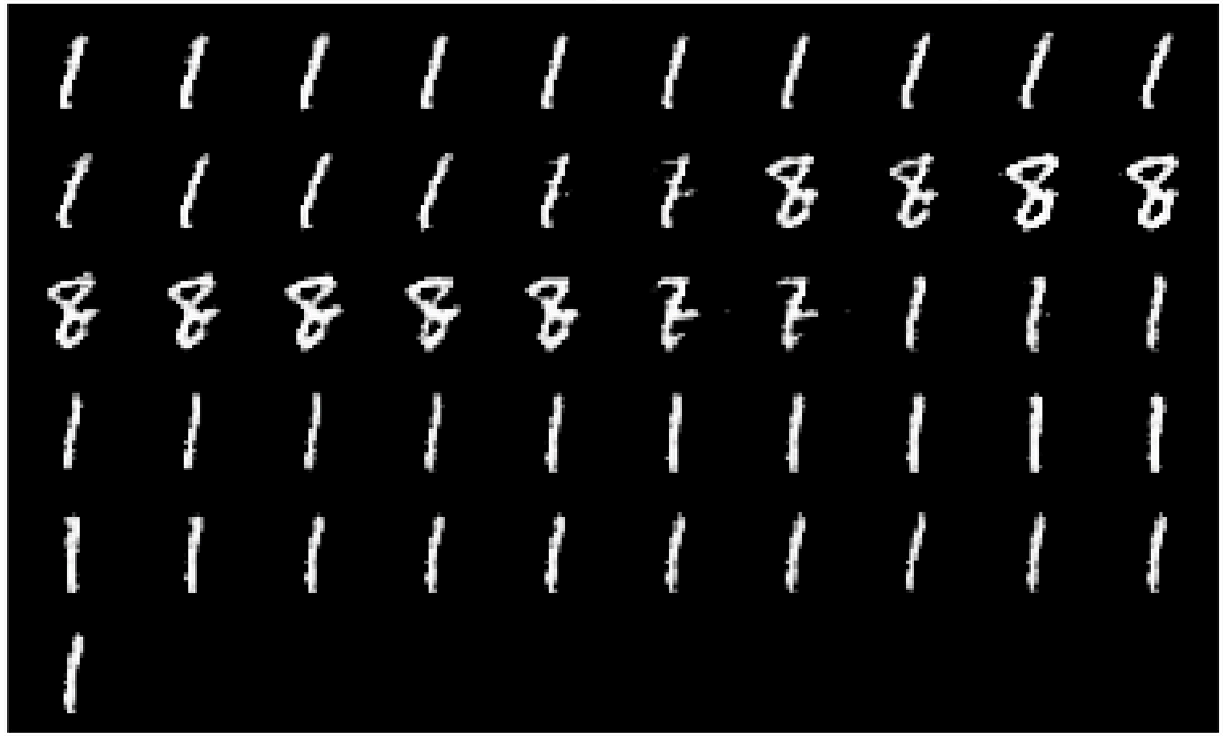


What does langevin sampling perform ?

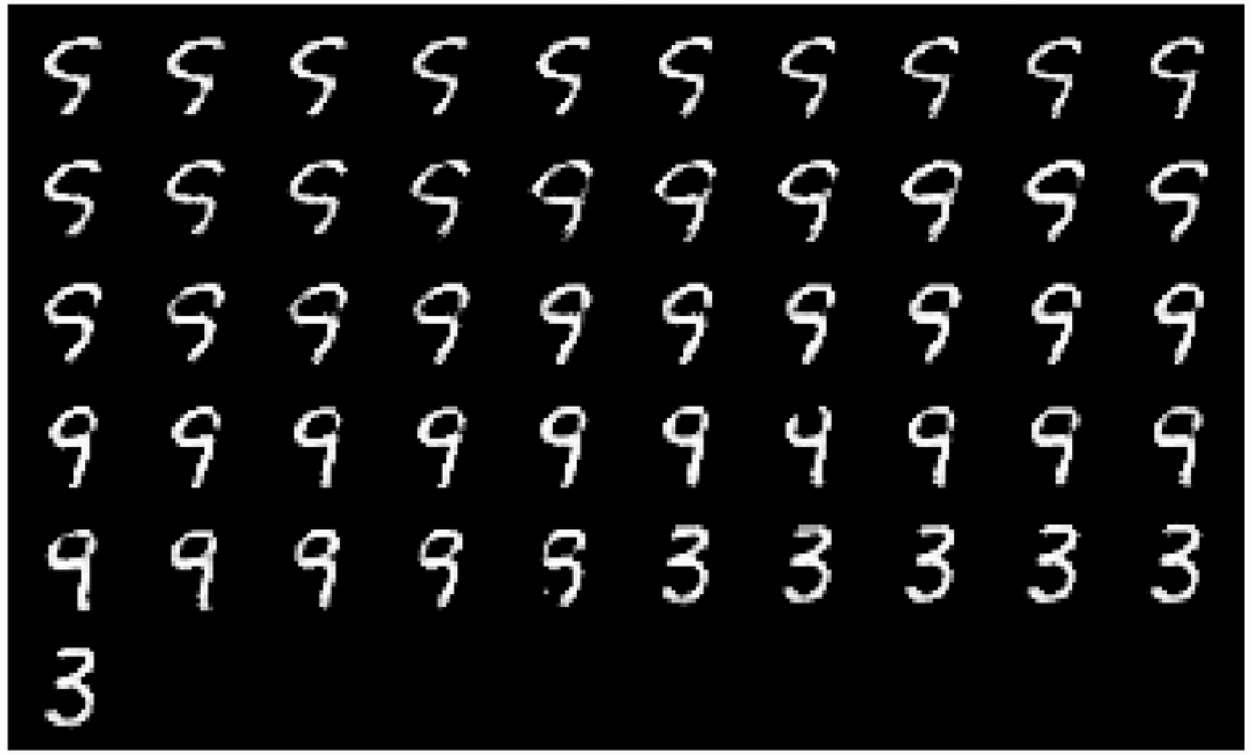
Evolution of the generated images



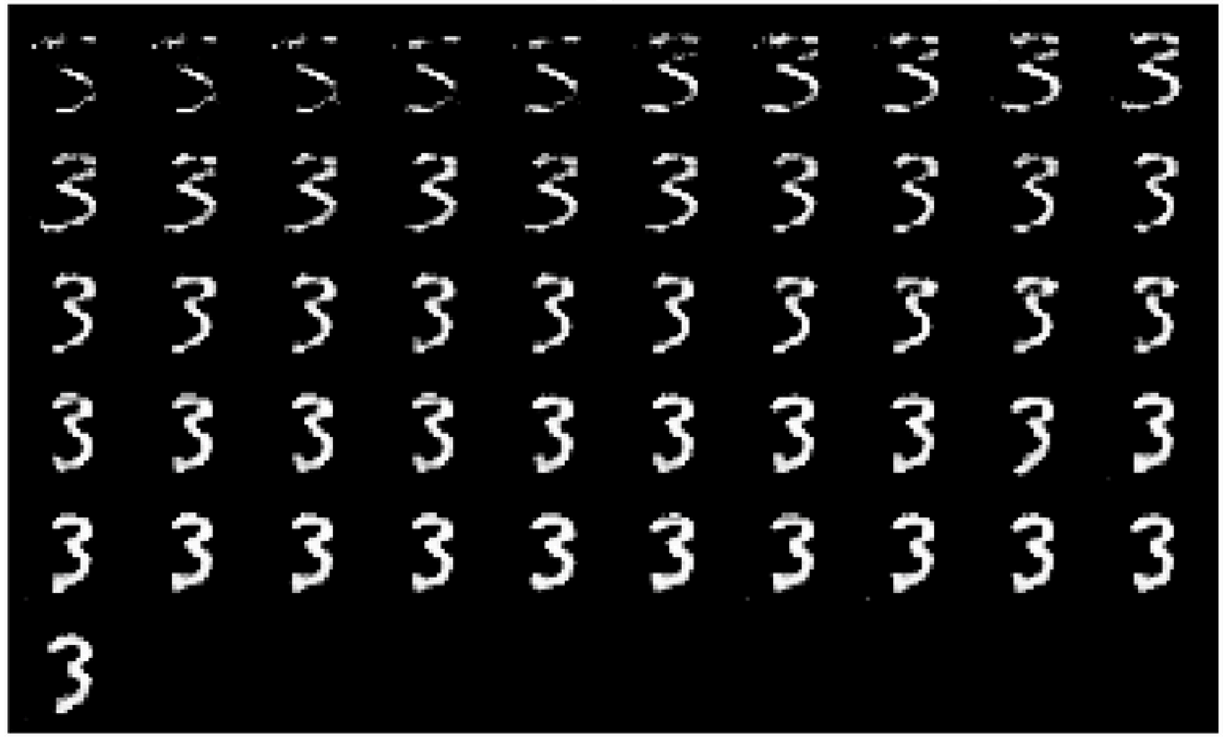
Evolution of the generated images



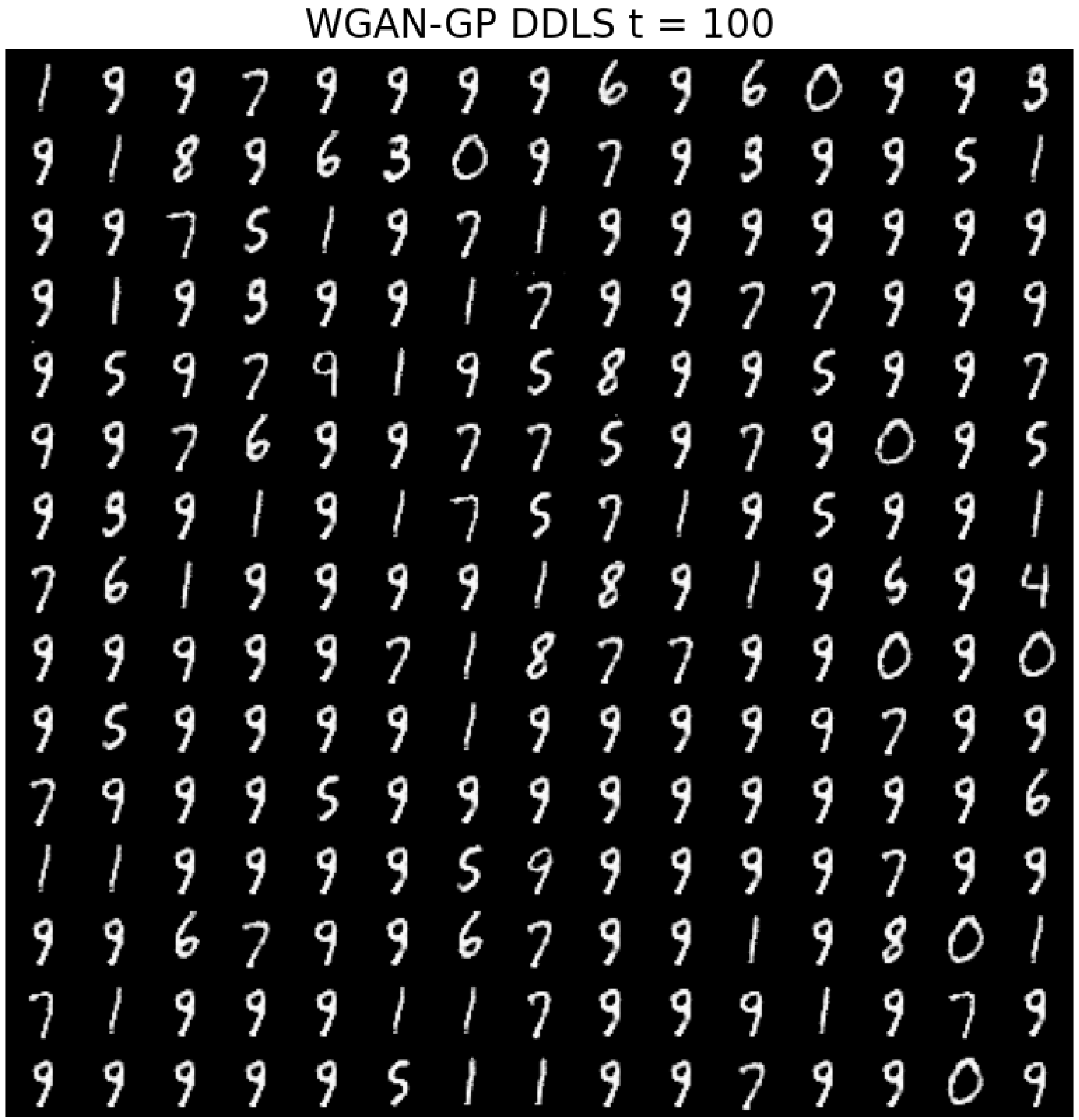
Evolution of the generated images



Evolution of the generated images

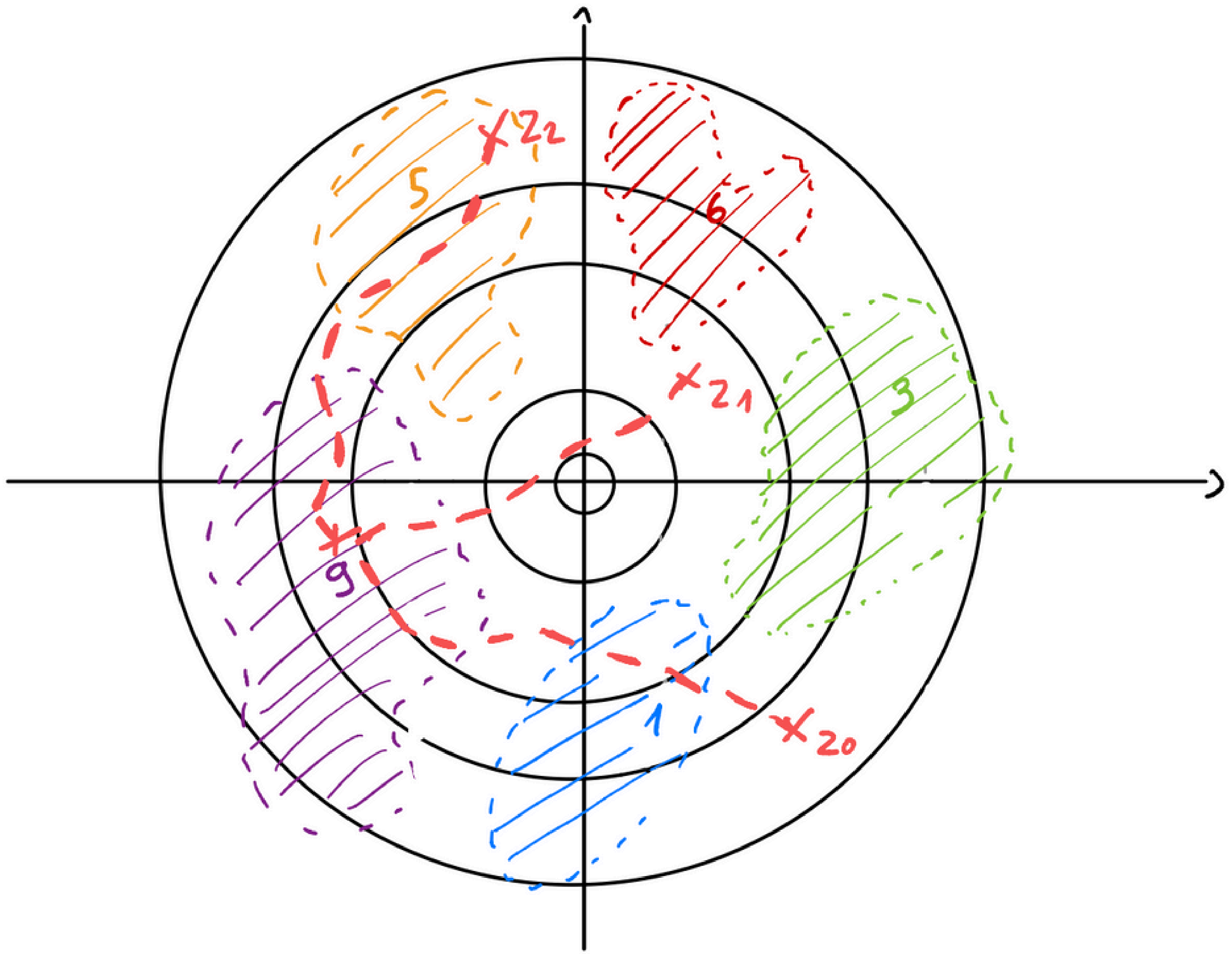


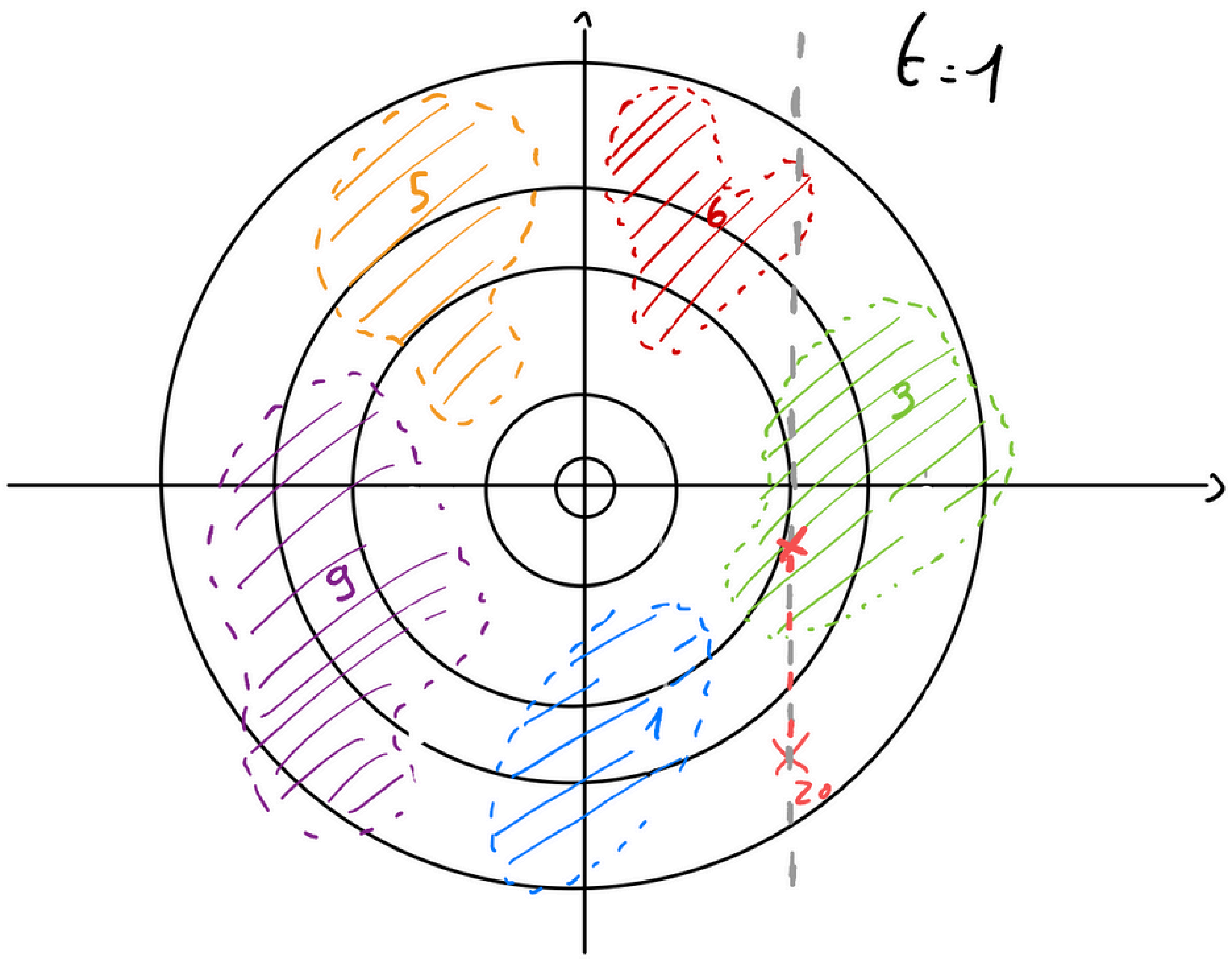
Problem



Dimension restriction

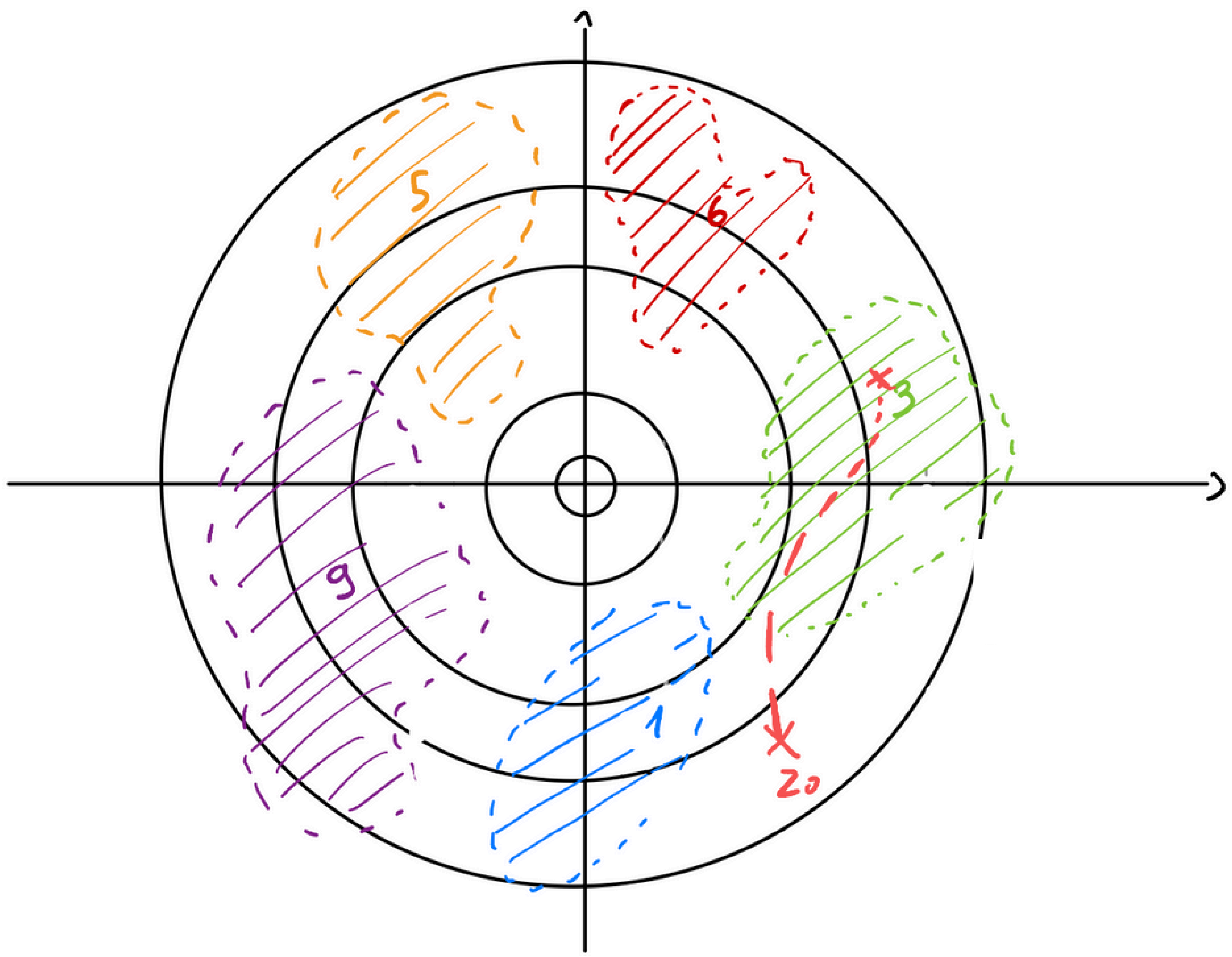
Instead of giving the Langevin Sampling algorithm the freedom to update z in latent space by updating all directions, we fix certain directions and force it to update only certain dimensions.





Langevin Sampling Tuning

Langevin rate decay

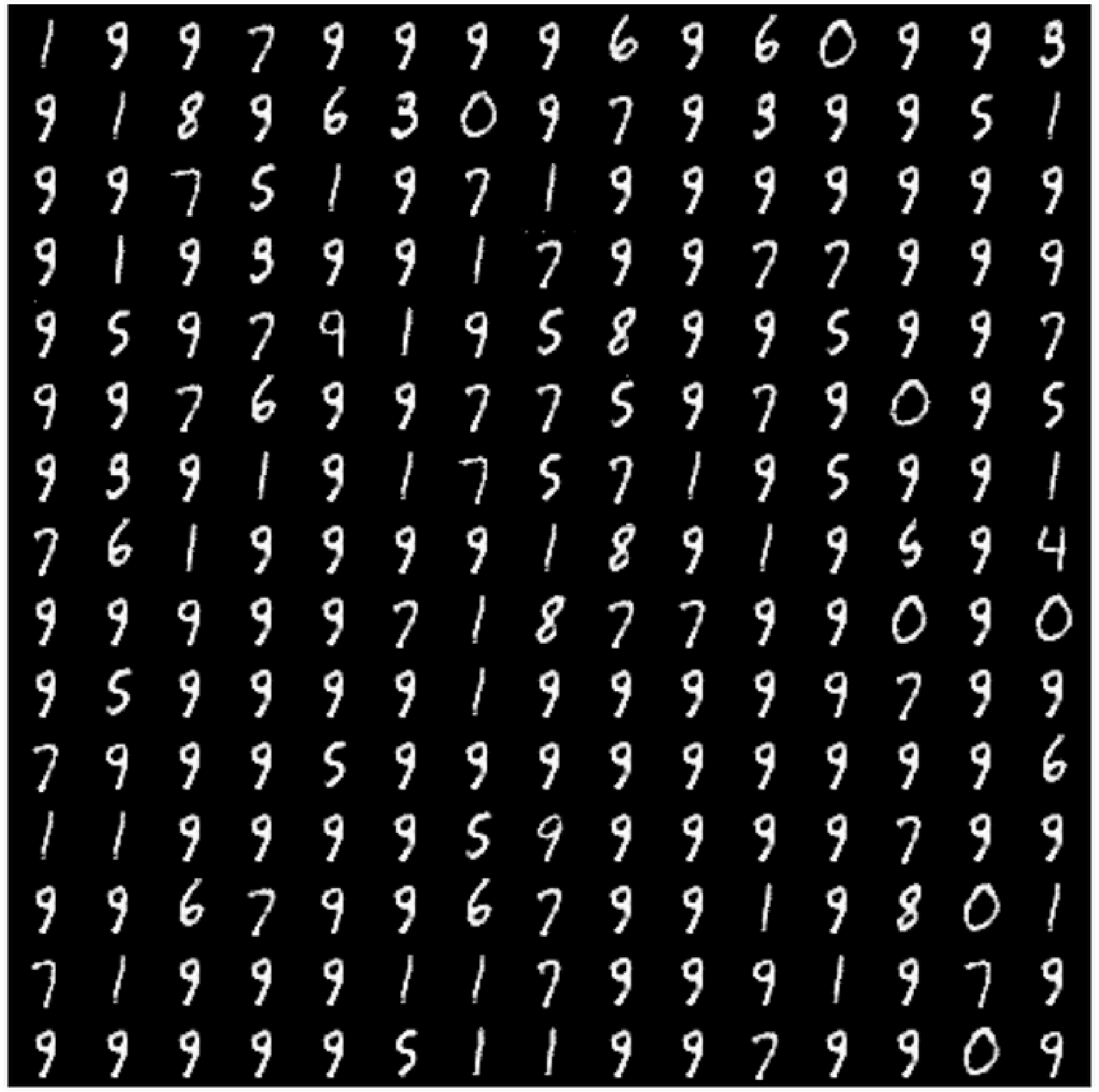


Regulating diversity

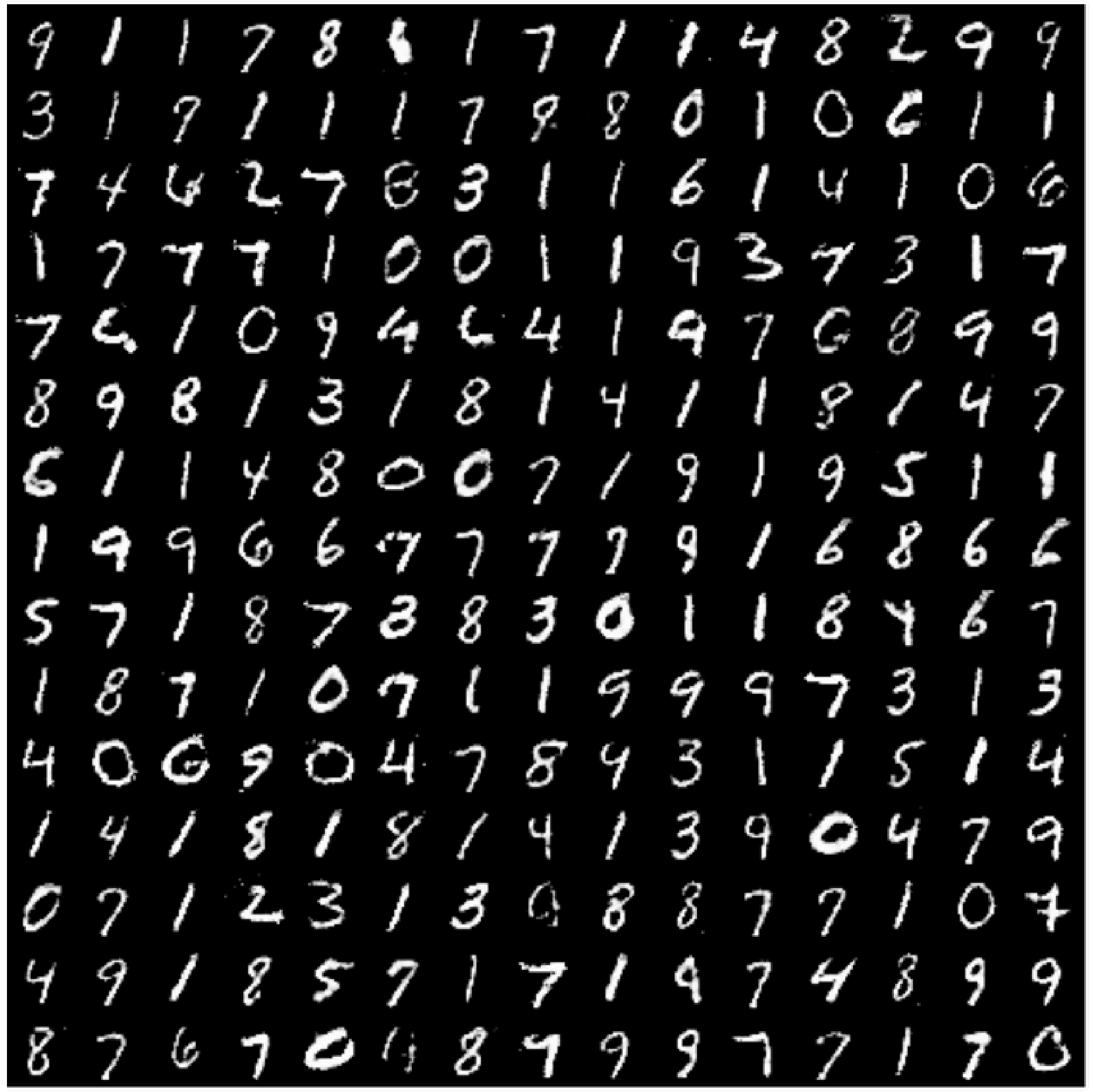
```
# Calculer la moyenne des distances de paire
mean_pairwise_distance = torch.mean(torch.pdist(zs, p=2))

# Calculer la perte de diversité en utilisant l'inverse de la moyenne des distances
diversity_loss = diversity_reg / (mean_pairwise_distance + 1e-8)

# Ajouter la perte de diversité à l'énergie
energy += diversity_loss
```

Precision Recall Tradeoff

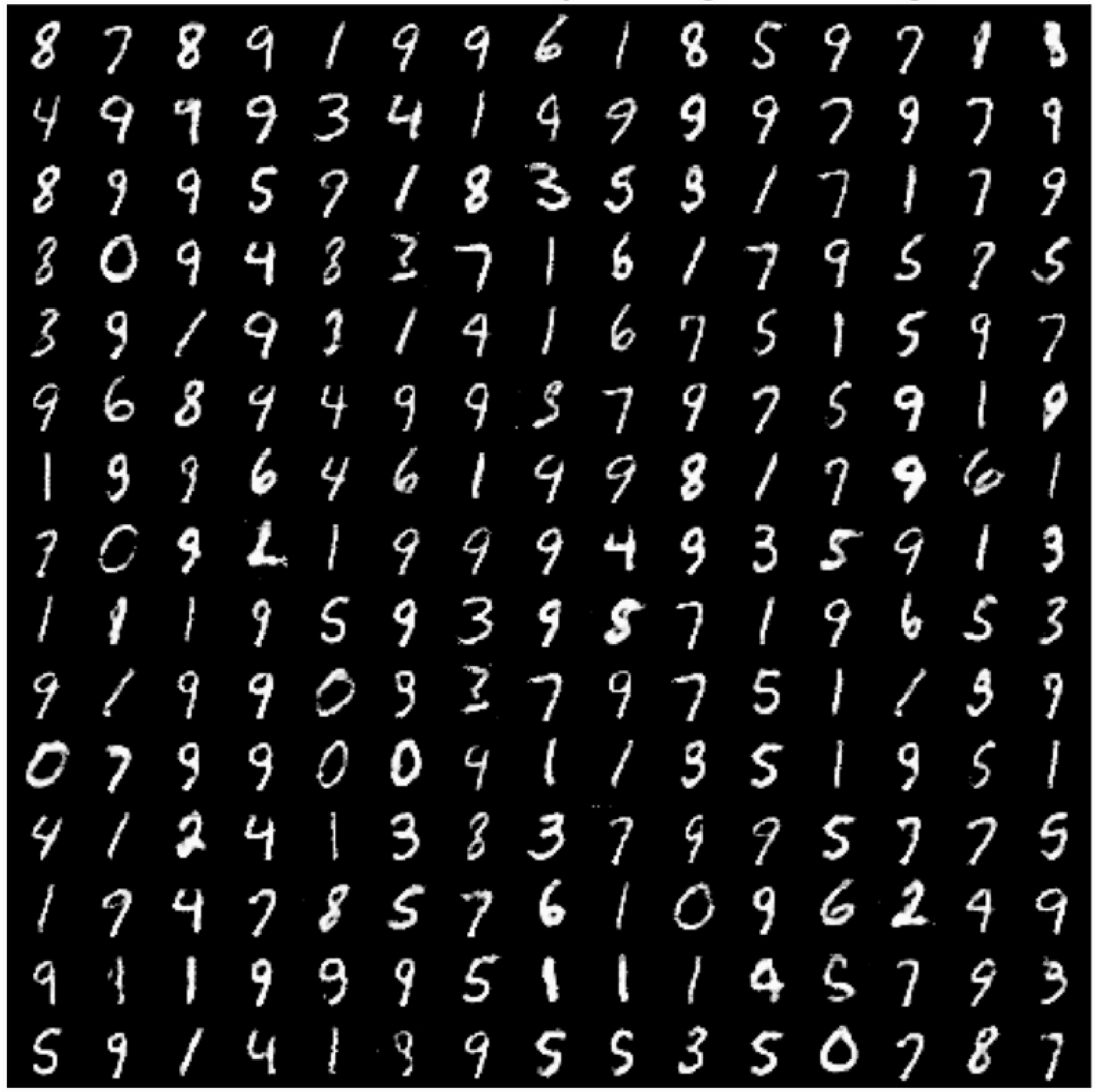
WGAN-GP DDLS t = 100



WGAN-GP DDLS t = 82



WGAN-GP t = 82, 150 steps, regularizing diversity



To go further

- Visualise the path of noise z in its latent space (Classifier, generator inversion, PCA to visualise)

- Since the assumption of a perfect discriminator: Improve the discirminateur to better drive the path of the Langevin sampling in the latent space

- Convolutional architecture