Thesis Defense

Compressed Sensing MRI using Score-based Implicit Model

Dimitrios Dimos

School of Electrical & Computer Engineering
National Technical University of Athens

Supervisors: Prof. Georgios Stamou, NTUA
Prof. Alexandros Dimakis, UT Austin

Nov. 10, 2022

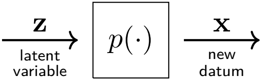
Summary

1. Introduction
2. Score-based Generative Modeling
3. Diffusion Models
4. Generation through Stochastic Differential Equations (SDEs)
5. Solving Inverse Problems using Score-based Models
6. Contribution - Score-based Implicit Model
7. Compressed Sensing MRI using SBIM

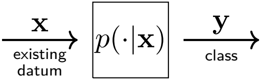
Introduction

General definitions

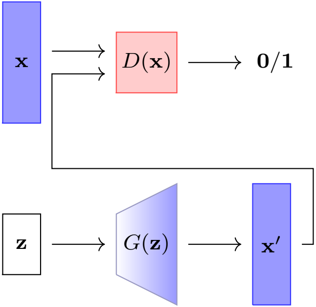
Generative model: captures (a representation) of the distribution $p(\mathbf{x})$



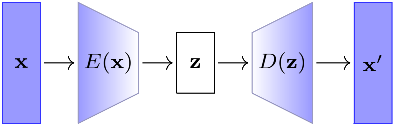
Discriminative model: captures the conditional probability $p(\mathbf{y}|\mathbf{x})$



Famous deep generative models (1/2)

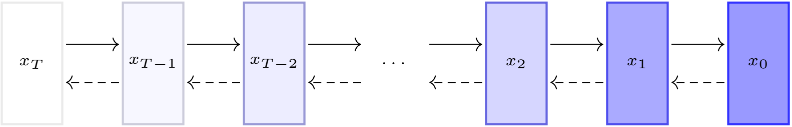


Generative Adversarial Networks (**GANs**)



Variational Autoencoders (**VAEs**)

Famous deep generative models (2/2)



Diffusion Models and Score-based Models

In literature, **both** models are often referred to as **diffusion-based** models, or simply **diffusion models**.

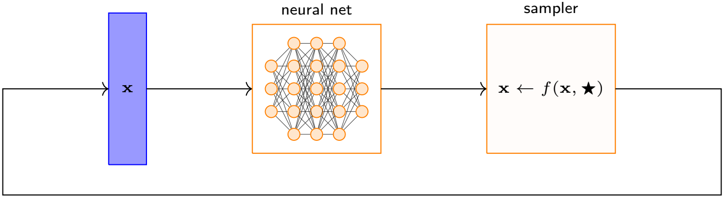
Why diffusion-based models?

Diffusion models have become increasingly popular the last 3 years. Why?

- State-of-the-art performance on many downstream tasks, such as:
  - image generation (beating even GANs !!!)
  - audio synthesis
  - shape generation
  - music generation
- Can be incorporated to solve inverse problems, among which are:
  - inpainting
  - deblurring
  - colorization
  - compressed sensing

Score-based Generative Modeling

Diffusion-based Generation Ingredients



- a **neural net**: produces a mathematical quantity ($\bigstar$)
- a **sampler**: an iterative procedure that updates $\mathbf{x}$ using $\bigstar$.

Langevin Dynamics

The first sampler for score-based models: **Langevin Dynamics**.

$$\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} + \frac{\epsilon}{2} \underbrace{\nabla_x \log p(\mathbf{x}_{t-1})}_{\text{score function}} + \sqrt{\epsilon}\, \mathbf{z}_t$$

- $p(\mathbf{x})$: distribution of $\mathbf{x}$
- $\epsilon > 0$: fixed step
- $\mathbf{z}_t \sim \mathcal{N}(0, I)$: Gaussian noise
- $x_0 \sim \pi(\mathbf{x})$: initial value from known prior

Intuition

$$\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} + \frac{\epsilon}{2} \nabla_{\mathbf{x}} \log p(\mathbf{x}_{t-1}) + \sqrt{\epsilon}\, \mathbf{z}_t$$

- Start from random point $\mathbf{x}_0$
- Repeat:
  - Move towards a maximum of $p(\mathbf{x})$ (using $\nabla_{\mathbf{x}} \log p(\mathbf{x})$)
  - Inject a little noise ($\sqrt{\epsilon}\, \mathbf{z}$) to avoid collapses into local maxima

Challenges

- We cannot have access to $\nabla_{\mathbf{x}} \log p(\mathbf{x})$. Why?

  1. **the manifold hypothesis**: most datasets live in low dimensional manifold embedded in a high dimensional space.

     Space where no $\mathbf{x}$ lives $\Rightarrow \nabla_{\mathbf{x}} \log p(\mathbf{x})$ is undefined

  2. **low data density regions**: even if we could move within the manifold, available data may not be covering all its areas $\Rightarrow$ cannot estimate $\nabla_{\mathbf{x}} \log p(\mathbf{x})$.

Solution (1/2)

- Gaussian noise does not suffer from these challenges
- Perturbing the data with Gaussian noise mitigates them.
- But if we add too much noise, we damage quality
- and if we add too little, we suffer from the challenges.

Solution (2/2)

Let $q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})$ be Gaussian noise

Perturbing the data yields: $q_\sigma(\tilde{\mathbf{x}}) \triangleq \int q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})\, p(\mathbf{x})\, \mathrm{d}\mathbf{x}$

It is proven[1] that we **CAN** learn $\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})$.

[1] Aapo Hyvärinen. "Estimation of Non-Normalized Statistical Models by Score Matching". In: *J. Mach. Learn. Res.* (2005), pp. 695–709.

Learning $\nabla_{\mathbf{x}} \log q_\sigma(\mathbf{x})$: Score matching

Score matching[2] : a technique to learn $\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})$.
Two popular ways to perform score matching:

- Denoising score matching[3]
- Sliced score matching[4]

Both are equally effective. We adopt **denoising score matching**, since it is slightly faster.

[2] Hyvärinen, "Estimation of Non-Normalized Statistical Models by Score Matching".

[3] Pascal Vincent. "A Connection between Score Matching and Denoising Autoencoders". In: *Neural Comput.* (2011).

[4] Yang Song et al. "Sliced Score Matching: A Scalable Approach to Density and Score Estimation". In: *Proceedings of the 35th Uncertainty in Artificial Intelligence Conference*. 2020, pp. 574–584.

Denoising score matching

The objective function our network has to minimize is proven[5] equal to:

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})p(\mathbf{x})}\left[||\mathbf{s}_\theta(\tilde{\mathbf{x}},\sigma) - \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})||_2^2\right]$$

$\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})$ is tractable since $q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})$ is Gaussian.

The optimal network $\mathbf{s}_\theta(\tilde{\mathbf{x}},\sigma)$ satisfies $\mathbf{s}_\theta(\tilde{\mathbf{x}},\sigma) = \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})$ (almost surely).

When $\sigma \to 0$: $\mathbf{s}_\theta(\tilde{\mathbf{x}},\sigma) = \nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x})$.

[5] Vincent, “A Connection between Score Matching and Denoising Autoencoders”.

Noise Conditional Score Networks

- Pre-specify a noise schedule $\{\sigma_i\}_{i=1}^{L}$
- σ_1 must be large (too much noise)
- σ_L must be small (almost no noise)
- A model $\boldsymbol{s}_\theta(\mathbf{x}, \sigma)$ that is trained to learn $\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}), \forall \sigma \in \{\sigma_i\}_{i=1}^{L}$ is named **Noise Conditional Score Network (NCSN)**.

Annealed Langevin Dynamics (1/2)

- If we performed Langevin Dynamics using $\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}})$ for some $\sigma \in \{\sigma_i\}_{i=1}^{L}$ then we will sample from $q_\sigma(\tilde{\mathbf{x}})$. Not good enough!

- Begin with $\nabla_{\tilde{\mathbf{x}}} \log q_{\sigma_1}(\tilde{\mathbf{x}})$ and reduce σ_i by a scale on every iteration.

- Then $\tilde{\mathbf{x}}_t$ "has enough time" to reach a region with well defined score.

- When $\sigma_i \to 0$ we end up sampling from $q_\sigma(\tilde{\mathbf{x}}) \approx p(\mathbf{x})$.

Annealed Langevin Dynamics (2/2)

This modification of Langevin Dynamics is inspired by simulated annealing and is called ***Annealed Langevin Dynamics***:

$$\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2}\,\mathbf{s}_\theta(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\,\mathbf{z}_t$$

α_i is also decreased at every step.

Score-based Modeling in a nutshell (1/2)

Training an NCSN

$$\nabla_{\tilde{\mathbf{x}}} \log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \nabla_{\tilde{\mathbf{x}}} \left\{ \log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{1}{2}\left(\frac{\tilde{\mathbf{x}}-\mathbf{x}}{\sigma}\right)^2 \right\} = -\frac{\tilde{\mathbf{x}}-\mathbf{x}}{\sigma^2}$$

Denoising score matching objective:

$$\ell(\theta,\sigma) \triangleq \frac{1}{2}\mathbb{E}_p \mathbb{E}_{\tilde{\mathbf{x}}\sim\mathcal{N}(\mathbf{x},\sigma^2 I)}\left[\left|\left|\mathbf{s}_\theta(\tilde{\mathbf{x}},\sigma) + \frac{\tilde{\mathbf{x}}-\mathbf{x}}{\sigma^2}\right|\right|^2\right]$$

Combined for all noise scales σ:

$$\mathcal{L}(\theta,\{\sigma_i\}_{i=1}^{L}) \triangleq \frac{1}{L}\sum_{i=1}^{L}\lambda(\sigma_i)\ell(\theta,\sigma_i)$$

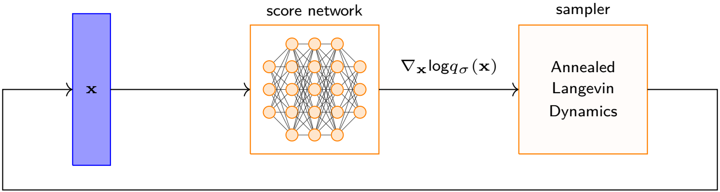
Score-based Modeling in a nutshell (2/2)

Inference

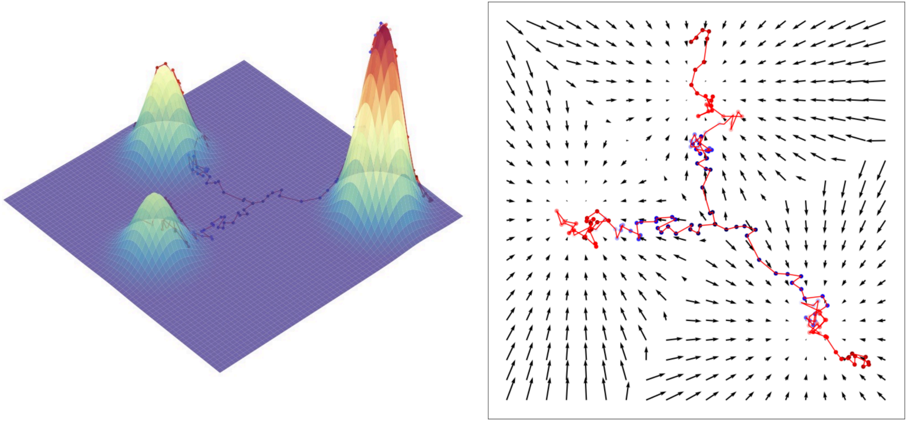
Annealed Langevin Dynamics

```
Require: $\{\sigma\}_{i=1}^{L}, \epsilon, T$
Initialize $\tilde{\mathbf{x}}_{\mathbf{0}}$
for $i \leftarrow 1$ to $L$ do
    $\alpha_i \leftarrow \epsilon \cdot \sigma_i^2 / \sigma_L^2$
    for $i \leftarrow 1$ to $T$ do
        $z_t \sim \mathcal{N}(0, I)$
        $\tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2} \mathbf{s}_\theta(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\, z_t$
    $\widetilde{\mathbf{x}_{\mathbf{0}}} \leftarrow \tilde{\mathbf{x}}_T$
return $\tilde{\mathbf{x}}_{\mathbf{0}}$
```

Recall our diffusion-based recipe


$\mathbf{x}$
score network
$\nabla_{\mathbf{x}} \log q_{\sigma}(\mathbf{x})$
sampler
Annealed Langevin Dynamics

Visualization of Langevin Dynamics



Three random sampling trajectories generated with Langevin dynamics. Credits:[6]

[6] Ling Yang et al. *Diffusion Models: A Comprehensive Survey of Methods and Applications.* 2022.

Diffusion Models

Introduction

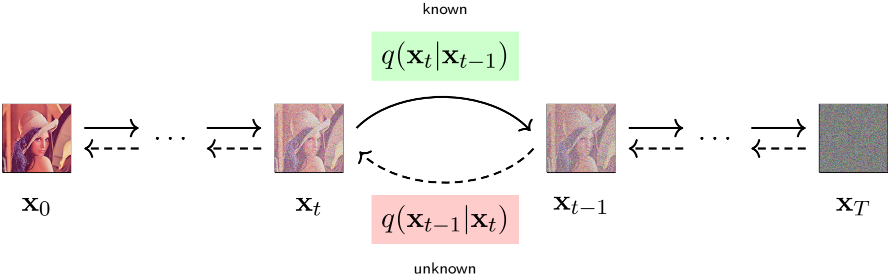
- Diffusion makes a datum to gradually lose its structure.
- Example: an image gradually turning into Gaussian noise.
- Backward diffusion is the reverse procedure.
- Example: Gaussian noise gradually taking the shape of an image.
- Forward diffusion is easy: gradually add noise to an datum.
- Reverse diffusion is not: it allows data generation from noise.

Forward Diffusion (1/3)

The forward diffusion process is fixed and known.

- Sample a datum: $\mathbf{x}_0 \sim p(\mathbf{x})$
- Set the diffusion steps T
- Set a variance schedule: $\{\beta_t \in (0,1)\}_{t=1}^{T}$
- Set a diffusion kernel (here gaussian):

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\, I)$$



known

$q(\mathbf{x}_t|\mathbf{x}_{t-1})$

$\mathbf{x}_0$ $\cdots$ $\mathbf{x}_t$ $\mathbf{x}_{t-1}$ $\cdots$ $\mathbf{x}_T$

$q(\mathbf{x}_{t-1}|\mathbf{x}_t)$

unknown

Forward and Reverse Diffusion Processes

Forward diffusion (2/3)

We can directly calculate $\mathbf{x}_t$ for **any forward** diffusion step:

$$\boxed{\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}}$$

where:

- $\alpha_t = 1 - \beta_t$
- $\bar{\alpha}_t = \prod_{i=0}^{t} \alpha_i$
- $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$

Forward diffusion (3/3)

Proof.

Let us draw $\mathbf{x}_t \sim q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1}, \beta_t\, I)$. Then:

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{a_t}\,\mathbf{x}_{t-1} + \sqrt{1-a_t}\,\boldsymbol{\epsilon}_{t-1} && \text{where } \boldsymbol{\epsilon}_{t-1} \sim \mathcal{N}(0, I)\\
&= \sqrt{a_t}\left[\sqrt{a_{t-1}}\,\mathbf{x}_{t-2} + \sqrt{1-a_{t-1}}\,\boldsymbol{\epsilon}_{t-2}\right] + \sqrt{1-a_t}\,\boldsymbol{\epsilon}_{t-1} && \text{where } \boldsymbol{\epsilon}_{t-2} \sim \mathcal{N}(0, I)\\
&= \sqrt{a_t a_{t-1}}\,\mathbf{x}_{t-2} + \underbrace{\sqrt{a_t - a_t a_{t-1}}\,\boldsymbol{\epsilon}_{t-2} + \sqrt{1-a_t}\,\boldsymbol{\epsilon}_{t-1}}
\end{aligned}$$

The underbraced quantity is the sum of two gaussian distributions with different variances, $\mathcal{N}(0, \sigma_1^2 I)$ and $\mathcal{N}(0, \sigma_2^2 I)$. This sum can be expressed as: $\mathcal{N}(0, (\sigma_1^2 + \sigma_2^2) I)$. Therefore:

$$\mathbf{x}_t = \sqrt{a_t a_{t-1}}\,\mathbf{x}_{t-2} + \sqrt{1 - a_t a_{t-1}}\,\bar{\boldsymbol{\epsilon}}_{t-2} \qquad \text{where } \bar{\boldsymbol{\epsilon}}_{t-2} \sim \mathcal{N}(0, I)$$

By repeating this thinking, $\mathbf{x}_t$ finally becomes equal to:

$$\mathbf{x}_t = \sqrt{\bar{a}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{a}_t}\,\boldsymbol{\epsilon} \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$$

□

Backward diffusion (1/6)

- When $\beta \to 0$, the backward diffusion kernel has the form of the forward one[7] $\implies q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is also Gaussian.
- To calculate $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, we need to use the entire dataset. Bad idea!
- We'll use a diffusion model to help approximate it.
- The reverse diffusion kernel will be:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

- Make $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ and $\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)$ tractable, and we're done!

[7] William Feller. "On the Theory of Stochastic Processes, with Particular Reference to Applications". In: *Proceedings of the [first] Berkeley Symposium on Mathematical Statistics and Probability*. 1949.

Backward Diffusion (2/6)

Notice that $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is not tractable, but $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ **is**:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I})$$

Therefore, we'll approximate $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$, instead.

Backward Diffusion (3/6)

We apply Bayes' rule:

$$
\begin{aligned}
q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) &= q(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\
&\propto \exp\Big[-\frac{1}{2}\Big(\frac{(\mathbf{x}_t-\sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t}+\frac{(\mathbf{x}_{t-1}-\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_{t-1}}-\underbrace{\frac{(\mathbf{x}_t-\sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}}_{C'(\mathbf{x}_t,\mathbf{x}_0)}\Big)\Big] \\
&= \exp\Big[-\frac{1}{2}\Big(\frac{\mathbf{x}_t^2-2\sqrt{\alpha_t}\mathbf{x}_t\mathbf{x}_{t-1}+\alpha_t\mathbf{x}_{t-1}^2}{\beta_t}+\frac{\mathbf{x}_{t-1}^2-2\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0\mathbf{x}_{t-1}+\bar{\alpha}_{t-1}\mathbf{x}_0^2}{1-\bar{\alpha}_{t-1}}-C'\Big)\Big] \\
&= \exp\Big[-\frac{1}{2}\Big(\underbrace{\Big(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}\Big)}_{A}\mathbf{x}_{t-1}^2-\underbrace{\Big(\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t+\frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\Big)}_{B}\mathbf{x}_{t-1}+\underbrace{C(\mathbf{x}_t,\mathbf{x}_0)}_{\Gamma}\Big)\Big]
\end{aligned}
$$

Where $C(\mathbf{x}_t,\mathbf{x}_0)$ is a function not involving $\mathbf{x}_{t-1}$.

Backward Diffusion (4/6)

$$= \exp\Big[-\frac{1}{2}\big(A\,\mathbf{x}_{t-1}^2 - B\,\mathbf{x}_{t-1} + \Gamma\big)\Big]$$

$$= \exp\Bigg[-\frac{1}{2}\left(\sqrt{A}\,\mathbf{x}_{t-1} - \frac{B}{2\sqrt{A}}\right)^2 - \left(\frac{B}{2\sqrt{A}}\right)^2 + \Gamma\Bigg]$$

$$\propto \exp\Bigg[-\frac{1}{2}\left(\sqrt{A}\,\mathbf{x}_{t-1} - \frac{B}{2\sqrt{A}}\right)^2\Bigg]$$

$$= \exp\Bigg[-\frac{1}{2}\left(\frac{\mathbf{x}_{t-1} - \frac{B}{2A}}{\frac{1}{\sqrt{A}}}\right)^2\Bigg]$$

Backward Diffusion(5/6)

Since $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ is Gaussian, it is now clear that:

$$\tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\,\beta_t$$

But we don't know $\mathbf{x}_0$.

In fact, **we can't know** $\mathbf{x}_0$ during backward diffusion.

Backward Diffusion (6/6)

Recall that we can directly calculate $\mathbf{x}_t$ for **any forward** diffusion step:

$$\boxed{\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}}$$

Solve for $\mathbf{x}_0$:

$$\mathbf{x}_0 = \frac{1}{\sqrt{\bar{a}_t}}\left(\mathbf{x}_t - \sqrt{1-\bar{a}_t}\,\boldsymbol{\epsilon}\right)$$

Plug in $\tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0)$:

$$\tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\right)$$

We now train our diffusion model to predict $\boldsymbol{\epsilon}_t$.

And we've finished! 😎

Training the model (1/2)

We want the model to predict $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$:

$$\tilde{\boldsymbol{\mu}}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\Big)$$

The model must minimize the difference between $\tilde{\boldsymbol{\mu}}_\theta(\mathbf{x}_t, t)$ and $\tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0)$.

$$\begin{aligned}
L_t &= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\Big[\frac{1}{2\|\boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\|_2^2}\|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)\|^2\Big] \\
&= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\Big[\frac{1}{2\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t\Big) - \frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\Big)\|^2\Big] \\
&= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\epsilon}}\Big[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\|\boldsymbol{\Sigma}_\theta\|_2^2}\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2\Big]
\end{aligned}$$

Training the model (2/2)

This simplified training loss was empirically found[8] to work better:

$$L_t = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}_t}\left[\left\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right\|^2\right]$$

Final objective: $L_{final} = L_t + C$, where C is a constant not dependent on θ.

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems*. 2020, pp. 6840–6851.

Ancestral Sampling

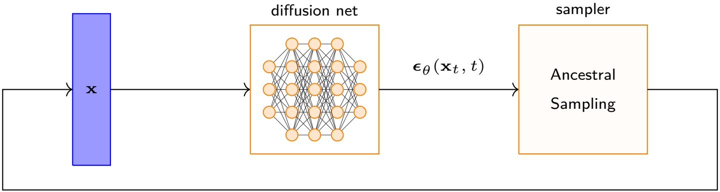
- Sampling is now straightforward.
- In general, when $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 \boldsymbol{I})$, then: $\mathbf{x} = \boldsymbol{\mu} + \sigma \boldsymbol{\epsilon}$
- Therefore: to sample from $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), {\sigma_t}^2\mathbf{I})$, which we approximate via a neural net as $\mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_\theta(\mathbf{x}_t, t), {\sigma_t}^2\mathbf{I})$, we do the following:

$$\mathbf{x}_{t-1} = \tilde{\boldsymbol{\mu}}_\theta(\mathbf{x}_t, t) + \sigma_t\, \boldsymbol{\epsilon} \Longrightarrow$$

$$\boxed{\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t\, \boldsymbol{\epsilon}}$$

- This sampler is named **Ancestral Sampling**.

Recall our diffusion-based recipe



Denoising Diffusion Implicit Model (DDIM)

- How will we accelerate reverse diffusion?

- Reduce steps, i.e. update every $\lceil T/S \rceil$ steps $(S < T) \implies$ quality is severely damaged.

- *Denoising Diffusion Implicit Model*[9] *(DDIM)* is a sampler that reparameterizes the reverse diffusion, allowing for less steps **with significantly less quality loss**.

- DDIM simulates a reverse procedure, whose forward procedure is not a diffusion $\implies$ *Implicit*.

[9] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In: *9th International Conference on Learning Representations, ICLR*. 2021.

DDIM Derivation (1/3)

Diffusion models approximate: $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I})$

Let: $\tilde{\beta}_t = \sigma_t^2$

We have seen that: $\boxed{\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_{t-1}}$

Therefore:

$$\begin{aligned}
\mathbf{x}_{t-1} &= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_{t-1} \\
&= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\boldsymbol{\epsilon}_t + \sigma_t\boldsymbol{\epsilon} \\
&= \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} + \sigma_t\boldsymbol{\epsilon} \Longrightarrow
\end{aligned}$$

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2\mathbf{I})$$

DDIM Derivation (2/3)

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2}\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1 - \bar{\alpha}_t}}, \sigma_t^2\mathbf{I})$$

- Set $\sigma_t^2 = \eta \cdot \tilde{\beta}_t$ ($\eta \in \mathbb{R}$ controls sampling stochasticity).
- $\eta = 0$ sampler becomes **deterministic**. **This is DDIM**.
- $\eta = 1 \implies$ sampler becomes **Ancestral Sampling**.

DDIM Derivation (3/3)

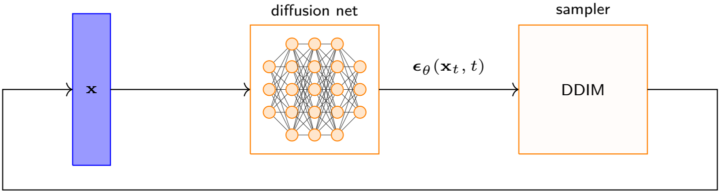
- Setting $\eta = 0$, we get DDIM:

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_{t-1} \Longrightarrow$$

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\,\frac{\mathbf{x}_t - \sqrt{1 - \bar{a}_t} \cdot \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{a}_t}} + \sqrt{1 - \bar{\alpha}_{t-1}}\,\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$$

- Produces best results when performed on a subset of the T steps.

Recall our diffusion-based recipe



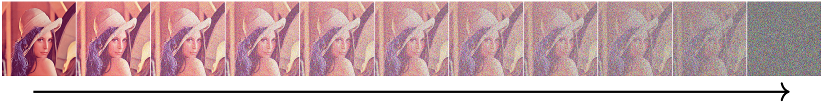
Generation through Stochastic Differential Equations (SDEs)

Continuous diffusion with SDEs (1/2)

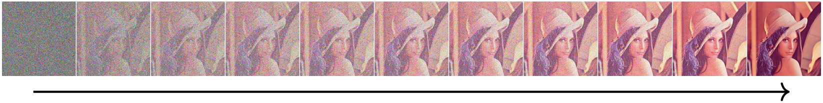
- Both score-based and diffusion models perturb the data using a noise schedule, lets say $\{\sigma_i\}_{i=1}^N$.
- If $N \to \infty$ and $\sigma_{i+1} - \sigma_i \approx 0$ then $\{\sigma_i\}_{i=1}^N$ becomes continuous $\sigma(t)$.
- The noise perturbation procedure becomes a continuous-time stochastic process[10].
- Such processes are solutions to SDEs.
- An SDE has a corresponding reverse SDE.

[10] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021.

Continuous diffusion with SDEs (2/2)



General forward SDE: $d\mathbf{x} = f(\mathbf{x}, t)\, dt + g(t)\, d\mathbf{w}$



General reverse SDE: $d\mathbf{x} = \left[f(\mathbf{x}, t) - g^2(t)\, \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right] dt + g(t)\, d\mathbf{w}$

- $f(\mathbf{x}, t)$: drift coefficient
- $g(t)$: diffusion coefficient
- $\mathbf{w}$: standard Brownian motion
- $d\mathbf{w}$: infinitesmall white noise

SDE that describes score-based modeling

- Markov chain for **score-based modeling**:

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\, \mathbf{z}_{i-1}$$

- As $\{\sigma_i\}_{i=0}^{L} \to \sigma(t)$: the Markov chain $\{\mathbf{x}\}_{i=0}^{L}$ becomes a stochastic process $\{\mathbf{x}(t)\}_{t=0}^{1}$

- $\{\mathbf{x}(t)\}_{t=0}^{1}$ is the solution to the following SDE:

$$d\mathbf{x} = \sqrt{\frac{d[\sigma^2(t)]}{dt}} d\mathbf{w}$$

called **Variance Exploding (VE)** SDE, since $\sigma(t) \in (0, +\infty)$.

SDE describing diffusion modeling

- Markov chain for **diffusion**:

$$\mathbf{x}_i = \sqrt{1-\beta_i}\,\mathbf{x}_{i-1} + \sqrt{\beta_i}\,\mathbf{z}_{i-1}$$

- As $\{\beta_i\}_{i=0}^{T} \to \beta(t)$: the Markov chain $\{\mathbf{x}\}_{i=0}^{L}$ becomes a stochastic process $\{\mathbf{x}(t)\}_{t=0}^{1}$

- $\{\mathbf{x}(t)\}_{t=0}^{1}$ is the solution to the following SDE:

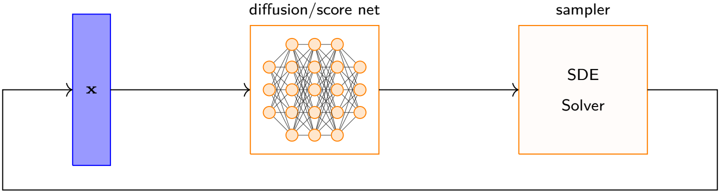
$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}dt + \sqrt{\beta(t)}d\mathbf{w}$$

called **Variance Preserving (VP)** SDE, since $\beta_i \in (0,1]$.

A unified framework

- Diffusion-based methodologies are discretizations of reverse SDE solving.
- A diffusion-based model can be fully determined by an SDE.
- For training we now use infinite noise scales to perturb the data.
- **Sample using ANY SDE solver that solves the reverse SDE of the one that the model was trained on**.
- **Langevin dynamics is a reverse VE SDE solver**.
- **Ancestral sampling is a reverse VP SDE solver**.

Recall our diffusion-based recipe



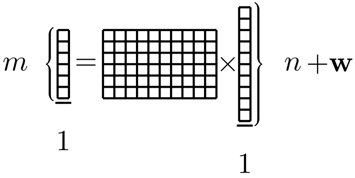
Solving Inverse Problems using Score-based Models

Definition of a linear inverse problem

Linear inverse problem (definition in the context of this thesis):
Recover a signal $\mathbf{x} \in \mathbb{C}^N$ given some measurements $\mathbf{y} \in \mathbb{C}^M$.

$$\mathbf{y} = A\,\mathbf{x} + \mathbf{w}\,, \qquad \mathbf{w} \sim \mathcal{N}_{\mathcal{C}}(0, \sigma^2 I)$$

Corruption matrix $A \in \mathbb{C}^{M \times N}$: it allows for a small quantity of $\mathbf{x}$'s information to survive in $\mathbf{y}$, i.e. the system is underdetermined.



No unique solution. We want to recovering the most probable one.

How score-based models solve inverse problems (1/2)

- Let $\mathbf{x} \sim p$.
- We can train a score-based model and sample from $p(\mathbf{x})$ via Langevin Dynamics.
- But this would produce random samples. What about the measurements $\mathbf{y}$?
- We must sample from $p(\mathbf{x}|\mathbf{y})$.

How score-based models solve inverse problems (2/2)

Since $\mathbf{y}|\mathbf{x}$ is Gaussian $(\mathbf{y} = A\,\mathbf{x} + \mathbf{w}) \implies \nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) = \frac{A^H(\mathbf{y}-A\mathbf{x})}{\sigma^2}$

By applying Bayes' rule, we obtain:

$$\begin{aligned}\nabla_{\mathbf{x}}\log p(\mathbf{x}|\mathbf{y}) &= \nabla_{\mathbf{x}}\log p(\mathbf{x}) + \nabla_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) \\ &= \nabla_{\mathbf{x}}\log p(\mathbf{x}) + \frac{A^H(\mathbf{y} - A\mathbf{x})}{\sigma^2}\end{aligned}$$

And Langevin Dynamics transforms into **Guided Langevin Dynamics**:

$$\underbrace{\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} + \frac{\alpha_i}{2}\mathbf{s}_\theta(\mathbf{x}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\,\boldsymbol{z}_t}_{\Downarrow}$$

$$\mathbf{x}_t \leftarrow \mathbf{x}_{t-1} + \frac{\alpha_i}{2}\left[\mathbf{s}_\theta(\mathbf{x}_{t-1}, \sigma_i) + \frac{A^H(\mathbf{y} - A\mathbf{x}_{t-1})}{\sigma_{\mathbf{y}}^2}\right] + \sqrt{\alpha_i}\,\boldsymbol{z}_t$$

Credits for Guided Langevin Dynamics

- This result was achieved by Jalal et al. in their 2019 work[11]: *Robust Compressed Sensing MRI with Deep Generative Priors*.
- They proposed a general framework to solve linear inverse problems using score-based models and applied it on Compressed Sensing MRI.

[11] Ajil Jalal et al. “Robust Compressed Sensing MRI with Deep Generative Priors”. In: *Advances in Neural Information Processing Systems* (2019).

Contribution - Score-based Implicit Model

Purpose of this thesis

- Guided Langevin Dynamics may need 4000 iterations to produce good reconstructions $\Longrightarrow$ low sampling speed.
- This thesis aims at speeding up this procedure: We want to use pretrained score-based models to solve inverse problems but **faster**.
- We propose two new samplers.
- We test them by performing Compressed Sensing on brain MRIs.

Bijective Relation between VP and VE SDEs (1/4)

- A score model is trained to learn the following:

$$\nabla_{\tilde{\mathbf{x}}_t} \log p(\tilde{\mathbf{x}}_t) = -\frac{\tilde{\mathbf{x}}_t - \mathbf{x}_t}{\sigma^2} = -\frac{\sigma\, \boldsymbol{\epsilon}}{\sigma^2} = -\frac{\boldsymbol{\epsilon}}{\sigma}$$

  where $\epsilon \sim \mathcal{N}(0, I)$.

- Considering VP SDE perturbations, we have:

$$\nabla_{\tilde{\mathbf{x}}_t} \log p(\tilde{\mathbf{x}}_t) = -\frac{\boldsymbol{\epsilon}}{\sqrt{1 - \bar{a}_t}} \qquad (1)$$

- This is a clear connection between a score-based and a diffusion model.

Bijective Relation between VP and VE SDEs (2/4)

The following proof is inspired by Kawar et al.[12] (App. B).

The Variance Exploding and Variance Preserving specifications can be considered equivalent up to rescaling of the noisy latents $\mathbf{x}_t$.

Proof.

- noise schedule for VE SDE is: $\sigma_t \in [0, +\infty)$
- noise schedule for VP SDE is: $a_t \in (0, 1]$ $(a_t = \prod_{i=1}^{t}(1 - \beta_t))$
- noise perturbations for VE: $q(x_t^{VE}|x_0) = \mathcal{N}(x_t^{VE}; x_0, \sigma_t^2 I)$
- noise perturbations for VP: $q(x_t^{VP}|x_0) = \mathcal{N}(x_t^{VE}; \sqrt{a_t}x_0, (1 - a_t)I)$

[12] Bahjat Kawar et al. *Denoising Diffusion Restoration Models*. 2022.

Bijective Relation between VP and VE SDEs (3/4)

- When $\mathbf{x}$ follows a VE trajectory:

$$\mathbf{x}_t^{VE} = \mathbf{x}_0 + \sigma_t \, \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(0, I) \tag{2}$$

- When $\mathbf{x}$ follows a VP trajectory:

$$\mathbf{x}_t^{VP} = \sqrt{a_t}\, \mathbf{x}_0 + \sqrt{1 - a_t}\, \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(0, I) \tag{3}$$

- Divide each member of Eq. 3 by $\sqrt{a_t}$:

$$\frac{\mathbf{x}_t^{VP}}{\sqrt{a_t}} = \mathbf{x}_0 + \sqrt{\frac{1}{a_t} - 1}\, \mathbf{z}$$

- Let $\sqrt{\frac{1}{a_t} - 1} = \sigma_t$. Then: $x_t^{VE} = \frac{x_T^{VP}}{\sqrt{a_t}}$

Bijective Relation between VP and VE SDEs (4/4)

We deduce the following bijections:

$$x_t^{VE} \longleftrightarrow \frac{x_t^{VP}}{\sqrt{a_t}} \tag{4}$$

$$\sigma_t \longleftrightarrow \sqrt{\frac{1}{a_t} - 1} \tag{5}$$

$$\frac{1}{1 + \sigma_t^2} \longleftrightarrow a_t \tag{6}$$

Therefore, we can use a model trained on one SDE to solve the other.

Transforming DDIM for Score-based models (1/3)

- We now reparameterize DDIM (which solves the VP SDE) in order to employ score-based models (which are trained to solve the VE SDE).

$$\mathbf{x}_{t-1}^{VP} \leftarrow \sqrt{a_{t-1}}\,\mathbf{x}_{0t}^{VP} + \sqrt{1-a_{t-1}}\,\boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{VP}) \iff$$

$$\mathbf{x}_{t-1}^{VP} \leftarrow \sqrt{a_{t-1}}\,\frac{\mathbf{x}_t^{VP} - \sqrt{1-a_t}\,\boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{VP})}{\sqrt{a_t}} + \sqrt{1-a_{t-1}}\,\boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{VP}) \overset{\text{Eq.}}{\underset{(1)}{\iff}}$$

$$\mathbf{x}_{t-1}^{VP} \leftarrow \sqrt{a_{t-1}}\,\frac{\mathbf{x}_t^{VP} + \sqrt{1-a_t}\cdot\sqrt{1-a_t}\,\mathbf{s}_\theta(\mathbf{x}_t^{VE},t)}{\sqrt{a_t}} - \sqrt{1-a_{t-1}}\cdot\sqrt{1-a_t}\,\mathbf{s}_\theta(\mathbf{x}_t^{VE},t)$$

$$\mathbf{x}_{t-1}^{VP} \leftarrow \sqrt{a_{t-1}}\,\frac{\mathbf{x}_t^{VP} + (1-a_t)\,\mathbf{s}_\theta(\mathbf{x}_t^{VE},t)}{\sqrt{a_t}} - \sqrt{1-a_{t-1}}\cdot\sqrt{1-a_t}\,\mathbf{s}_\theta(\mathbf{x}_t^{VE},t)$$

Transforming DDIM for Score-based models (2/3)

- Score-based models are trained on VE SDEs, hence **their input must be accordingly perturbed**. Therefore, every time we input $\mathbf{x}_t$ into our model, we must first divide it by $\sqrt{a_t}$ as by Eq. (4).
- We divide both members of the last iteration rule by a_{t-1}. At the same time, we know that $\mathbf{x}^{VE}$ and $\mathbf{x}^{VP}$ converge to the same $\mathbf{x}_0$, as a_t is slowly decreasing. Based on the above and Eq. (4) we get:

$$\mathbf{x}_{t-1}^{VE} \leftarrow \frac{\sqrt{a_t}\,\mathbf{x}_t^{VE} + (1-a_t)\,\mathbf{s}_\theta(\mathbf{x}_t^{VE}, t)}{\sqrt{a_t}} - \frac{\sqrt{1-a_{t-1}} \cdot \sqrt{1-a_t}}{\sqrt{a_{t-1}}}\,\mathbf{s}_\theta(\mathbf{x}_t^{VE}, t) \Longleftrightarrow$$

$$\mathbf{x}_{t-1}^{VE} \leftarrow \mathbf{x}_t^{VE} + \frac{(1-a_t)\,\mathbf{s}_\theta(\mathbf{x}_t^{VE}, t)}{\sqrt{a_t}} - \frac{\sqrt{1-a_{t-1}} \cdot \sqrt{1-a_t}}{\sqrt{a_{t-1}}}\,\mathbf{s}_\theta(\mathbf{x}_t^{VE}, t) \Longleftrightarrow$$

$$\mathbf{x}_{t-1}^{VE} \leftarrow \mathbf{x}_t^{VE} + \left[\frac{1-a_t}{\sqrt{a_t}} - \frac{\sqrt{1-a_{t-1}} \cdot \sqrt{1-a_t}}{\sqrt{a_{t-1}}}\right]\mathbf{s}_\theta(\mathbf{x}_t^{VE}, t)$$

Transforming DDIM for Score-based models (3/3)

From Eq. (6), we obtain:

$$1 - a_t = 1 - \frac{1}{1+\sigma_t^2} = \frac{\sigma_t^2}{1+\sigma_t^2} \tag{7}$$

From Eq. (7) and (6), we obtain:

$$\mathbf{x}_{t-1}^{VE} \leftarrow \mathbf{x}_t^{VE} + \left[\frac{\frac{\sigma_t^2}{1+\sigma_t^2}}{\sqrt{\frac{1}{1+\sigma_t^2}}} - \frac{\sqrt{\frac{\sigma_{t-1}^2}{1+\sigma_{t-1}^2}} \cdot \sqrt{\frac{\sigma_t^2}{1+\sigma_t^2}}}{\sqrt{\frac{1}{1+\sigma_{t-1}^2}}} \right] \mathbf{s}_\theta(\mathbf{x}_t^{VE}, t) \iff$$

$$\mathbf{x}_{t-1}^{VE} \leftarrow \mathbf{x}_t^{VE} + \left[\frac{\sigma_t^2}{\sqrt{1+\sigma_t^2}} - \frac{\sigma_{t-1} \cdot \sigma_t^2}{\sqrt{1+\sigma_t^2}} \right] \mathbf{s}_\theta(\mathbf{x}_t^{VE}, t) \iff$$

$$\mathbf{x}_{t-1}^{VE} \leftarrow \mathbf{x}_t^{VE} + \frac{\sigma_t}{\sqrt{1+\sigma_t^2}} \left[\sigma_t - \sigma_{t-1} \right] \mathbf{s}_\theta(\mathbf{x}_t^{VE}, t)$$

Score-based Implicit Model

We name the last iterative procedure ***Score-based Implicit Model (SBIM)***, since it has been derived in our endeavor to perform DDIM using a score model.

Score-based Implicit Model

$$\mathbf{x}_{t-1} \leftarrow \mathbf{x}_t + \frac{\sigma_t}{\sqrt{1+\sigma_t^2}} \left[\sigma_t - \sigma_{t-1} \right] \mathbf{s}_\theta(\mathbf{x}_t, t)$$

Guided Score-based Implicit Model

We want to incorporate the measurements $\mathbf{y}$ into SBIM to solve inverse problems. Following Jalal et al.'s reasoning, we obtain:

Guided Score-based Implicit Model

$$\mathbf{x}_{t-1} \leftarrow \mathbf{x}_t + \frac{\sigma_t}{\sqrt{1+\sigma_t^2}}\left[\sigma_t - \sigma_{t-1}\right]\left(\mathbf{s}_\theta(\mathbf{x}_t, t) + \frac{A^H(\mathbf{y} - A\mathbf{x}_t)}{\sigma_{\mathbf{y}}^2}\right)$$

A little stochasticity always helps

- A Predictor-Corrector (PC) sampler repeats on every iteration:
  - one step using a predictor
  - several steps using a corrector

- We choose **Guided SBIM** as a predictor and **Langevin Dynamics** as a corrector $\Longrightarrow$ **Guided SBIM with LD corrector**

- Following Jolicoeur-Martineau et al.[13], we also add an extra denoising step at the end of generation.

[13] Alexia Jolicoeur-Martineau et al. *Adversarial score matching and improved sampling for image generation*. 2020.

Guided SBIM with Langevin Dynamics Corrector

Guided SBIM with LD corrector

```
Require: ${\sigma_t}_{t=0}^{N}$, M
$\mathbf{x}_T \sim \mathcal{N}(0, I)$
for $t \leftarrow T$ to 1 do
    $\mathbf{x}_{t-1} \leftarrow \mathbf{x}_t + \frac{\sigma_t}{\sqrt{1+\sigma_t^2}} \left(\sigma_t - \sigma_{t-1}\right) \cdot \left(s_\theta(\mathbf{x}_t, t) + \frac{A^H(\mathbf{y} - A\mathbf{x}_t)}{\sigma_{\mathbf{y}}^2}\right)$
    $\alpha_i \leftarrow \epsilon \cdot \sigma_t^2 / \sigma_L^2$
    for $i \leftarrow 1$ to $M$ do
        $z \sim \mathcal{N}(0, I)$
        $\mathbf{x}_{t-1} \leftarrow \mathbf{x}_{t-1} + \frac{\alpha_i}{2} \left(s_\theta(\mathbf{x}_{t-1}, t-1) + \frac{A^H(y - A\mathbf{x}_{t-1})}{\sigma_{\mathbf{y}}^2}\right) + \sqrt{\alpha_i}\, z$
return $\mathbf{x}_0 + \sigma_0^2\, s_\theta(\mathbf{x}_0, \sigma_0)$
```

Compressed Sensing MRI using SBIM

Multi-coil Compressed Sensing MRI in a nutshell

Inverse Problem: Recover a signal $\mathbf{x} \in \mathbb{C}^N$ given measurements $\mathbf{y} \in \mathbb{C}^M$ (possibly perturbed).

$$\mathbf{y} = A\,\mathbf{x} + \mathbf{w}\,, \qquad \mathbf{w} \sim \mathcal{N}_{\mathcal{C}}(0, \sigma_{\mathbf{y}}^2 I)$$

Multi-coil MRI:

$$\mathbf{y}_i = P F S_i\,\mathbf{x} + \mathbf{w}$$

pointwise multiplication of the i_{th} coil sensitivity map

spatial Fourier transform

k-space sampling operator

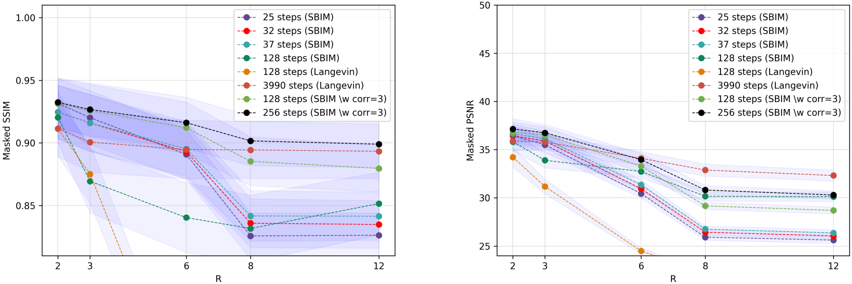
Experiment Configuration

- **Test dataset**: 200 brain MRIs from the fastMRI dataset[14].
- **Neural Net**:
  - NCSNv2 (RefineNet backbone, publicly available at Jalal et al.[15]'s official Github repository)
  - Pretrained on 14,539 brain MRIs.
- **Evaluation Metrics**:
  - **Structural Similarity Index (SSIM)**: quantifies image quality degradation caused by processing.
  - **Peak signal-to-noise ratio (PSNR)**: quantifies the noise corrupting a signal.

[14] Jure Zbontar et al. "fastMRI: An Open Dataset and Benchmarks for Accelerated MRI". In: 2018.

[15] Jalal et al., "Robust Compressed Sensing MRI with Deep Generative Priors".

Experimental Results (1/2)

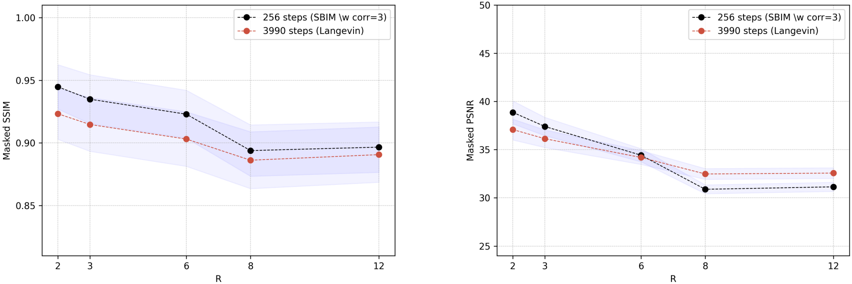


(a) Masked SSIM

(b) Masked PSNR

Sampler Comparison - Vertical Mask

Experimental Results (2/2)

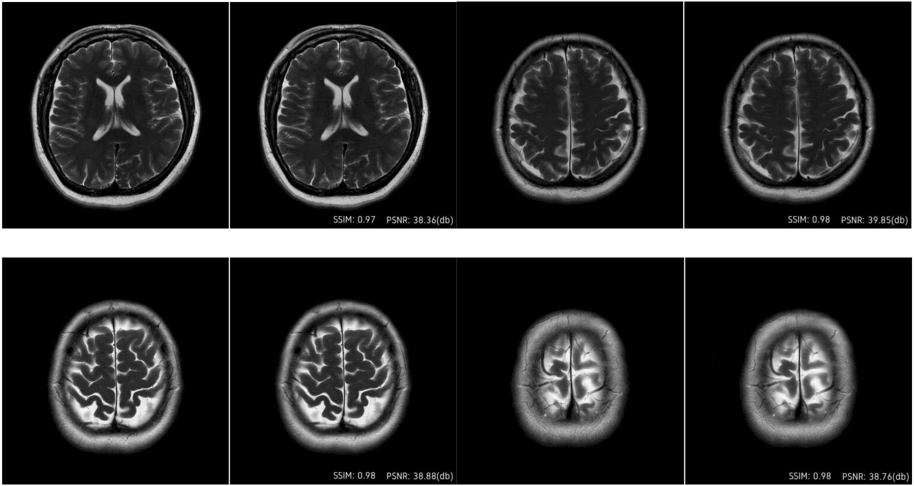


(c) Masked SSIM

(d) Masked PSNR

Sampler Comparison - Horizontal Mask

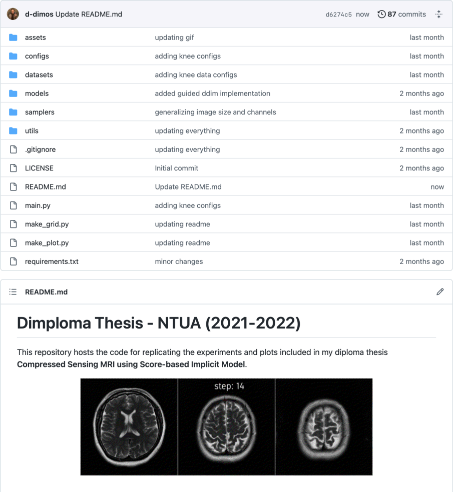
Sample Reconstructions


SSIM: 0.97 PSNR: 38.36(db)
SSIM: 0.98 PSNR: 39.85(db)
SSIM: 0.98 PSNR: 38.88(db)
SSIM: 0.98 PSNR: 38.76(db)

$R = 4$, 96 steps (32 predictor steps with 2 corrector steps)

Code

Code available at: https://github.com/d-dimos/thesis_ntua_sbim



d-dimos Update README.md d6274c5 now **87** commits

assets	updating gif	last month
configs	adding knee configs	last month
datasets	adding knee data configs	last month
models	added guided ddim implementation	2 months ago
samplers	generalizing image size and channels	last month
utils	updating everything	2 months ago
.gitignore	updating everything	2 months ago
LICENSE	Initial commit	2 months ago
README.md	Update README.md	now
main.py	adding knee configs	last month
make_grid.py	updating readme	last month
make_plot.py	updating readme	last month
requirements.txt	minor changes	2 months ago

README.md

Dimploma Thesis - NTUA (2021-2022)

This repository hosts the code for replicating the experiments and plots included in my diploma thesis **Compressed Sensing MRI using Score-based Implicit Model**.

Conclusion

Summary

- We propose (Guided) Score-based Implicit Model and a variation that also uses LD corrector.
- We evaluate our samplers' performance on Compressed Sensing MRI.
- Our Guided SBIM with LD corrector beats the pure LD inverse problem framework in terms of SSIM, PSNR and speed.

Future Directions

- Compare samplers on out-of-distribution data (e.g. knee MRIs).
- Compare samplers using diffusion models.
- Apply knowledge distillation algorithms for sampling acceleration.

Fin