Guidance of a diffusion model for material decomposition in photon-counting computed tomography

C.Vazia[1] A.Bousse[3] B.Vedel[1]
F.Vermet[2] J.Froment[1]

[1, 2]LMBA
Université Bretagne Sud[1]
Université de Bretagne Occidentale[2]

[3]LaTIM
Université de Bretagne Occidentale

Objective

- Spectral CT uses energy-dependent information
  ➜ Allows to reconstruct basis material density images.
- It is an ill-posed inverse problem
  ➜ Requires regularization or *prior*.
- From handcrafted to learned *prior*
  ➜ Generative model as regularization.

Standard Computed Tomography (CT)

We consider the Linear Attenuation Coefficient (LAC) of an object.
For an incoming ray of intensity s, Beer-Lambert's law gives the relative absorption of photons as a function of the LAC :

$$s \longrightarrow \boxed{\mu(l)} \longrightarrow s + ds$$

$$\frac{ds}{s} = -\mu(l)dl.$$

$$\implies s_f = s_0 \exp\left(\underbrace{-\int_{\mathcal{L}} \mu(l)dl}_{:=R(\mu)} \right).$$

Standard Computed Tomography (CT)

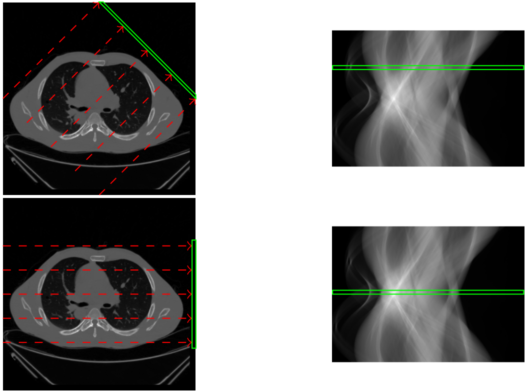


Figure: LAC image $\boldsymbol{\mu} \in \mathbb{R}^{\mathrm{J}}$ and its corresponding projection $R(\boldsymbol{\mu}) \in \mathbb{R}^{\mathrm{I}}$.

Standard Computed Tomography (CT)

In standard CT, we aim to reconstruct a linear attenuation coefficient (LAC) image $\boldsymbol{\mu} \in \mathbb{R}^{\mathrm{J}}$ from the measures $\mathbf{y} \in \mathbb{R}^{\mathrm{I}}$. We assume that $\mathbf{y} = (y_1, y_2, \ldots, y_{\mathrm{I}}) \in \mathbb{R}^{\mathrm{I}}$ is a realization of a random Poisson variable $(\mathbf{Y}|\boldsymbol{\mu})$.

$$(\mathbf{Y} \mid \boldsymbol{\mu}) \sim \mathrm{Poisson}(\bar{\mathbf{y}}(\boldsymbol{\mu})),$$
$$\bar{y}_i(\boldsymbol{\mu}) = s e^{-[R(\boldsymbol{\mu})]_i}.$$

Spectral CT

Spectral CT : Discriminate energy dependance of the LAC.

$$\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}(\epsilon)$$

$$s \leftarrow s(\epsilon)$$

$$\bar{y}_i(\boldsymbol{\mu}) = s e^{-[R(\boldsymbol{\mu})]_i} \leftarrow \int_0^{+\infty} s(\epsilon) e^{-[R\boldsymbol{\mu}(\epsilon)]_i} d\epsilon.$$

$s(\epsilon)$ is the photon flux (intensity) as a function of the energy.

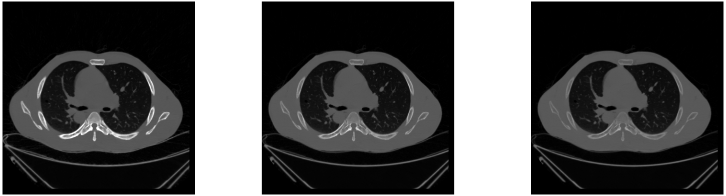
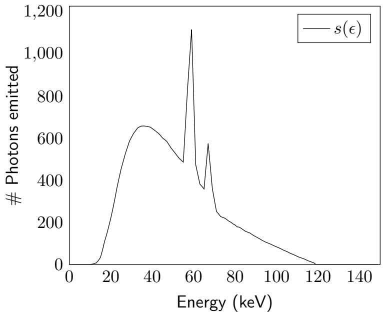


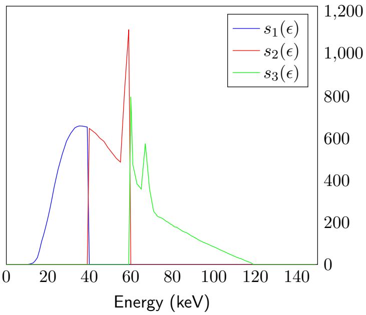
Figure: Example of a spectral image $\boldsymbol{\mu}(\epsilon)$ for three selected energies (40, 80 and 120 keV).

Spectrum partition into energy bins

$$\bar{y}_i(\mathbf{x}) = \int_0^{+\infty} s(\epsilon) e^{-[R\boldsymbol{\mu}(\epsilon)]_i} d\epsilon.$$

We divide the energy spectrum into $\mathtt{K}$ energy bins (intervals of the form $[\epsilon_k, \epsilon_{k+1}]$) and obtain measures for each bins regrouped into the random variable $\mathbf{Y} \in \mathbb{R}^{\mathtt{I}\times\mathtt{K}}$.





Spectral CT

We can leverage this energy dependent LAC function $\boldsymbol{\mu}(\epsilon)$ to obtain $\mathtt{N}$ material images $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{\mathtt{N}})$ using the following relation:

$$\boldsymbol{\mu}(\epsilon) = Q(\mathbf{x}, \epsilon) = \sum_{n=1}^{\mathtt{N}} q_n(\epsilon)\mathbf{x}_n$$

where q_n is the known n-th material attenuation function.

Figure: Example of a material image $\mathbf{x}$ for two materials (bones and soft tissues).

Forward models

➜ **Spectral** forward model:

$$(\mathbf{Y}_{sCT} \mid \boldsymbol{\mu}) \sim \mathrm{Poisson}(\bar{\mathbf{y}}(\boldsymbol{\mu})),$$
$$\bar{y}_{k,i}(\boldsymbol{\mu}) = \int_0^{+\infty} s_k(\epsilon) e^{-[R\boldsymbol{\mu}(\epsilon)]_i} d\epsilon.$$

Forward models

➜ **Spectral** forward model:

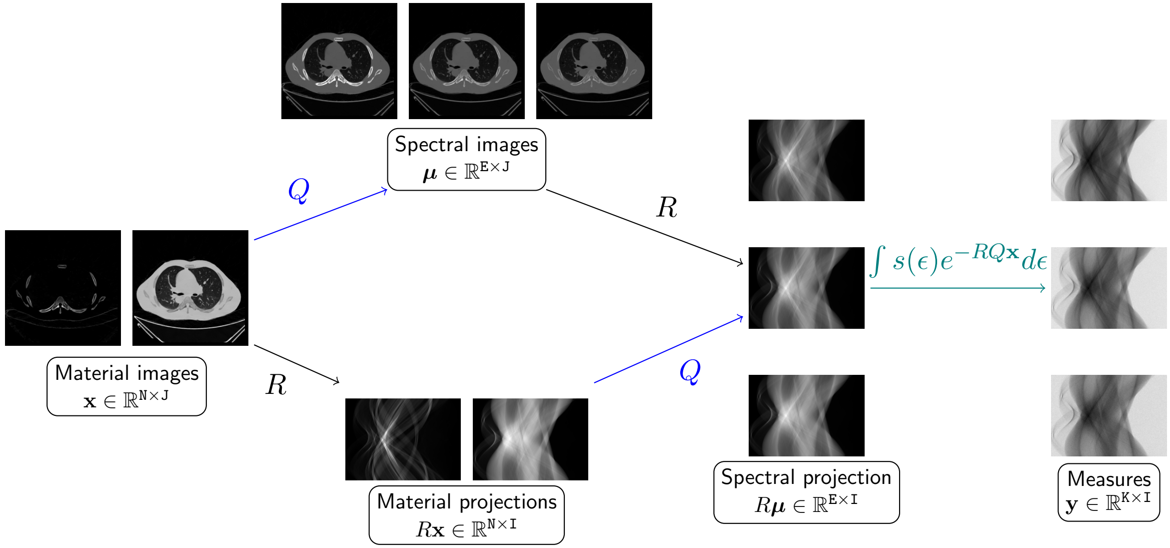
$$(\mathbf{Y}_{sCT} \mid \boldsymbol{\mu}) \sim \mathrm{Poisson}(\bar{\mathbf{y}}(\boldsymbol{\mu})),$$
$$\bar{y}_{k,i}(\boldsymbol{\mu}) = \int_0^{+\infty} s_k(\epsilon) e^{-[R\boldsymbol{\mu}(\epsilon)]_i} d\epsilon.$$

We can replace the spectral image $\boldsymbol{\mu}$ by it's material representation $\boldsymbol{\mu} = Q(\mathbf{x}, \epsilon)$.

➜ **Material** forward model:

$$\begin{aligned}(\mathbf{Y}_{MD} \mid \mathbf{x}) &= (\mathbf{Y}_{sCT} \mid \boldsymbol{\mu} = Q(\mathbf{x}, \epsilon)) \\ &\sim \mathrm{Poisson}(\bar{\mathbf{y}}(\mathbf{x})),\end{aligned}$$
$$\bar{y}_{l,i}(\mathbf{x}) = \int_0^{+\infty} s_k(\epsilon) e^{-[RQ(\mathbf{x},\epsilon)]_i} d\epsilon.$$

Modelisation



One-step and Two-step approaches

Given an energy-binned measurement $\mathbf{y} \in \mathbb{R}^{\mathtt{K} \times \mathtt{I}}$, maximum a *posteriori* (MAP) spectral CT material decomposition can be achieved in two ways:

Two-step approach

$$(\hat{\boldsymbol{\mu}}_k)_{k=1,2,\ldots,\mathtt{K}} \in \underset{\boldsymbol{\mu} \in \mathbb{R}^{\mathtt{K} \times \mathtt{J}}}{\arg\max}\ p_{sCT}(\mathbf{y}|\boldsymbol{\mu}) \cdot p_{\boldsymbol{\mu}}(\boldsymbol{\mu})$$

then solving

$$Q(\mathbf{x}, \epsilon_k) = \hat{\boldsymbol{\mu}}_k \quad \forall k = 1, 2, \ldots, \mathtt{K}.$$

(Or the other way around)

One-step and Two-step approaches

Given an energy-binned measurement $\mathbf{y} \in \mathbb{R}^{\mathtt{K}\times\mathtt{I}}$, maximum a *posteriori* (MAP) spectral CT material decomposition can be achieved in two ways:

Two-step approach

$$(\hat{\boldsymbol{\mu}}_k)_{k=1,2,\ldots,\mathtt{K}} \in \arg\max_{\boldsymbol{\mu}\in\mathbb{R}^{\mathtt{K}\times\mathtt{J}}} p_{sCT}(\mathbf{y}|\boldsymbol{\mu})\cdot p_{\boldsymbol{\mu}}(\boldsymbol{\mu})$$

then solving

$$Q(\mathbf{x}, \epsilon_k) = \hat{\boldsymbol{\mu}}_k \quad \forall k = 1, 2, \ldots, \mathtt{K}.$$

(Or the other way around)

One-step approach

$$\hat{\mathbf{x}} \in \arg\max_{\mathbf{x}\in\mathbb{R}^{\mathtt{N}\times\mathtt{J}}} p_{MD}(\mathbf{y}|\mathbf{x}) \cdot p_{\mathbf{X}}(\mathbf{x})$$

➜ Combines both inverse problems.

➜ Using a generative model to learn the *prior* $p_{\boldsymbol{\mu}}$ / $p_{\mathbf{X}}$.

Diffusion models [3, 6] (forward)

We denote by $\mathbf{W} \in \{\boldsymbol{\mu}, \mathbf{X}\}$ the random vector which can be either $\boldsymbol{\mu}$ or $\mathbf{X}$ depending on which strategy we wish to use (One-step or Two-step).

Diffusion models [3, 6] (forward)

We denote by $\mathbf{W} \in \{\boldsymbol{\mu}, \mathbf{X}\}$ the random vector which can be either $\boldsymbol{\mu}$ or $\mathbf{X}$ depending on which strategy we wish to use (One-step or Two-step).
Denoising Diffusion Probabilistic Model (DDPM) (Ho et al.) version of diffusion models:
Starting from $\mathbf{W}_0 \sim p_{\mathrm{data}}(\mathbf{w}) \approx p(\mathbf{w})$, we apply a Markov process:

$$
\begin{aligned}
&(\mathbf{W}_t \mid \mathbf{W}_{t-1} = \mathbf{w}_{t-1}) \sim \mathcal{N}(\sqrt{\alpha_t}\;\mathbf{w}_{t-1}, (1-\alpha_t)I_J),\; t \in \{1, 2, \dots, T\} \\
\Longrightarrow\;\; &(\mathbf{W}_t \mid \mathbf{W}_0 = \mathbf{w}_0) \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\;\mathbf{w}_0, (1-\bar{\alpha}_t)I_J),\; t \in \{1, 2, \dots, T\}
\end{aligned}
$$

with $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.

Diffusion models [3, 6] (forward)

We denote by $\mathbf{W} \in \{\boldsymbol{\mu}, \mathbf{X}\}$ the random vector which can be either $\boldsymbol{\mu}$ or $\mathbf{X}$ depending on which strategy we wish to use (One-step or Two-step).
Denoising Diffusion Probabilistic Model (DDPM) (Ho et al.) version of diffusion models:
Starting from $\mathbf{W}_0 \sim p_{\text{data}}(\mathbf{w}) \approx p(\mathbf{w})$, we apply a Markov process:

$$\begin{aligned} &(\mathbf{W}_t \mid \mathbf{W}_{t-1} = \mathbf{w}_{t-1}) \sim \mathcal{N}(\sqrt{\alpha_t}\ \mathbf{w}_{t-1}, (1-\alpha_t)I_J),\ t \in \{1, 2, \dots, T\} \\ \implies\ &(\mathbf{W}_t \mid \mathbf{W}_0 = \mathbf{w}_0) \sim \mathcal{N}(\sqrt{\bar{\alpha}_t}\ \mathbf{w}_0, (1-\bar{\alpha}_t)I_J),\ t \in \{1, 2, \dots, T\} \end{aligned}$$

with $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.

$$\mathbf{W}_T \xrightarrow[T \to \infty]{\mathcal{L}} \mathbf{Z} \text{ with } \mathbf{Z} \sim \mathcal{N}(0, I_J).$$

We assume that each $\mathbf{W}_t$ admits a density p_t.

Diffusion models (reverse)

Reverse diffusion process [1, 3]:

$$\mathbf{W}_T \sim \mathcal{N}(0, I_J)$$

$$(\mathbf{W}_{t-1} \mid \mathbf{W}_t = \mathbf{w}_t) \sim \mathcal{N}\left(\mathbf{m}(\mathbf{w}_t, t), \sigma(t)^2 I_J\right),$$

with

$$\mathbf{m}(\mathbf{w}_t, t) = \frac{1}{\sqrt{\alpha_t}}\mathbf{w}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}} \underbrace{\nabla \log(p_t)(\mathbf{w}_t)}_{\text{"score"}}$$

Diffusion models (reverse)

Reverse diffusion process [1, 3]:

$$\mathbf{W}_T \sim \mathcal{N}(0, I_J)$$

$$(\mathbf{W}_{t-1} \mid \mathbf{W}_t = \mathbf{w}_t) \sim \mathcal{N}\left(\mathbf{m}(\mathbf{w}_t, t), \sigma(t)^2 I_J\right),$$

with

$$\mathbf{m}(\mathbf{w}_t, t) = \frac{1}{\sqrt{\alpha_t}}\mathbf{w}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}} \underbrace{\nabla \log(p_t)(\mathbf{w}_t)}_{\text{"score"}}$$

The score function is intractable → Approximation with a neural network $s_\theta(\mathbf{w}_t, t)$ and the *Score Matching by Denoising* technique [5, 7].

$$\hat{\theta} \in \underset{\theta}{\operatorname{Argmin}}\ \mathbb{E}_{\mathbf{W}_0, t, (\mathbf{W}_t | \mathbf{W}_0)} \left\{ \left\| s_\theta(\mathbf{w}_t, t) - \frac{\mathbf{w}_t - \sqrt{\bar{\alpha}_t}\mathbf{w}_0}{1 - \bar{\alpha}_t} \right\|_2^2 \right\},$$

The right term is a proxy for the true score, using Tweedie's formula. Learning the score is in fact closely related to learning a denoising function.

Diffusion models as an a priori

Alter the diffusion process to take into account the measures
$\rightarrow$ *conditional* score function $\nabla \log p_t(\cdot \mid \mathbf{y})$. Bayes rules:

$$\nabla \log p_t(\mathbf{w}_t \mid \mathbf{y}) = \nabla \log p_t(\mathbf{w}_t) + \nabla \log p_t(\mathbf{y} \mid \mathbf{w}_t).$$

[1] *Diffusion posterior sampling for general noisy inverse problems.*, Chung et al. ICLR 2024

Diffusion models as an a priori

Alter the diffusion process to take into account the measures
$\rightarrow$ *conditional* score function $\nabla \log p_t(\cdot \mid \mathbf{y})$. Bayes rules:

$$\nabla \log p_t(\mathbf{w}_t \mid \mathbf{y}) = \nabla \log p_t(\mathbf{w}_t) + \nabla \log p_t(\mathbf{y} \mid \mathbf{w}_t).$$

Diffusion Posterior Sampling [1] approximation:

$$\nabla \log p_t(\mathbf{y} \mid \mathbf{w}_t) \approx \nabla \log p(\mathbf{y} \mid \hat{\mathbf{w}}_0(\mathbf{w}_t, t)).$$

- $p(\mathbf{y} \mid \mathbf{w}_0)$ is one of the two forward models described earlier.
- $\hat{\mathbf{w}}_0(w_t, t)$ is obtained using Tweedie's formula.

[1] *Diffusion posterior sampling for general noisy inverse problems.*, Chung et al. ICLR 2024

Diffusion models as an a priori

Alter the diffusion process to take into account the measures
$\rightarrow$ *conditional* score function $\nabla \log p_t(\cdot \mid \mathbf{y})$. Bayes rules:

$$\nabla \log p_t(\mathbf{w}_t \mid \mathbf{y}) = \nabla \log p_t(\mathbf{w}_t) + \nabla \log p_t(\mathbf{y} \mid \mathbf{w}_t).$$

Diffusion Posterior Sampling [1] approximation:

$$\nabla \log p_t(\mathbf{y} \mid \mathbf{w}_t) \approx \nabla \log p(\mathbf{y} \mid \hat{\mathbf{w}}_0(\mathbf{w}_t, t)).$$

- $p(\mathbf{y} \mid \mathbf{w}_0)$ is one of the two forward models described earlier.
- $\hat{\mathbf{w}}_0(w_t, t)$ is obtained using Tweedie's formula.

$$\mathbf{m}(\mathbf{w}_t, \mathbf{y}, t) = \frac{1}{\sqrt{\alpha_t}}\mathbf{w}_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}\left(\underbrace{\nabla \log p_t(\mathbf{w}_t)}_{\textit{unconditional score}} + \overbrace{\nabla \log p\left(\mathbf{y} \mid \hat{\mathbf{w}}_0(\mathbf{w}_t, t)\right)}^{\textit{conditional guidance}}\right)$$

[1] *Diffusion posterior sampling for general noisy inverse problems.*, Chung et al. ICLR 2024

One-step and Two-step DPS

Two-step Diffusion Posterior Sampling (TDPS) : reconstructs the spectral image $\mathbf{x}$ then applies material composition matrix pseudo inverse in order to obtain material image $\mathbf{z}$.

$$\hat{\theta}_{\mathrm{TDPS}} \in \operatorname*{Argmin}_{\theta} \mathbb{E}_{\boldsymbol{\mu}_0, t, (\boldsymbol{\mu}_t | \boldsymbol{\mu}_0)} \left\{ \left\| s_\theta(\boldsymbol{\mu}_t, t) - \frac{\boldsymbol{\mu}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{\mu}_0}{1 - \bar{\alpha}_t} \right\|_2^2 \right\}.$$

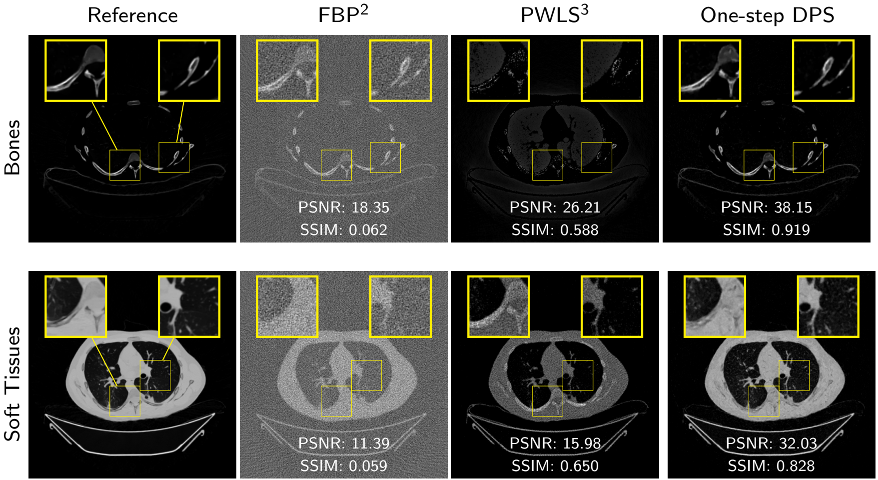
One-step and Two-step DPS

Two-step Diffusion Posterior Sampling (TDPS) : reconstructs the spectral image $\mathbf{x}$ then applies material composition matrix pseudo inverse in order to obtain material image $\mathbf{z}$.

$$\hat{\theta}_{\mathrm{TDPS}} \in \operatorname*{Argmin}_{\theta} \mathbb{E}_{\boldsymbol{\mu}_0, t, (\boldsymbol{\mu}_t | \boldsymbol{\mu}_0)} \left\{ \left\| s_\theta(\boldsymbol{\mu}_t, t) - \frac{\boldsymbol{\mu}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{\mu}_0}{1 - \bar{\alpha}_t} \right\|_2^2 \right\}.$$

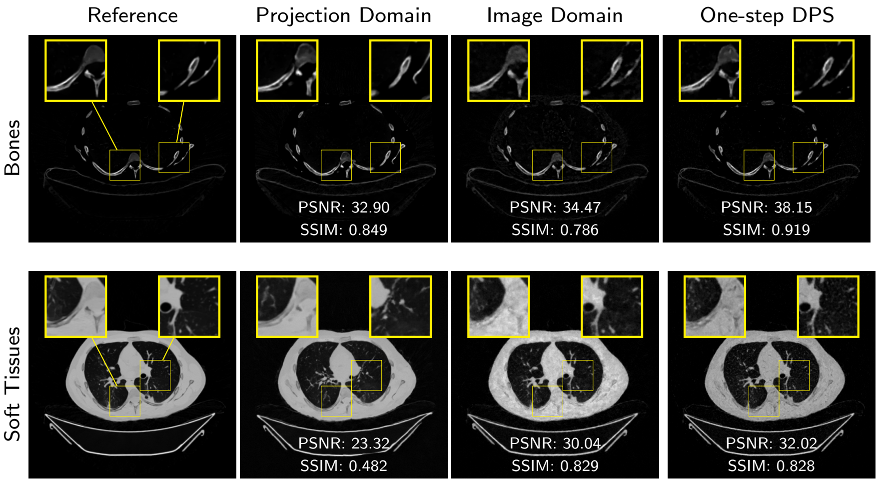
One-step Diffusion Posterior Sampling (ODPS) : Reconstructs directly the material images $\mathbf{z}$

$$\hat{\theta}_{\mathrm{ODPS}} \in \operatorname*{Argmin}_{\theta} \mathbb{E}_{\mathbf{x}_0, t, (\mathbf{x}_t | \mathbf{x}_0)} \left\{ \left\| s_\theta(\mathbf{x}_t, t) - \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{1 - \bar{\alpha}_t} \right\|_2^2 \right\}.$$



[2] Filtered Backprojection and Pseudo inverse

[3] Penalized Weighted Least Square and Pseudo inverse


Reference
Projection Domain
Image Domain
One-step DPS
Bones
Soft Tissues
PSNR: 32.90
SSIM: 0.849
PSNR: 34.47
SSIM: 0.786
PSNR: 38.15
SSIM: 0.919
PSNR: 23.32
SSIM: 0.482
PSNR: 30.04
SSIM: 0.829
PSNR: 32.02
SSIM: 0.828

Remarks and perspectives

- Ordered subsets
- Log-likelihood gradient approximation :

$$\nabla_{\mathbf{w}_t} \log p(\mathbf{y} \mid \hat{\mathbf{w}}_0(\mathbf{w}_t, t)) \propto \nabla_{\mathbf{w}_0} \log p(\mathbf{y} \mid \hat{\mathbf{w}}_0(\mathbf{w}_t, t))$$

- Diffusion on latent spaces or transformation of images (Cascaded diffusion models [4], Wavelets Diffusion Models [2], ...)

References I

[1] Brian D.O. Anderson.
Reverse-time diffusion equation models.
Stochastic Processes and their Applications, 12(3):313–326, 1982.

[2] Florentin Guth, Simon Coste, Valentin De Bortoli, and Stephane Mallat.
Wavelet score-based generative modeling, 2022.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel.
Denoising diffusion probabilistic models.
Advances in neural information processing systems, 2020.

[4] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans.
Cascaded diffusion models for high fidelity image generation, 2021.

References II

[5] Aapo Hyvärinen.
Estimation of non-normalized statistical models by score matching.
Journal of Machine Learning Research, 6:695–709, 01 2005.

[6] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole.
Score-based generative modeling through stochastic differential equations.
In *International Conference on Learning Representations*, 2021.

[7] Pascal Vincent.
A connection between score matching and denoising autoencoders.
Neural Computation, 23(7):1661–1674, 2011.

Tweedie formula for DDPM

From the Markov process $X_t \mid X_{t-1} = x_{t-1}$, we can deduce that:

$$(X_t \mid X_0 = x_0) \sim \mathcal{N}(\sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t)),$$

with $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$.
Tweedie formula :

$$\begin{aligned} \mathbb{E}[\sqrt{\bar{\alpha}_t} X_0 \mid X_t = x_t] &= x_t + (1 - \bar{\alpha}_t) \nabla \log(p_t)(x_t), \\ \rightarrow \hat{x_0}(x_t, t) &= \frac{x_t + (1 - \bar{\alpha}_t) \nabla \log(p_t)(x_t)}{\sqrt{\bar{\alpha}_t}}, \end{aligned}$$

$$\hat{x_0}(x_t, t, s_\theta) = \frac{x_t + (1 - \bar{\alpha}_t) s_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}}.$$